Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №3 по дисциплине

"Физические основы информационных технологий"

Название: Прохождение сигнала через четырёхполюсник

|  |  |
| --- | --- |
| Фамилия И.О.: | Смирнов Д.Ю. |
| 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”.

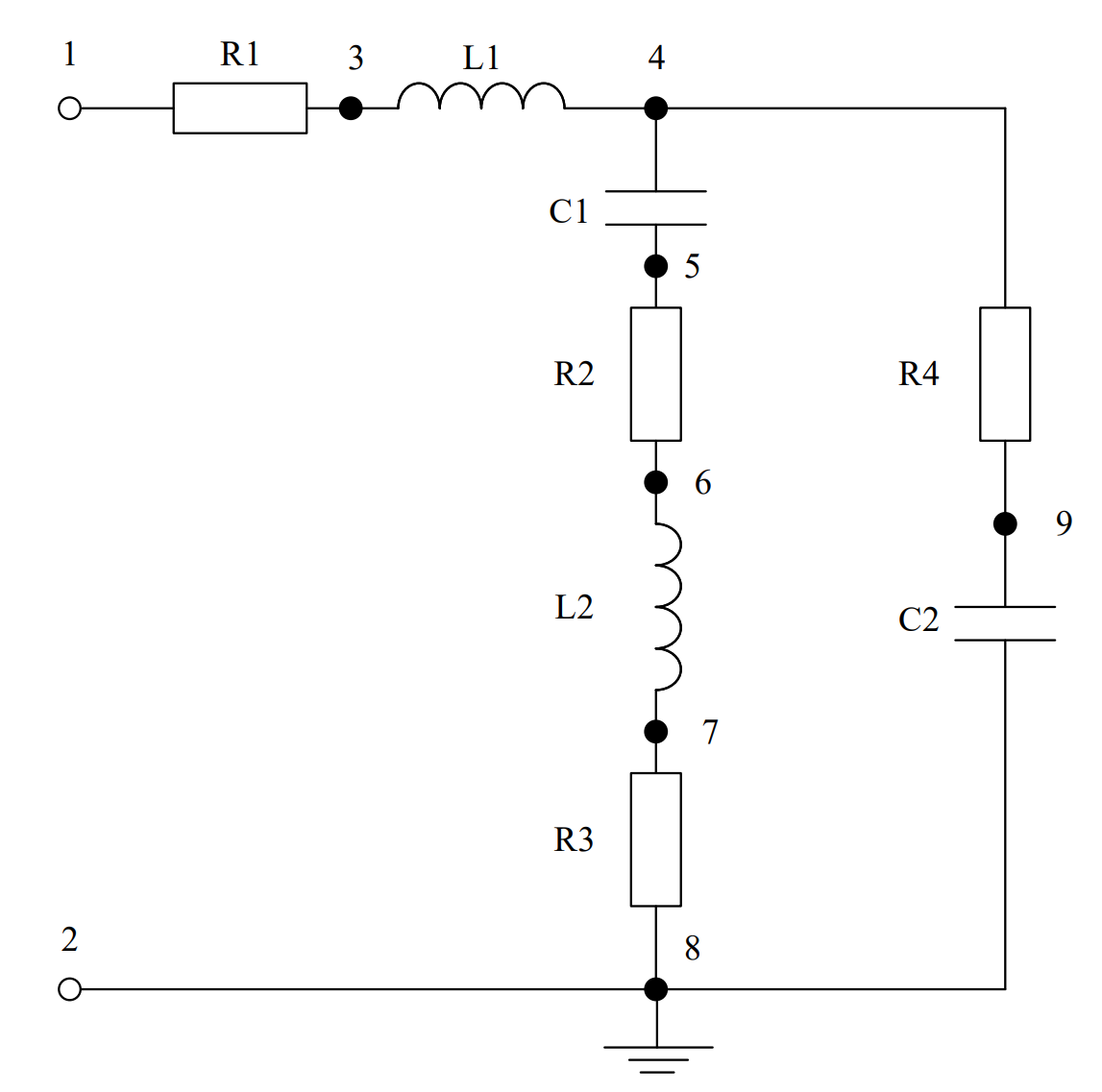


Рисунок 1. Исследуемый четырехполюсник

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

Исходные данные:

Вариант 25

L1: 12.9613676535114 (Гн)

L2: 0.496169770763215 (Гн)

С1: 1.15709335234433E-05 (Ф)

C2: 1.16514071253152E-05 (Ф)

R1: 100.244891037914 (Ом)

R2: 37.7326519768152 (Ом)

R3: 1067.73817189554 (Ом)

R4: 536.657874938589 (Ом)

N: 8192

dt: 0.0196349540849362

Контакты выхода: 4 и 9

Номер гармоники: 4

Теоретические положения

Передаточная функция системы H =

Она показывает связь между входным и выходным напряжениями.

Для нахождения входного и выходного напряжения нужно знать импеданс Z – комплексное сопротивление для гармонического сигнала:

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

**ПРИЛОЖЕНИЕ А**

Файл main.py:

import numpy as np

import matplotlib.pyplot as plt

# Consts

def L1():

return 12.9613676535114

def L2():

return 0.496169770763215

def C1():

return 1.15709335234433E-05

def C2():

return 1.16514071253152E-05

def R1():

return 100.244891037914

def R2():

return 37.7326519768152

def R3():

return 1067.73817189554

def R4():

return 536.657874938589

def N():

return 8192

def dt():

return 0.0196349540849362

# Functions

def capacitor(omega, C):

cap = (1j \* omega \* C)

if cap == 0:

return np.inf

return 1 / cap

def inductor(omega, L):

return 1j \* omega \* L

def leftBranch(omega):

return capacitor(omega, C1()) + R2() + inductor(omega, L2()) + R3()

def rightBranch(omega):

return R4() + capacitor(omega, C2())

def parallel(omega):

divisor = 1/leftBranch(omega) + 1/rightBranch(omega)

if divisor == 0:

return np.inf

return 1 / divisor

def findCircuitI(omega, Uin):

return Uin / (R1() + inductor(omega, L1()) + parallel(omega))

def findUparallel(omega, Uin):

return findCircuitI(omega, Uin) \* parallel(omega)

def findInRightBranchI(omega, Uin):

return findUparallel(omega, Uin) / rightBranch(omega)

def Uout(omega, Uin):

return findInRightBranchI(omega, Uin) \* R4()

def result(omega, Uin):

return Uout(omega, Uin) / Uin

def main():

t = N() \* dt()

omega = np.linspace(0, 100, 1000)

h = [np.abs(result(omg, 1)) for omg in omega]

plt.plot(omega, h)

plt.suptitle('АЧХ')

plt.show()

signal = np.loadtxt('25.txt')

plt.plot(np.arange(0, t, dt()), signal)

plt.suptitle('Сигнал')

plt.show()

singalFourier = np.fft.fft(signal)

df = 1/t

plt.plot(2 \* np.pi \* df \* np.arange(N()//2), np.abs(singalFourier[:singalFourier.size//2]) \* 2 / singalFourier.size)

plt.suptitle('Спектр')

plt.show()

print(np.abs(result(40, 1)))

return

if \_\_name\_\_ == '\_\_main\_\_':

main()