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

Факультет программной инженерии и компьютерной техники

Теория Систем

Лабораторная работа №5

Исследование систем управления

Выполнил:

Маликов Глеб Игоревич

Группа № P3324

Преподаватель:

Русак Алена Викторовна

Санкт-Петербург

2025

**Содержание**

[Задание 3](#_Toc193898099)

[Реализация 4](#_Toc193898100)

[Код 10](#_Toc193898101)

[Вывод 12](#_Toc193898102)

# Задание

1. По заданному дифференциальному уравнению второго порядка (см. варианты задания для лабораторной работы "Оценка параметров математической модели") перейти к описанию в пространстве состояний и в виде передаточной функции.
2. Исследовать динамические и частотные характеристики объекта управления: оценить устойчивость, управляемость и наблюдаемость, построить переходную и импульсную характеристику, диаграмму Боде, фазовый портрет.
3. Перейти к дискретному описанию системы при шаге дискретизации T = 0.1 с.
4. Оценить устойчивость дискретной системы, построить график переходного процесса.

Вариант 5

# Решение

## Переход описаний

Передаточную функцию получим от заданного дифференциального уравнения y¨+3y˙+4y=4u˙+2u, учитывая форму:

Получим:

Для получения модели состояния вводим следующие переменные:

Тогда:

Модель задаётся как:

# ÿ + 3ẏ + 4y = 4 ̇ + 2u  
  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy import signal, integrate  
  
num\_c = [4, 2] # 4s + 2  
den\_c = [1, 3, 4] # s² + 3s + 4  
sys\_c = signal.TransferFunction(num\_c, den\_c)  
  
A = np.array([[0, 1],  
 [-4, -3]])  
B = np.array([[0],  
 [1]])  
C = np.array([[2, 4]])  
D = np.array([[0]])

## Исследование характеристик

### Устойчивость

Система является устойчивой если все действительные части корней уравнения знаменателя (полюса) лежат на левой полуплоскости, то есть меньше нуля.

poles = sys\_c.poles  
print("Poles:", poles)  
if np.all(np.real(poles) < 0):  
 print("The system is stable.")  
else:  
 print("The system is unstable.")

Poles: [-1.5+1.32287566j -1.5-1.32287566j]

Обе действительные части отрицательны, соответственно, система устойчива.

### Управляемость и наблюдаемость

Система считается управляемой если любая точка состояния может быть достигнута из любого начального состояния за конечное время T, подбирая соответствующий вход . Данное условие подтверждается если

В данном случае должна иметь ранг 2:

Q\_c = np.hstack([B, A @ B])  
print("Rank of controllability matrix:", np.linalg.matrix\_rank(Q\_c))  
if np.linalg.matrix\_rank(Q\_c) == A.shape[0]:  
 print("The system is controllable.")  
else:  
 print("The system is uncontrollable.")

Rank of controllability matrix: 2

The system is controllable.

Система считается наблюдаемой если по истории выхода на некотором интервале можно однозначно восстановить начальное состояние . Данное условие проверяется при

В данном случае должна иметь ранг 2:

Q\_o = np.vstack([C, C @ A])  
print("Rank of observability matrix:", np.linalg.matrix\_rank(Q\_o))  
if np.linalg.matrix\_rank(Q\_o) == A.shape[0]:  
 print("The system is observable.")  
else:  
 print("The system is unobservable.")

Rank of observability matrix: 2

The system is observable.

### Переходная и импульсная характеристики

Были построены графики динамических характеристик во времени:

t = np.linspace(0, 10, 1000)  
t\_step, y\_step = signal.step(sys\_c, T=t)  
t\_imp, y\_imp = signal.impulse(sys\_c, T=t)  
plt.figure()  
plt.plot(t\_step, y\_step)  
plt.xlabel("t, s")  
plt.ylabel("y(t)")  
plt.title("Step response")  
plt.grid(True)  
plt.figure()  
plt.plot(t\_imp, y\_imp)  
plt.xlabel("t, s")  
plt.ylabel("y(t)")  
plt.title("Impulse response")  
plt.grid(True)  
plt.show()

A graph with a line

AI-generated content may be incorrect.

Рисунок 1 - Переходная характеристика

A graph of a function

AI-generated content may be incorrect.

Рисунок 2 - Импульсная характеристика

И были вычислены некоторые показатели от этих график:

# Показатели переходной характеристики  
print("Step response parameters:")  
# установившееся значение  
y\_steady = y\_step[-1]  
print("Steady state value: {:.3f}".format(y\_steady))  
  
# перерегулирование  
overshoot = (y\_step.max() - y\_steady) / y\_steady \* 100  
print("Overshoot: {:.3f}%".format(overshoot))  
  
# время подъема  
rise\_time = t\_step[np.where(y\_step >= 0.9 \* y\_steady)[0][0]] - t\_step[np.where(y\_step >= 0.1 \* y\_steady)[0][0]]  
print("Rise time: {:.3f} s".format(rise\_time))  
  
# время установления  
tolerance = 0.02 \* y\_steady  
for idx in range(len(y\_step)):  
 if np.all(np.abs(y\_step[idx:] - y\_steady) <= tolerance):  
 settling\_time = t\_step[idx]  
 break  
else:  
 settling\_time = np.nan # не установилась  
  
print("Settling time: {:.3f} s".format(settling\_time))

Step response parameters:

Steady state value: 0.500

Overshoot: 116.389%

Rise time: 0.120 s

Settling time: 3.684 s

Установившиеся значение — это значение, к которому система стремится в бесконечности.

Перерегулирование — это максимальное отклонение выходного сигнала от установленного значения, выраженное в процентах.

Время подъема — это время, за которое выходной сигнал системы поднимается от 10% до 90% установленного значения.

Время установления — это время, за которое выходной сигнал системы остается в пределах заданного диапазона отклонения от установленного значения (например, ±2%).

# Показатели импульсной характеристики  
print("Impulse response parameters:")  
# устойчивое значение  
y\_steady\_imp = y\_imp[-1]  
print("Steady state value: {:.3f}".format(y\_steady\_imp))  
  
# статический коэффициент k\_DC (равен установленному значению в переходной характеристике)  
k\_DC = np.trapezoid(y\_imp, t\_imp)  
print("Static gain k\_DC: {:.3f}".format(k\_DC))  
  
# Колебательные параметры:  
from scipy.signal import find\_peaks  
peaks,\_ = find\_peaks(y\_imp)  
t\_p1 = t\_imp[peaks[0]]; y\_p1 = y\_imp[peaks[0]]  
t\_p2 = t\_imp[peaks[1]]; y\_p2 = y\_imp[peaks[1]]  
T\_d = t\_p2 - t\_p1; omega\_d = 2\*np.pi/T\_d  
delta = np.log(y\_p1/y\_p2)  
zeta = delta/np.sqrt(4\*np.pi\*\*2 + delta\*\*2)  
omega\_n= omega\_d/np.sqrt(1-zeta\*\*2)  
  
print("Damping ratio zeta: {:.3f}".format(zeta))  
print("Natural frequency omega\_n: {:.3f} rad/s".format(omega\_n))  
print("Damped frequency omega\_d: {:.3f} rad/s".format(omega\_d))

Impulse response parameters:

Steady state value: 0.000

Static gain k\_DC: 0.500

Damping ratio zeta: 0.750

Natural frequency omega\_n: 2.002 rad/s

Damped frequency omega\_d: 1.324 rad/s

Устойчивое значение — это значение, к которому система стремится в бесконечности. При нуле подтверждает устойчивость системы.

Статический коэффициент k\_DC — это значение выходного сигнала системы при постоянном входном сигнале. Совпадает с установившимся значением переходной характеристики.

Демпфирование — это мера затухания колебаний системы. Чем больше значение приближается к 1, тем быстрее система возвращается к установленному значению.

Натуральная частота — это частота, с которой система колебалась бы в отсутствие демпфирования.

Демпфированная частота — это частота, с которой система реально колеблется.

### Диаграммы Боде

A graph with a blue line

AI-generated content may be incorrect.

Рисунок 3 - Амплитудно-частотная характеристика

A graph with a line

AI-generated content may be incorrect.

Рисунок 4 - Фазо-частотная характеристика

### Фазовый портрет

На фазовом портрете решается система

И строятся траектории для разных начальных точек .

def f(t, x):  
 return (A @ x).flatten()  
  
# множество начальных точек  
init\_conds = [np.array([1, 0]),  
 np.array([-1, 0]),  
 np.array([0.5, 1]),  
 np.array([0.5, -1]),  
 np.array([-0.5, 1]),  
 np.array([-0.5, -1]),  
 np.array([0, 1]),  
 np.array([0, -1]),  
 np.array([1, 0.5]),  
 np.array([-1, 0.5]),  
 np.array([1, -0.5]),  
 np.array([-1, -0.5])  
 ]  
  
plt.figure()  
  
# решаем ОДУ для каждой точки  
for x0 in init\_conds:  
 sol = integrate.solve\_ivp(  
 fun=f,  
 t\_span=(0, 10),  
 y0=x0,  
 t\_eval=np.linspace(0,10,800)  
 )  
 plt.plot(sol.y[0], sol.y[1])  
  
plt.xlabel("x₁ = y")  
plt.ylabel("x₂ = ẏ")  
plt.title("Phase portrait")  
plt.grid(True)

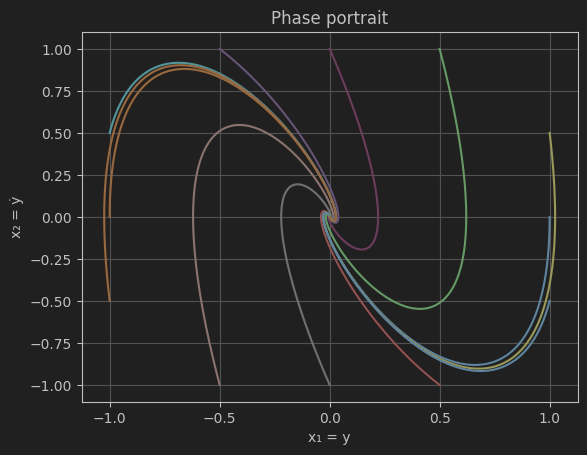


Рисунок 5 - Фазовый портрет

## Дискретное описание системы

Для дискретизации системы используется ZOH-дискретизация (Zero-Order Hold), в котором предполагается, что вход между отсчётами и остаётся постоянным.

Для этого используем следующий код:

T\_s = 0.1  
Ad, Bd, Cd, Dd, dt = signal.cont2discrete((A, B, C, D), T\_s, method='zoh')  
  
print("Ad =\n", Ad)  
print("Bd =\n", Bd)  
print("Cd =\n", Cd)  
print("Dd =", Dd)  
  
# Оценка устойчивости: собственные значения Ad  
eigs = np.linalg.eigvals(Ad)  
print("\nСобственные значения Ad:", eigs)  
print("Максимальное |eig| =", np.max(np.abs(eigs)))  
print("Устойчиво ли (|eig|<1)?", np.all(np.abs(eigs) < 1))  
  
# Построение переходного процесса (step response)  
N\_steps = 100 # число отсчётов  
u = np.ones(N\_steps) # единичная ступень  
t = np.arange(N\_steps) \* T\_s # время в секундах  
  
t\_out, y\_out, x\_out = signal.dlsim((Ad, Bd, Cd, Dd, dt), u)  
  
plt.figure()  
plt.step(t, y.squeeze(), where='post')  
plt.xlabel('t, с')  
plt.ylabel('y[k]')  
plt.title('Дискретный переходный процесс (T\_s = 0.1 с)')  
plt.grid(True)  
plt.show()

Ad =

[[ 0.98191772 0.08581998]

[-0.34327991 0.72445779]]

Bd =

[[0.00452057]

[0.08581998]]

Cd =

[[2 4]]

Dd = [[0]]

Собственные значения Ad: [0.85318776+0.11352916j 0.85318776-0.11352916j]

Максимальное |eig| = 0.860707976425058

Устойчиво ли (|eig|<1)? True

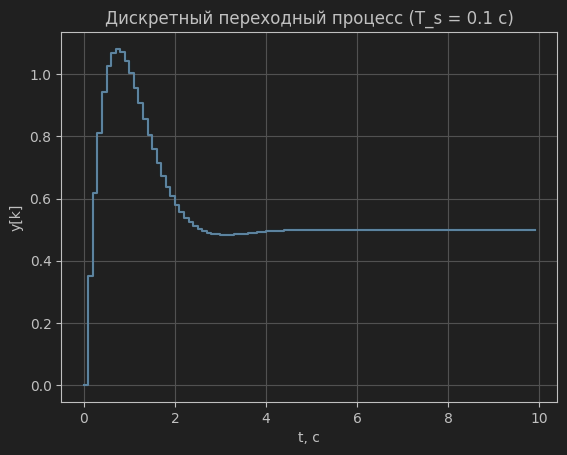


Рисунок 6 - Дискретный переходный портрет

# Вывод

В ходе лабораторной работы была получена передаточная функция и эквивалентная модель состояний. Система оказалась асимптотически устойчивой, управляемой и наблюдаемой. Построенные переходная и импульсная характеристики, Bode-диаграмма и фазовый портрет подтвердили умеренно затухающе-колебательный отклик с перерегулированием. После дискретизации с T=0,1 сек., все собственные числа матрицы Ad лежат внутри единичного круга, что гарантирует устойчивость модели, а её шаговый отклик хорошо повторяет контур непрерывной системы.