Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №3 по дисциплине

"Физические основы информационных технологий"

Название: Прохождение сигнала через четырёхполюсник

|  |  |
| --- | --- |
| Фамилия И.О.: | Хабибуллина А.М |
| группа: | 1303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 22.11.23

Санкт-Петербург 2023

Условие задания 3

Сигнал поступает на вход четырёхполюсника (контакты 1 и 2), показанного на рис.1. Четырехполюсник состоит из катушек индуктивности (L1 и L2), конденсаторов (C1 и C2) и резисторов (R1, R2, R3 и R4). Сигнал представляет собой суперпозицию гармонических функций:

.

Определить во сколько раз уменьшится амплитуда *k*-й гармоники на выходе четырехполюсника по сравнению с входным значением. В данном задании сигнал задается в виде массива из N элементов, который нужно взять из текстового файла. Каждый элемент массива соответствует определённому моменту времени *ti*. Интервал между соседними моментами времени также дается в задании и обозначается как *δt=ti+1-ti*. В данном задании требуется найти АЧХ передаточной характеристики четырехполюсника, а также применить алгоритм дискретного преобразования Фурье для определения спектрального состава сигнала. Необходимо построить график АЧХ в диапазоне циклических частот от 0 до 100 рад/с, а также графики сигнала и его спектра.

Параметры четырехполюсника, номер гармоники, номера выходных контактов четырехполюсника, массивы со входным сигналом можно найти в файле “FOIT\_IDZ3.xlsx”.

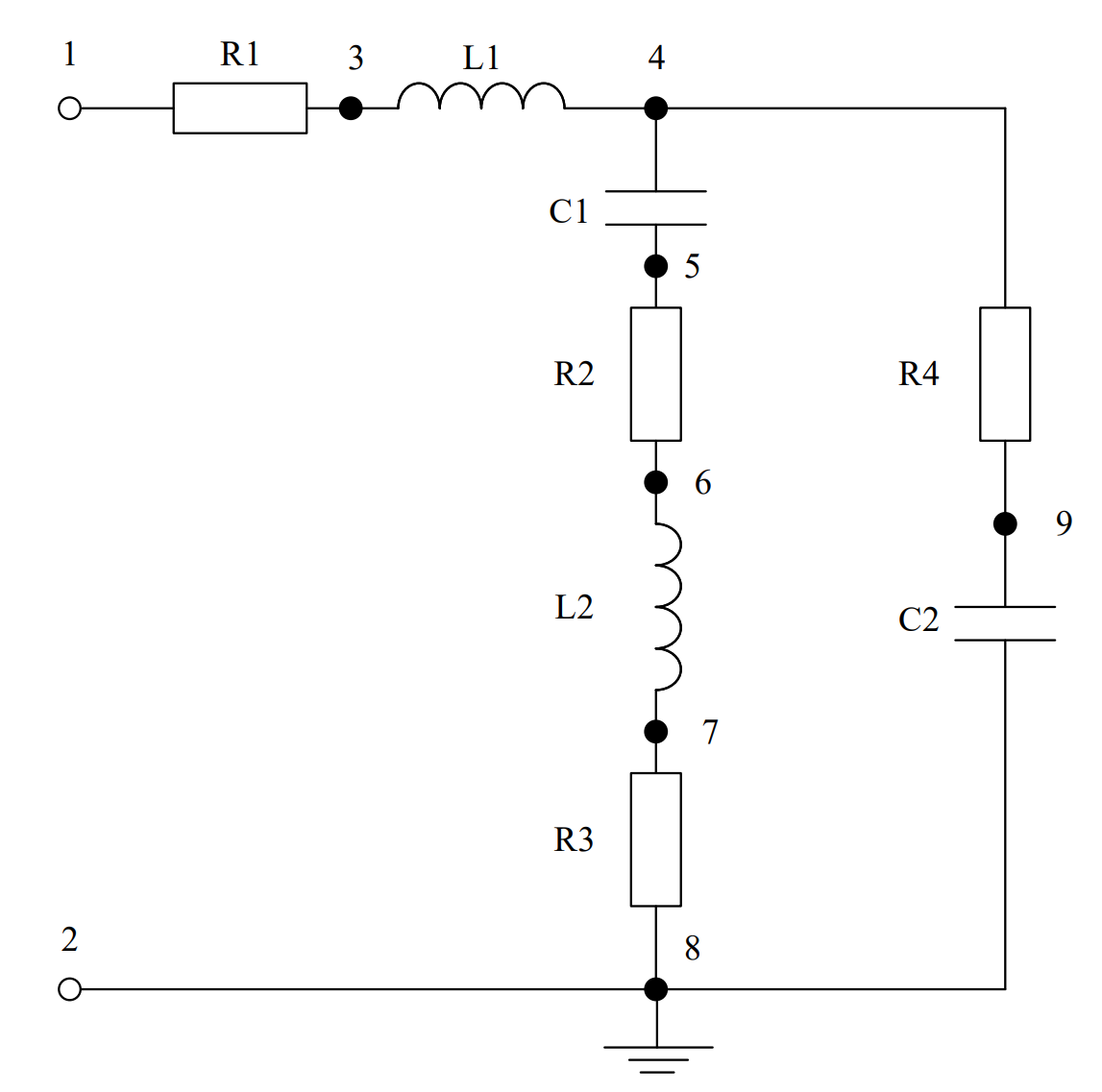


Рисунок 1. Исследуемый четырехполюсник

Отношение амплитуды *k*-й гармоники на выходе к амплитуде *k*-й гармонике на входе () нужно вычислить и записать в файл IDZ3.txt в папке IDZ3. Помимо текстового файла IDZ3.txt в папке IDZ3 должен находиться Word-файл с отчетом, а также файл с кодом (Python, Mathcad, Mathematica). Для лучшего понимания отчетности смотрите папку “Пример организации яндекс-папки студентов”.

Пример содержания файла IDZ3.txt:

4.53258

**Дано:**

**Вариант13**

L1 = 13.5934758394208

L2 = 0.736428750945079

C1 = 1.14807684098339E-05

C2 = 1.19450718128781E-05

R1 = 104.70009686908

R2 = 32.9926577881827

R3 = 1095.36423315565

R4 = 508.587107056158

Количество отсчетов(элементов) = 8192

Время между соседними отсчетами = 0.0196349540849362

Контакты выхода 4 и 9

Номер гармонии = 3

**Основные теоретические сведения**  
Передаточная функция (АЧХ) — это отношение амплитуды выходного сигнала к амплитуде входного сигнала в зависимости от частоты. Она обычно представляется как функция циклической частоты.

Математически передаточная функция *H*(*ω*) может быть записана как:



где *U*вых​(*ω*) — амплитуда выходного сигнала, а *U*вх​(*ω*) — амплитуда входного сигнала.

Дискретное преобразование Фурье (ДПФ): ДПФ используется для анализа частотного содержания дискретных сигналов. Для последовательности сигналов *x*[*n*] ДПФ определяется следующим образом:



где *X*(*k*) — комплексные амплитуды частот, *n* — индекс семпла, *N* — общее количество семплов.

Амплитуда гармоники: Амплитуда k-й гармоники в сигнале представляется как *Ampk*​. Входной сигнал представляет собой сумму гармонических функций:



**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

IDZ3.py

import numpy as np

import matplotlib.pyplot as plt

L1 = 13.5934758394208

L2 = 0.736428750945079

C1 = 0.0000114807684098339

C2 = 0.0000119450718128781

R1 = 104.70009686908

R2 = 32.9926577881827

R3 = 1095.36423315565

R4 = 508.587107056158

dt = 0.0196349540849362

N1 = 8192

time\_values = np.arange(0, N1 \* dt, dt)

signal = np.loadtxt("13.txt")

plt.figure()

plt.plot(time\_values, signal, label='Сигнал')

plt.xlabel('Время')

plt.ylabel('Значение сигнала')

plt.legend()

plt.show()

Fsig = np.fft.fft(signal)

frequency\_values = np.fft.fftfreq(N1, dt)

plt.figure()

FourAbs = np.abs(Fsig)

plt.plot(2 \* np.pi \* frequency\_values, FourAbs, label='Спектр')

plt.xlabel('Частота')

plt.ylabel('Амплитуда')

plt.legend()

plt.show()

w\_values = np.linspace(0, 100, 1000)

with np.errstate(divide='ignore', invalid='ignore'):

imp\_1 = R4 + 1 / (1j \* w\_values \* C2)

imp\_2 = 1 / (1j \* w\_values \* C1) + R2 + 1j \* w\_values \* L2 + R3

imp\_parall = 1 / (1 / imp\_1 + 1 / imp\_2)

I\_1 = 1 / (R1 + 1j \* w\_values \* L1 + imp\_parall)

U\_parall = I\_1 \* imp\_parall

I\_2 = U\_parall / imp\_1

U\_out = I\_2 \* R4

ACH = np.abs(U\_out)

plt.figure()

plt.plot(w\_values, ACH, label='АЧХ')

plt.xlabel('Частота')

plt.ylabel('Амплитуда')

plt.legend()

plt.show()

w = 30;

imp\_1 = R4 + 1 / (1j \* w \* C2)

imp\_2 = 1 / (1j \* w \* C1) + R2 + 1j \* w \* L2 + R3

imp\_parall = 1 / (1 / imp\_1 + 1 / imp\_2)

I\_1 = 1 / (R1 + 1j \* w \* L1 + imp\_parall)

U\_parall = I\_1 \* imp\_parall

I\_2 = U\_parall / imp\_1

U\_out = I\_2 \* R4

res = np.abs(U\_out/1);

print(res)