МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»
(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и
автоматизированных систем

Лабораторная работа №6

по дисциплине: Системное моделирование

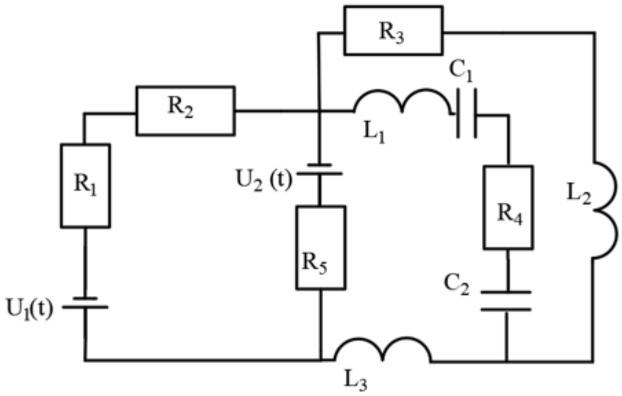
тема: Переходные процессы в электрических цепях

Выполнил: ст. группы ПВ-223
Игнатьев Артур Олегович
Проверил:
Полунин Александр Иванович

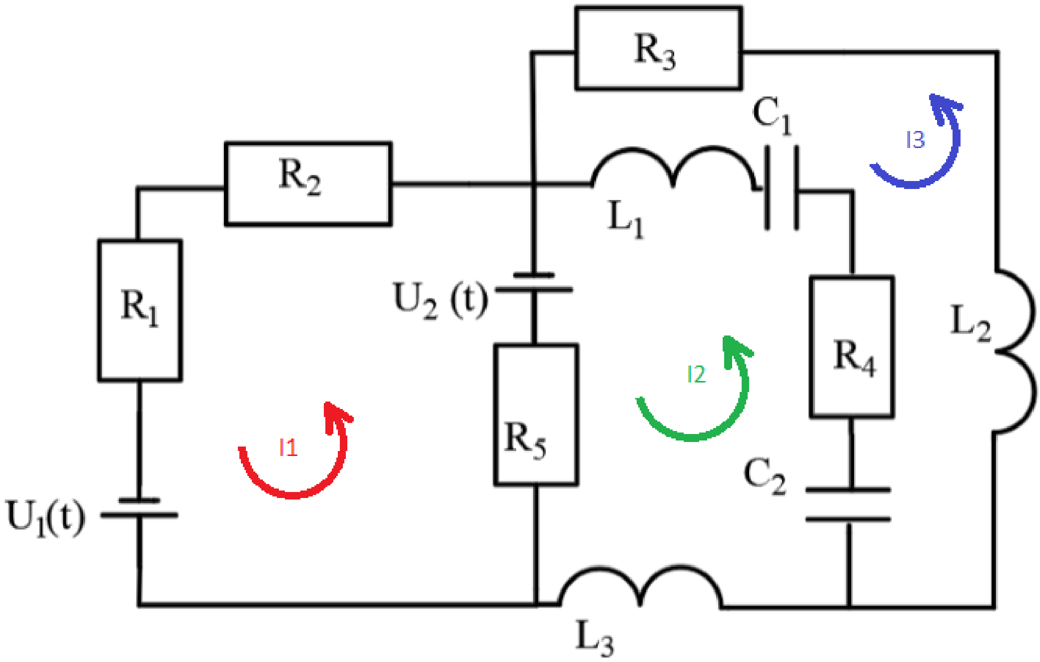
Белгород 2024 г.

Вариант 3

3



$C_1 = 2*10^{-6}$; $C_2 = 10^{-6}$;
$L_1 = 10^{-2}$; $L_2 = 10^{-2}$; $L_3 = 10^{-2}$;
$R_1 = 10$; $R_2 = 11$; $R_3 = 9$;
$R_4 = 5$; $R_5 = 7$;
$U_1 = 10$; $U_2 = 10$.



На основании второго закона Кирхгофа получим:

$$I_1R_1 + I_1R_2 + I_1R_5 - I_2R_5 - I_3R_5 = U_1(t) - U_2(t)$$

$$L_1\frac{dI_2}{dt} + \frac{1}{c_1}\int_0^t I_2dt + I_2R_4 + \frac{1}{c_2}\int_0^t I_2dt + L_3\frac{dI_2}{dt} + L_3\frac{dI_3}{dt} + I_2R_5 + I_3R_5 - I_1R_5 = U_2(t)$$

$$I_3R_3 + L_2\frac{dI_3}{dt} + L_3\frac{dI_3}{dt} + L_3\frac{dI_2}{dt} + I_3R_5 - I_2R_5 - I_1R_5 = U_2(t)$$

Перенесем производные в левую часть и подставим:

$$Q = \int_{0}^{t} \mathrm{I}_2 \, \mathrm{dt}$$

Тогда имеем систему уравнений:

$$\begin{cases} I_1 = \dfrac{U_1 - U_2 + I_2R_5 + I_3R_5}{R_1 + R_2 + R_5} \\ L_1\dfrac{dI_2}{dt} + L_3\dfrac{dI_2}{dt} + L_3\dfrac{dI_3}{dt} = U_2 - \dfrac{Q}{c_1} - I_2R_4 - \dfrac{Q}{c_2} - I_2R_5 + I_1R_5 - I_3R_5 \\ L_2\dfrac{dI_3}{dt} + L_3\dfrac{dI_3}{dt} + L_3\dfrac{dI_2}{dt} = U_2 - I_3R_3 - I_3R_5 - I_2R_5 + I_1R_5 \\ Q = \displaystyle\int_0^t I_2 dt \end{cases}$$

Решим получившуюся систему методом обратной матрицы и получим уравнения:

$$\begin{cases} \dfrac{dI_2}{dt} = \dfrac{L_3 + L_2 * U_2 - L_3 * U_2 + \left((L_3 + L_2) * \left(-R_5(-I_2 - I_3 - I_1)\right) - L_3 * (-I_3R_5)\right) + \left((L_3 + L_2) * \left(\frac{-Q}{C_2}\right) - L_3 * (-I_2R_5)\right) + ((L_3 + L_2) * \left(\frac{-Q}{C_1}\right) - L_3 * (-I_3R_3))}{L_3L_2 + L_1L_3 + L_1L_2} \\ \dfrac{dI_3}{dt} = \dfrac{-L_3 * U_2 - L_3 + L_1 * U_2 + \left(-L_3 * \left(-R_5(-I_2 - I_3 - I_1)\right) - (L_3 + L_1) * (-I_3R_5)\right) + \left(-L_3 * \left(\frac{-Q}{C_2}\right) - (L_3 + L_1) * (-I_2R_5)\right) + (-L_3 * \left(\frac{-Q}{C_1}\right) - (L_3 + L_1) * (-I_3R_3))}{L_3L_2 + L_1L_3 + L_1L_2} \\ \dfrac{dQ_1}{dt} = \dfrac{U_1 - U_2 + I_2R_5 + I_3R_5}{R_1 + R_2 + R_5} \\ \dfrac{dQ_2}{dt} = I_2 \\ \dfrac{dQ_3}{dt} = I_3 \end{cases}$$

Код программы:

```
import numpy as np
from scipy.integrate import ode
import matplotlib.pyplot as plt


# Функция, задающая систему дифференциальных уравнений
def func(t, y):
    # Константы (емкости, индуктивности и сопротивления)
    C1 = 2E-6  # Емкость C1 в Фарадах
    C2 = 1E-6  # Емкость C2 в Фарадах
    L1, L2, L3 = 1E-2, 1E-2, 1E-2  # Индуктивности L1, L2, L3 в Генри
    R1, R2, R3, R4, R5 = 10, 11, 9, 5, 7  # Сопротивления R1, R2, R3, R4,
R5 в Омах
    U1 = 10  # Входное напряжение 1 в Вольтах
    U2 = 10  # Входное напряжение 2 в Вольтах

    # Суммы индуктивностей
```

```
    sum_L3_L2 = L1 + L2
    sum_L1_L3 = L1 + L3

    # Переменные состояния (заряды и токи)
    Q1, Q2, Q3, I1, I2, I3 = y

    # Определение производных
    dI1dt = 0  # Производная тока I1 (заданная как 0)
    dI2dt = ((sum_L3_L2 * U2 - L3 * U2) +
             (sum_L3_L2 * (-R5 * (-I2 - I3 + I1)) - L3 * (-I3 * R5)) +
             (sum_L3_L2 * (-Q2 / C2) - L3 * (-I2 * R5)) +
             (sum_L3_L2 * (-I2 * R4) - L3 * (I1 * R5)) +
             (sum_L3_L2 * (-Q2 / C1) - L3 * (-I3 * R3))) / (L3 * L2 + L1 *
L3 + L1 * L2)
    dI3dt = ((-L3 * U2 + sum_L1_L3 * U2) +
             (-L3 * (-R5 * (-I2 - I3 + I1)) + sum_L1_L3 * (-I3 * R5)) +
             (-L3 * (-Q2 / C2) + sum_L1_L3 * (-I2 * R5)) +
             (-L3 * (-I2 * R4) + sum_L1_L3 * (I1 * R5)) +
             (-L3 * (-Q2 / C1) + sum_L1_L3 * (-I3 * R3))) / (L3 * L2 + L1 *
L3 + L1 * L2)
    dQ1dt = (U1 - U2 + I2 * R5 + I3 * R5) / (R1 + R2 + R5)
    dQ2dt = I2
    dQ3dt = I3

    return [dQ1dt, dQ2dt, dQ3dt, dI1dt, dI2dt, dI3dt]


if __name__ == '__main__':
    # Начальные условия для переменных состояния (заряды и токи)
    y0 = [0, 0, 0, -1, -2, -5]  # Q1, Q2, Q3, I1, I2, I3

    # Временные параметры: начальное и конечное время, шаг интегрирования
    t0, t1 = 0, 0.04  # Начальное и конечное время интегрирования в
секундах
    dt = 0.000001  # Шаг интегрирования в секундах

    # Создание объекта интегратора
    r = ode(func)
    r.set_integrator('dopri5')
    r.set_initial_value(y0, t0)

    # Массивы для хранения времени и результатов интегрирования
    t = [t0]
    y = [y0]

    # Численное интегрирование системы уравнений
    while r.successful() and r.t < t1:
        ti = r.t + dt
        yi = r.integrate(ti)
        t.append(ti)
        y.append(yi)

    # Преобразование списков в массивы numpy для удобства работы
    t = np.array(t)
    y = np.array(y)

    # Построение графиков результатов
    plt.plot(t, y[:, 0], 'r', label='I1(t)')
    plt.plot(t, y[:, 4], 'g', label='I2(t)')
    plt.plot(t, y[:, 5], 'b', label='I3(t)')
    plt.legend(loc='best')
```

```
    plt.xlabel('t')
    plt.grid()
    plt.show()
```

Результат работы программы: