**Министерство образования и науки Российской Федерации** ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина»

**РАСЧЕТ ВРЕМЕННЫХ ХАРАКТЕРИСТИК ЛИНЕЙНЫХ СИСТЕМ**

Методические указания и задания к практическому занятию по дисциплине "Основы теории управления" для студентов направления подготовки 09.03.01 - Информатика и вычислительная техника

Студенты:

Бальбола Айад шади Ри-320944

Митреи Джошва Саид Камал РИ-320913

Салама нур самер махмуд саад

Преподаватель:

**Цветков Александр Владимирович**

Екатеринбург 2016

**Расчет временных характеристик линейной системы**

**Вариант 10**

**Данные системы**

Матрицы системы:

* Матрица

A=

* Матрица
* Матрица

**1. Матрица перехода**

Матрица перехода определяется как:

Для данной матрицы AA это:

Результат рассчитан для интервала времени с шагом

**2. Импульсная переходная функция K(t)K(t)**

Импульсная переходная функция вычисляется по формуле:

Результаты для также рассчитаны для интервала времени На графике видно, что экспоненциально уменьшается со временем.

**3. Переходная функция S(t)S(t)**

Переходная функция вычисляется как интеграл от импульсной функции:

Для численного интегрирования использован метод накопления суммы (cumulative sum).

**Графики**

1. **Импульсная переходная функция**
2. **Переходная функция:**

A green line on a white background

Description automatically generated  
 1. import numpy as np

2. import scipy.linalg as la

3. import matplotlib.pyplot as plt

4.

5. # Given matrices for Variant 10

6. A = np.array([[-0.01, 0], [0, -0.02]]) # Dynamics matrix

7. B = np.array([[0.001], [0.002]]) # Input matrix

8. C = np.array([[0.75, 0.25]]) # Output matrix

9.

10. # Time vector for plotting

11. t = np.linspace(0, 100, 500) # From 0 to 100 seconds

12.

13. # Transition matrix Phi(t)

14. Phi\_t = np.array([la.expm(A \* ti) for ti in t])

15.

16. # Impulse response matrix K(t)

17. K\_t = np.array([C @ Phi @ B for Phi in Phi\_t])

18.

19. # Step response matrix S(t) (cumulative integral of K(t))

20. S\_t = np.cumsum(K\_t, axis=0) \* (t[1] - t[0]) # Approximate integral using cumulative sum

21.

22. # Plot Impulse Response

23. plt.figure(figsize=(10, 6))

24. plt.plot(t, K\_t[:, 0, 0], label='Impulse Response (K(t))', color='blue')

25. plt.title("Impulse Response of the System")

26. plt.xlabel("Time (s)")

27. plt.ylabel("K(t)")

28. plt.grid()

29. plt.legend()

30. plt.show()

31.

32. # Plot Step Response

33. plt.figure(figsize=(10, 6))

34. plt.plot(t, S\_t[:, 0, 0], label='Step Response (S(t))', color='green')

35. plt.title("Step Response of the System")

36. plt.xlabel("Time (s)")

37. plt.ylabel("S(t)")

38. plt.grid()

39. plt.legend()

40. plt.show()

41.