

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



ЗВІТ  
про виконання лабораторної роботи №6  
з теми АНАЛІЗ ДИНАМІЧНИХ ПРОЦЕСІВ НЕЛІНІЙНИХ  
КОРИГУЮЧИХ ЛАНОК СИСТЕМ КЕРУВАННЯ  
з навчальної дисципліни: “Чисельні методи”

*Виконав:*

*студент групи IP-24*

*Шийка Андрій*

*Прийняв:*

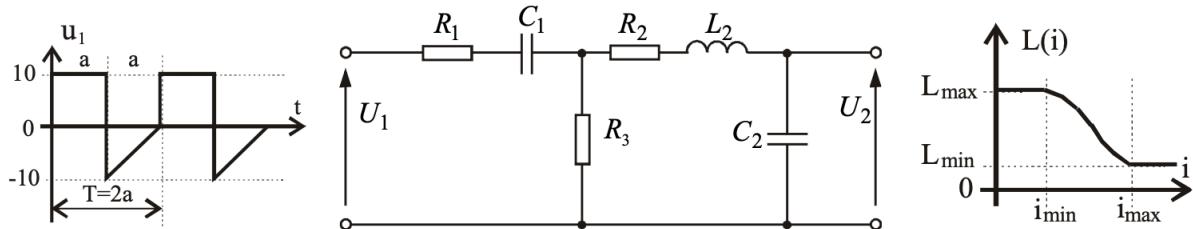
*Сиротюк С. В.*

*Львів - 2025*

**Мета роботи:** провести аналіз динамічних процесів у нелінійній RLC-ланці з використанням чисельних методів інтегрування диференціальних рівнянь.

**Завдання:**

### Варіант 12



Метод: Рунге-Кутта (формули б)).

Дані параметрів схеми:  $C_1 = 34 \text{ мФ}$ ,  $C_2 = 26 \text{ мФ}$ ,  $L_{max} = 8.3 \text{ Гн}$ ,  $L_{min} = 0.83 \text{ Гн}$ ,  $i_{min} = 1 \text{ A}$ ,  $i_{max} = 2 \text{ A}$ ,  $R_1 = 11 \text{ Ом}$ ,  $R_2 = 22 \text{ Ом}$ ,  $R_3 = 33 \text{ Ом}$ .

Апроксимація залежності  $L_2(i_2)$  виконується з вибором розрахункової формули

$$L_2(i_2) = \begin{cases} L_{max}, & \text{if } |i_2| \leq i_{min}, \\ S_3(i_2), & \text{if } i_{min} < |i_2| < i_{max}, \\ L_{min}, & \text{if } |i_2| \geq i_{max}, \end{cases}$$

де  $S_3(i_2)$  - кубічний сплайн. Напруга живлення задана на рисунку, де  $a = 0.02 \text{ с}$ .

**Короткі теоретичні відомості:**

**Розрахунок проміжної напруги у вузлі (після  $C_1$  та перед  $R_2$ ):**

Позначимо напругу у вузлі розгалуження як  $u_A$ . Рівняння струмів для вузла  $u_A$ :

$$\frac{u_1 - u_{C1} - u_A}{R_1} = \frac{u_A}{R_3} + i_2$$

$$u_A \left( \frac{1}{R_1} + \frac{1}{R_3} \right) = \frac{u_1 - u_{C1}}{R_1} - i_2$$

$$u_A = \frac{\frac{u_1 - u_{C1}}{R_1} - i_2}{\frac{1}{R_1} + \frac{1}{R_3}}$$

**Система диференціальних рівнянь у нормальній формі Коші:**

1. Для конденсатора  $C_1$ :

$$\frac{du_{C1}}{dt} = \frac{1}{C_1} \cdot \frac{u_1 - u_{C1} - u_A}{R_1}$$

2. Для нелінійної індуктивності  $L_2$ : Рівняння другого контуру:  $u_A - i_2 R_2 - u_{L2} - u_{C2} = 0$ . Звідси напруга на індуктивності:  $u_{L2} = u_A - i_2 R_2 - u_{C2}$ . Враховуючи нелінійність  $L(i)$ , запишемо:

$$\frac{di_2}{dt} = \frac{u_A - u_{C2} - i_2 R_2}{L(i_2)}$$

3. Для конденсатора  $C_2$  (вихідна напруга  $U_2$ ): Струм, що втікає в вузол  $C_2$ , дорівнює  $i_2$ . Вважаючи  $R_4$  відсутнім (згідно з даними варіанту):

$$\frac{du_{C2}}{dt} = \frac{i_2}{C_2}$$

**Алгоритм розв'язання:**

Для чисельного інтегрування використано метод Рунге-Кутта 3-го порядку, який забезпечує третій порядок точності.

Розрахункові формули для кроку  $h$ :

1.  $K_1 = h \cdot f(t_n, y_n)$
2.  $K_2 = h \cdot f(t_n + \frac{1}{3}h, y_n + \frac{1}{3}K_1)$
3.  $K_3 = h \cdot f(t_n + \frac{2}{3}h, y_n + \frac{2}{3}K_2)$
4.  $y_{n+1} = y_n + \frac{1}{4}(K_1 + 3K_2 + K_3)$

**Код програми:**

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

R1 = 11.0
R2 = 22.0
R3 = 33.0

# R4 в умові відсутній, вважаємо розрив кола (некінчений опір)
C1 = 34e-6 # 34 мкФ
C2 = 26e-6 # 26 мкФ

L_MAX = 8.3
L_MIN = 0.83
I_MIN = 1.0
I_MAX = 2.0

a = 0.02      # параметр часу
T = 2 * a     # Период
T_SIM = 5 * T # Час моделювання (5 періодів)
POINTS_PER_PERIOD = 100
h = T / 400   # Крок інтегрування (згідно вимог <= T/400)

def get_u1(t):
    """
```

Формування вхідної напруги  $u_1(t)$ .

Графік: Трикутна форма (пилкоподібна або трикутна).

Трактування 'а а' та  $T=2a$ :

```

0 -> a/2: зростання 0 -> 10
a/2 -> 1.5a: спад 10 -> -10
1.5a -> 2a: зростання -10 -> 0
"""
t_cycle = t % T # Час в межах одного періоду
# Амплітуда
U_peak = 10.0

# Точки перегину для симетричного трикутника з періодом 2a
t1 = a / 2
t2 = 3 * a / 2

if t_cycle <= t1:
    # Лінійне зростання: y = kx => 10 = k(a/2) => k = 20/a
    return (U_peak / t1) * t_cycle
elif t_cycle <= t2:
    # Лінійний спад від 10 до -10
    # y - 10 = k(x - a/2). k = -20/a
    return U_peak - (2 * U_peak / a) * (t_cycle - t1)
else:
    # Лінійне зростання від -10 до 0
    return -U_peak + (U_peak / (T - t2)) * (t_cycle - t2)

def get_inductance(i_val):
"""
Аproxимація нелінійної індуктивності L(i).
Використовується кубічний сплайн на інтервалі [i_min, i_max].
"""
abs_i = abs(i_val)

if abs_i <= I_MIN:
    # Код для малих значень i
    pass
else:
    # Код для великих значень i
    pass

```

```

        return L_MAX

elif abs_i >= I_MAX:

    return L_MIN

else:

    # Розрахунок коефіцієнтів сплайну

    # x1 = I_MIN, x2 = I_MAX, y1 = L_MAX, y2 = L_MIN

    # m1 = 0, m2 = 0 (похідні на кінцях рівні нулю для гладкості)

    x = abs_i

    x1 = I_MIN

    x2 = I_MAX

    h_spline = x2 - x1

    # Базисні функції Ерміта

    # y(x) = (b1*y1 + b2*y2)/h^3 + (b3*m1 + b4*m2)/h^2

    # Оскільки m1=m2=0, друга частина формули зникає.

    b1 = (2 * (x - x1) + h_spline) * (x2 - x)**2

    b2 = (2 * (x2 - x) + h_spline) * (x - x1)**2

    L_spline = (b1 * L_MAX + b2 * L_MIN) / (h_spline**3)

    return L_spline


def system_equations(t, Y):

    """
    Система диференціальних рівнянь.

    Y[0] = uC1
    Y[1] = i2
    Y[2] = uc2
    """

    uc1 = Y[0]
    i2 = Y[1]
    uc2 = Y[2]

    u1 = get_u1(t)

```

```

# Розрахунок проміжної напруги uA (вузол між R1, R3, R2)
# Рівняння вузла: (u1 - uc1 - uA)/R1 = uA/R3 + i2
# (u1 - uc1)/R1 - i2 = uA(1/R1 + 1/R3)

G_eq = (1/R1 + 1/R3)
term_source = (u1 - uc1) / R1

uA = (term_source - i2) / G_eq

# 1. duC1/dt = i1 / C1
# i1 = (u1 - uc1 - uA) / R1
i1 = (u1 - uc1 - uA) / R1
duc1_dt = i1 / C1

# 2. di2/dt = uL / L(i)
# uA - i2*R2 - uL - uc2 = 0 => uL = uA - uc2 - i2*R2
uL = uA - uc2 - i2 * R2
L_val = get_inductance(i2)
di2_dt = uL / L_val

# 3. duC2/dt = iC2 / C2
# i2 = iC2 (оскільки R4 відсутній/нескінчений)
# Якщо б R4 був: iC2 = i2 - uc2/R4
duc2_dt = i2 / C2

return np.array([duc1_dt, di2_dt, duc2_dt])

def runge_kutta_3_b(t, Y, h):
    """
    Метод Рунге-Кутта 3-го порядку
    Аналог методу Ралстона.
    """
    # K1 = h * f(xn, yn)
    k1 = h * system_equations(t, Y)

```

```

# K2 = h * f(xn + 1/3 h, yn + 1/3 K1)
k2 = h * system_equations(t + h/3.0, Y + k1/3.0)

# K3 = h * f(xn + 2/3 h, yn + 2/3 K2)
k3 = h * system_equations(t + 2.0*h/3.0, Y + 2.0*k2/3.0)

# Yn+1 = Yn + (K1 + 3K3)/4
# Перевірка: метод порядку 3, ваги 1/4 і 3/4.
Y_next = Y + (k1 + 3*k3) / 4.0

return Y_next

```

```

t_values = []
u1_values = []
uc1_values = []
i2_values = []
uc2_values = []
l_values = [] # Для перевірки зміни індуктивності

# Початкові умови [uC1, i2, uC2]
Y = np.array([0.0, 0.0, 0.0])
t = 0.0

print("Початок розрахунку...")
print(f"Крок: {h}, Кількість точок: {int(T_SIM/h)}")

while t <= T_SIM:
    # Збереження результатів
    t_values.append(t)
    u1_values.append(get_u1(t))
    uc1_values.append(Y[0])
    i2_values.append(Y[1])
    uc2_values.append(Y[2])

```

```

l_values.append(get_inductance(Y[1]))


# Крок інтегрування
Y = runge_kutta_3_b(t, Y, h)
t += h


data = {
    'Time': t_values,
    'U1': u1_values,
    'Uc1': uc1_values,
    'I2': i2_values,
    'U2': uc2_values
}

df = pd.DataFrame(data)

df.to_csv('rezult_variant12.dat', sep='\t', index=False,
          float_format='%.6f')


print("Розрахунок завершено. Результати збережено в
      rezult_variant12.dat")

# --- ПОВУДОВА ГРАФІКІВ ---
plt.figure(figsize=(12, 10))

plt.subplot(3, 1, 1)
plt.plot(t_values, u1_values, label='Вхідна напруга U1(t)',
          color='blue')

plt.plot(t_values, uc2_values, label='Вихідна напруга U2(t)',
          color='red', linestyle='--')

plt.title('Графіки напруг')
plt.xlabel('Час, с')
plt.ylabel('Напруга, В')
plt.legend()
plt.grid(True)

plt.subplot(3, 1, 2)
plt.plot(t_values, i2_values, label='Струм індуктивності i2(t)',
          color='green')

```

```

plt.title('Графік струму')
plt.xlabel('Час, с')
plt.ylabel('Струм, А')
plt.legend()
plt.grid(True)

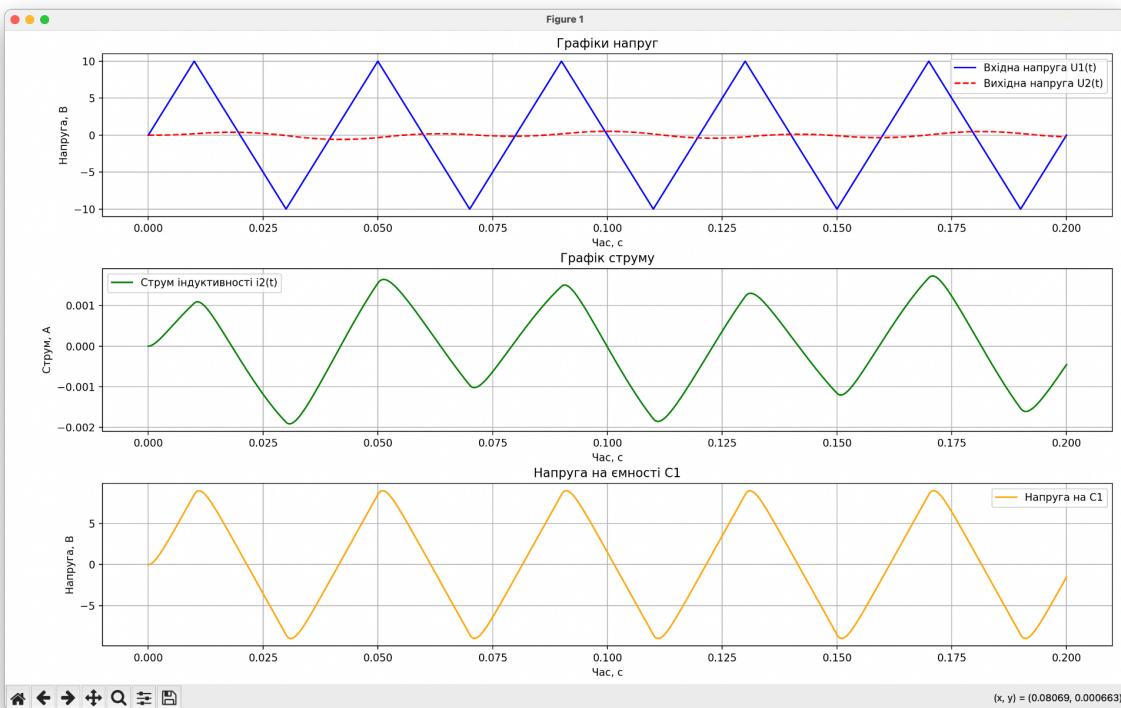
plt.subplot(3, 1, 3)
plt.plot(t_values, ucl_values, label='Напруга на C1', color='orange')
plt.title('Напруга на емності C1')
plt.xlabel('Час, с')
plt.ylabel('Напруга, В')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()

```

## Результат:

Time	U1	Uc1	I2	U2
0.000000	0.000000	0.000000	0.000000	0.000000
0.000100	0.100000	0.003268	0.000000	0.000001
0.000200	0.200000	0.012793	0.000002	0.000005
0.000300	0.300000	0.028171	0.000004	0.000015
0.000400	0.400000	0.049027	0.000007	0.000035
0.000500	0.500000	0.075006	0.000010	0.000067
0.000600	0.600000	0.105781	0.000014	0.000113
0.000700	0.700000	0.141040	0.000019	0.000177
0.000800	0.800000	0.180497	0.000024	0.000261
0.000900	0.900000	0.223880	0.000030	0.000365
0.001000	1.000000	0.270938	0.000037	0.000493



## Висновок:

У ході виконання комплексної лабораторної роботи було:

- Проведено аналіз схеми варіанту №12 та складено систему нелінійних алгебро-диференціальних рівнянь, що описують її динаміку.
- Розроблено програму мовою Python, яка реалізує чисельне розв'язання системи рівнянь методом Рунге-Кутта 3-го порядку (модифікація "б").
- Реалізовано алгоритм апроксимації нелінійної характеристики індуктивності за допомогою кубічного сплайну для забезпечення гладкості функції  $L(i)$ .
- Отримано графіки переходних процесів для вхідної напруги, напруг на конденсаторах та струму індуктивності протягом 5 періодів.
- Результати показують, що наявність нелінійної індуктивності призводить до зміни форми струму порівняно з лінійним колом, а параметри  $C_1$  та  $C_2$  впливають на згладжування та фазовий зсув вихідного сигналу  $U_2$ .