Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №3 по дисциплине

"Физические основы информационных технологий"

Название: Прохождение сигнала через четырёхполюсник

| Фамилия И.О.: | Герасименко Я.Д. |
| --- | --- |
| группа: | 1303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 22.11.23

Санкт-Петербург 2023

Условие задания

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”.

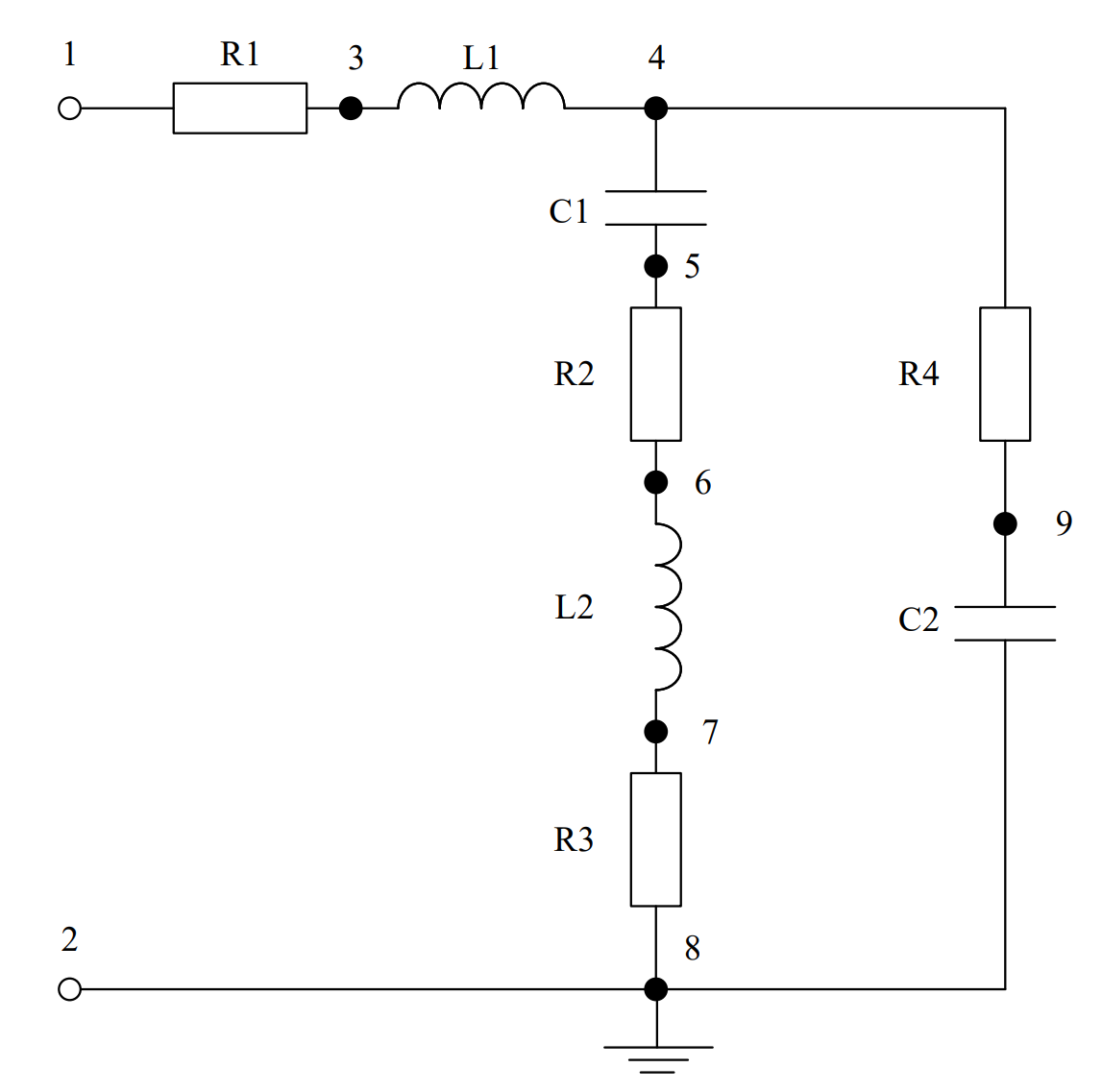


Рисунок 1 Исследуемый четырехполюсник

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

Исходные данные

Вариант 19

| L1,Гн | 13,58012711 |
| --- | --- |
| L2,Гн | 0,633743504 |
| С1,Ф | 1,1852E-05 |
| С2,Ф | 1,21571E-05 |
| R1,Ом | 112,1102388 |
| R2,Ом | 39,62672997 |
| R3,Ом | 1049,731549 |
| R4,Ом | 519,3809414 |
| Количество отсчетов N | 8192 |
| Время между соседними отсчетами (δt), c | 0,0196349540849362 |
| Контакты выхода | 5 и 6 |
| Номер гармоники | 2 |

Основные теоретические положения

Функция передачи системы H определяется как отношение выходного напряжения к входному, H =

Эта функция демонстрирует взаимосвязь между входным и выходным напряжениями в системе. Для определения входного и выходного напряжений необходимо учитывать импеданс Z, который представляет собой комплексное сопротивление для гармонических сигналов.

| Резистор | ZR | R |
| --- | --- | --- |
| Конденсатор | ZC |  |
| Катушка индуктивности | ZL | jωL |

Амплитудно-частотная характеристика (АЧХ) системы определяется как модуль H(jω), отражающий вариации амплитуды сигнала при различных частотах.

Фурье-преобразование – это процесс, который преобразует функцию времени в функцию, описывающую её разложение на базовые гармонические волны разной частотности.

Для создания спектра заданного последовательностью значений сигнала применяется дискретное Фурье-преобразование.

Fi=

Модуль F, или |F|, определяет амплитудно-частотную характеристику (АЧХ) спектра сигнала. АЧХ включает в себя шкалы для измерения амплитуды и частоты. В случае гармонических колебаний АЧХ будет изображена в виде одной линии, где её положение по оси ω указывает на соответствующую частоту этой гармоники в исходном сигнале.

**ПРИЛОЖЕНИЕ А**

**ПРОГРАММА MAIN.PY**

import numpy as np

from matplotlib import pyplot as plt

# Функция расчета коэффициента передачи цепи

def calculate\_response(frequency, input\_volt):

# Вычисляем отношение выходного напряжения к входному

return calculate\_output\_voltage(frequency, input\_volt) / input\_volt

# Функция для расчета выходного напряжения

def calculate\_output\_voltage(frequency, input\_volt):

# Выходное напряжение равно произведению тока на сопротивление R2

return calculate\_current(frequency, input\_volt) \* R2

# Функция для расчета тока в цепи

def calculate\_current(frequency, input\_volt):

# Ток равен напряжению на импедансе Z2

return calculate\_voltage(frequency, input\_volt) / calculate\_impedance\_Z2(frequency)

# Функция для расчета напряжения на цепи

def calculate\_voltage(frequency, input\_volt):

# Напряжение равно произведению входного тока на полный импеданс

return calculate\_input\_current(frequency, input\_volt) \* total\_impedance(frequency)

# Функция для расчета входного тока

def calculate\_input\_current(frequency, input\_volt):

# Входной ток равен входному напряжению деленному на сумму сопротивлений и импедансов

return input\_volt / (R1 + 1j \* frequency \* L1 + total\_impedance(frequency))

# Функция для расчета полного импеданса цепи

def total\_impedance(frequency):

# Полный импеданс равен сумме импедансов Z1 и Z2

return 1 / (1 / impedance\_Z1(frequency) + 1 / calculate\_impedance\_Z2(frequency))

# Функция для расчета импеданса Z1

def impedance\_Z1(frequency):

# Использование np.where для обработки массивов частот

impedance = R4 + 1 / (1j \* frequency \* C2)

return np.where(frequency == 0, np.inf, impedance)

# Функция для расчета импеданса Z2

def calculate\_impedance\_Z2(frequency):

# Использование np.where для обработки массивов частот

impedance = 1 / (1j \* frequency \* C1) + R2 + 1j \* frequency \* L2 + R3

return np.where(frequency == 0, np.inf, impedance)

L1 = 13.58012711 # Индуктивность L1, Гн

L2 = 0.633743504 # Индуктивность L2, Гн

C1 = 1.1852E-05 # Емкость C1, Ф

C2 = 1.21571E-05 # Емкость C2, Ф

R1 = 112.1102388 # Сопротивление R1, Ом

R2 = 39.62672997 # Сопротивление R2, Ом

R3 = 1049.731549 # Сопротивление R3, Ом

R4 = 519.3809414 # Сопротивление R4, Ом

N1 = 8192 # Количество элементов

# Интервал между соседними моментами времени

dt = 0.0196349540849362

# Время для всего набора данных

t = dt \* N1

def create\_plot(x\_data, y\_data, axis\_labels, plot\_title, legend\_title):

plt.figure(figsize=(10, 5)) # Установка размера графика

plt.plot(x\_data, y\_data, label=legend\_title, color='blue', linewidth=2) # Настройка линии графика

plt.xlabel(axis\_labels[0])

plt.ylabel(axis\_labels[1])

plt.title(plot\_title)

plt.grid(True) # Добавление сетки

plt.legend()

plt.tight\_layout() # Улучшенное размещение элементов на графике

plt.show()

# Построение графика АЧХ

frequency\_range = np.linspace(0, 100, 1000)

response\_values = np.abs(calculate\_response(frequency\_range, 10))

create\_plot(frequency\_range, response\_values, ['Частота (Гц)', 'Амплитуда'], 'АЧХ цепи', 'Отклик')

# Загрузка и отображение входного сигнала

input\_signal = np.loadtxt("./20.txt")

sampling\_times = np.linspace(0, t, N1)

create\_plot(sampling\_times, input\_signal, ['Время (с)', 'Амплитуда сигнала'], 'Входной сигнал', 'Сигнал')

# Преобразование Фурье и отображение спектра

fft\_signal = np.fft.fft(input\_signal)

freqs = np.fft.fftfreq(N1, dt) \* 2 \* np.pi

create\_plot(freqs, np.abs(fft\_signal), ['Частота (рад/с)', 'Амплитуда'], 'Спектр сигнала', 'Спектр')

# Анализ гармоники на частоте 10 Гц

harmonic\_freq = 20

response\_at\_harmonic = np.abs(calculate\_response(harmonic\_freq, 10))

print("Отклик на частоте 10 Гц:", response\_at\_harmonic)