

Санкт-Петербургский Государственный Электротехнический Университет
"ЛЭТИ"

кафедра физики

Задание №3 по дисциплине

"Физические основы информационных технологий"

Название: Прохождение сигнала через четырёхполюсник

Фамилия И.О.: Поршнев Р.А.

группа: 1304

Преподаватель: Альтмарк А.М.

Итоговый балл:

Крайний срок сдачи: 22.11.23

Условие задания 3

Сигнал поступает на вход четырехполюсника (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:

$$s[t] = \sum_{k=1}^4 Amp_k \cos[\omega_k t].$$

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 t_i . Интервал между соседними моментами времени также дается в задании и обозначается как $\delta t = t_{i+1} - t_i$. 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 ω 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”.

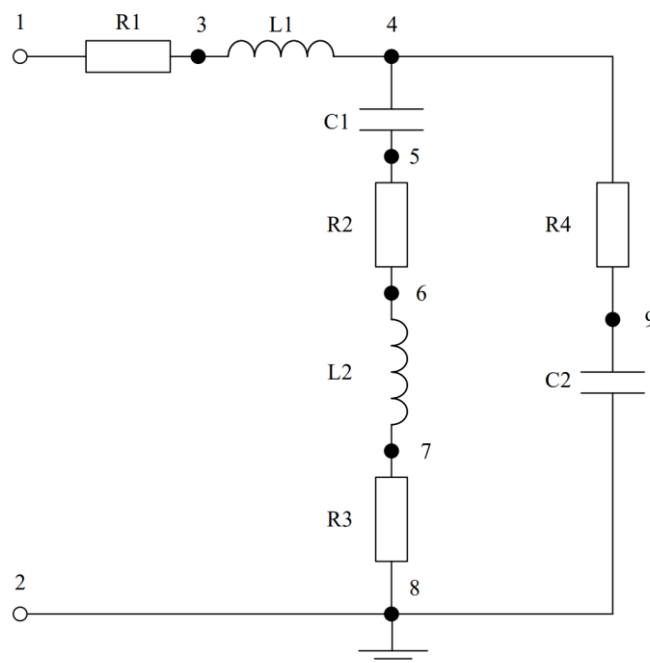


Рисунок 1. Исследуемый четырехполюсник

Отношение амплитуды k -й гармоники на выходе к амплитуде k -й гармоники на входе ($Amp_k^{Вых}/Amp_k^{Вх}$) 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

Вариант 16.

$L1 = 13.2111694517397 \text{ Гн}$

$L2 = 0.524511360054314 \text{ Гн}$

$C1 = 1.02992617837037\text{E-}05 \text{ Ф}$

$C2 = 1.43453621996884\text{E-}05 \text{ Ф}$

$R1 = 100.793558967849 \text{ Ом}$

$R2 = 35.96448954979 \text{ Ом}$

$R3 = 1054.70791442856 \text{ Ом}$

$R4 = 538.353431816832 \text{ Ом}$

$dt = 0.0196349540849362 \text{ с}$

Контакты выхода: 4 и 9

Номер гармоники 1.

Основные теоретические положения.

Допустим, имеется некоторый сигнал следующего вида:

$s_{\text{вход}}[t] = \sum_{k=1}^4 Amp_k \cos[\omega_k t]$ в виде набора N точек таких, что $2^n = N$, где n – натуральное число, при этом значение сигнала для каждой точки измерялось каждые dt секунд. Тогда для того, чтобы узнать АЧХ данного сигнала, к нему можно применить Дискретное преобразование Фурье, которое выглядит следующим образом:

$$F_k^{\text{вход}} = \sum_{n=0}^{N-1} s_n e^{-i \frac{2\pi}{N} nk}, \text{ где } i = \sqrt{-1}, k = 0..N-1.$$

Тогда по определению: $A_{\text{вход}}(\omega) = |F_i|$.

После данного преобразование получается набор точек по оси ординат, которым можно сопоставить точки по оси абсцисс следующим образом:

$$\omega_k = 2\pi \frac{k}{dt \cdot N}.$$

Таким образом получена зависимость амплитуды A от циклической частоты ω .

Следующим шагом можно определить передаточную характеристику следующим образом: $H(i\omega) = \frac{U_{\text{выход}}}{U_{\text{вход}}}$, где $U_{\text{вход}} = I(R_1 + Z_{L1} + R')$, $R' =$

$$\frac{(R_4 + Z_{C2})(Z_{C1} + R_2 + Z_{L2} + R_3)}{R_4 + Z_{C2} + Z_{C1} + R_2 + Z_{L2} + R_3}, Z_{Cn} = \frac{1}{i\omega C_n}, Z_{Ln} = i\omega L_n, n = \{1, 2\}.$$

Для нахождения $U_{\text{выход}}$ нужно выполнить следующие преобразования:

$I_{\text{левая}}(Z_{C1} + R_2 + Z_{L2} + R_3) = I_{\text{правая}}(R_4 + Z_{C2})$, где $I_{\text{левая}}$ и $I_{\text{правая}}$ – силы тока в левом и правом участке разветвления цепи соответственно.

Так как $I_{\text{левая}} + I_{\text{правая}} = I$, следовательно, $I_{\text{правая}} = I \frac{Z_{C1} + R_2 + Z_{L2} + R_3}{R_4 + Z_{C2} + Z_{C1} + R_2 + Z_{L2} + R_3}$.

Тогда $U_{\text{выход}} = I_{\text{правая}} R_4$.

$$\text{Следовательно, } H(i\omega) = \frac{I R_4 \frac{Z_{C1} + R_2 + Z_{L2} + R_3}{R_4 + Z_{C2} + Z_{C1} + R_2 + Z_{L2} + R_3}}{I(R_1 + Z_{L1} + R')} = \frac{R_4 \frac{Z_{C1} + R_2 + Z_{L2} + R_3}{R_4 + Z_{C2} + Z_{C1} + R_2 + Z_{L2} + R_3}}{(R_1 + Z_{L1} + R')}.$$

Для того, чтобы узнать во сколько раз изменилась амплитуда некоторой гармоники, достаточно взять модуль от передаточной характеристики.

Выполнение работы.

1. С помощью функции `read_file()` были считаны данные в виде значений сигнала и занесены в список.
2. Для отрисовки сигнала была произведена разметка временной шкалы в виде списка `time`.
3. Выполнено ДПФ для входного сигнала с помощью библиотеки `numpy`.
4. Для всех значений спектра высчитан модуль для последующей отрисовки графика зависимости амплитуды входного сигнала от циклической частоты.
5. Графически найдена циклическая частота для первой гармоники.
6. Рассчитано $H(i\omega)$, где ω – циклическая частота первой гармоники в промежутке от 0 до 100 рад/с с помощью функции `calculate_h(omega)`.

7. С помощью функций *draw_signal()*, *draw_amplitude()*, *draw_afr()* отрисована зависимость сигнала от времени, зависимость амплитуды сигнала от циклической частоты и график АЧХ.
8. Вывод передаточной характеристики для первой гармоники в консоль.

ПРИЛОЖЕНИЕ А. ИСХОДНЫЙ КОД ПРОГРАММЫ MAIN.PY

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

def read_file():
    f = open('16.txt', 'r')
    data = []
    while len(data) < N:
        data.append(float(f.readline()))
    f.close()
    return data

def draw_signal():
    plt.title('s_вход(t)')
    plt.xlabel('t, c')
    plt.ylabel('s_вход')
    plt.plot(time, signal)
    plt.show()

def draw_amplitude():
    plt.title('A_вход( $\omega$ )')
    plt.xlabel('ω, рад/с')
    plt.ylabel('A_вход')
    plt.plot(frequency[:stop_i], spectre_module[:stop_i])
    plt.show()

def calculate_h(omega):
    Z_C1 = 1 / (j * omega * C1)
    Z_C2 = 1 / (j * omega * C2)
    Z_L1 = j * omega * L1
    Z_L2 = j * omega * L2
    r_output = R4 * (Z_C1 + R2 + Z_L2 + R3) / (Z_C1 + R2 + Z_L2 + R3 +
R4 + Z_C2)
    r_parallel = ((R4 + Z_C2) * (Z_C1 + R2 + Z_L2 + R3)) / (R4 + Z_C2
+ Z_C1 + R2 + Z_L2 + R3)
    r_input = R1 + Z_L1 + r_parallel
    return r_output / r_input

def calculate_stop_i():
    stop_i = 0
    for i in range(N):
        if 2 * np.pi * df * i > 100 and stop_i == 0:
            stop_i = i + 1
            frequency.append(2 * np.pi * df * i)
    return stop_i

def draw_afr():
    plt.title('|H(iω)|')
    plt.xlabel('ω, рад/с')
    plt.ylabel('|H|')
    plt.plot(frequency[1:stop_i], H[:stop_i-1])
    plt.show()

if __name__ == '__main__':
    j = complex(0, 1)
```

```
L1 = 13.2111694517397
L2 = 0.524511360054314
C1 = 1.02992617837037E-05
C2 = 1.43453621996884E-05
R1 = 100.793558967849
R2 = 35.96448954979
R3 = 1054.70791442856
R4 = 538.353431816832
N = 8192
dt = 0.0196349540849362
df = 1 / (dt * N)
signal = read_file()
time = [dt * i for i in range(N)]
spectre = np.fft.fft(signal)
spectre_module = [abs(number) for number in spectre]
frequency = []
stop_i = calculate_stop_i()
H = [abs(calculate_h(omega)) for omega in frequency[1:]]
H_ans = calculate_h(10)      # у первой гармоники  $\omega = 10$  рад/с
draw_signal()
draw_amplitude()
draw_afr()
print(abs(H_ans))
```