Санкт-Петербургский Государственный Электротехнический Университет

«ЛЭТИ»

кафедра физики

Задание №3 по дисциплине

«Физические основы информационных технологий» Название: Прохождение сигнала через четырёхполюсник.

Фамилия И.О.: Токун Г.С. группа: 1303

Преподователь: Альтмарк А.М. Итоговый балл:

Крайник срок сдачи: 22.11.2023

Санкт-Петербург 2023

# Задание

Сигнал поступает на вход четырёхполюсника (контакты 1 и 2), показанно- го 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*] =

Σ4

*k*=1

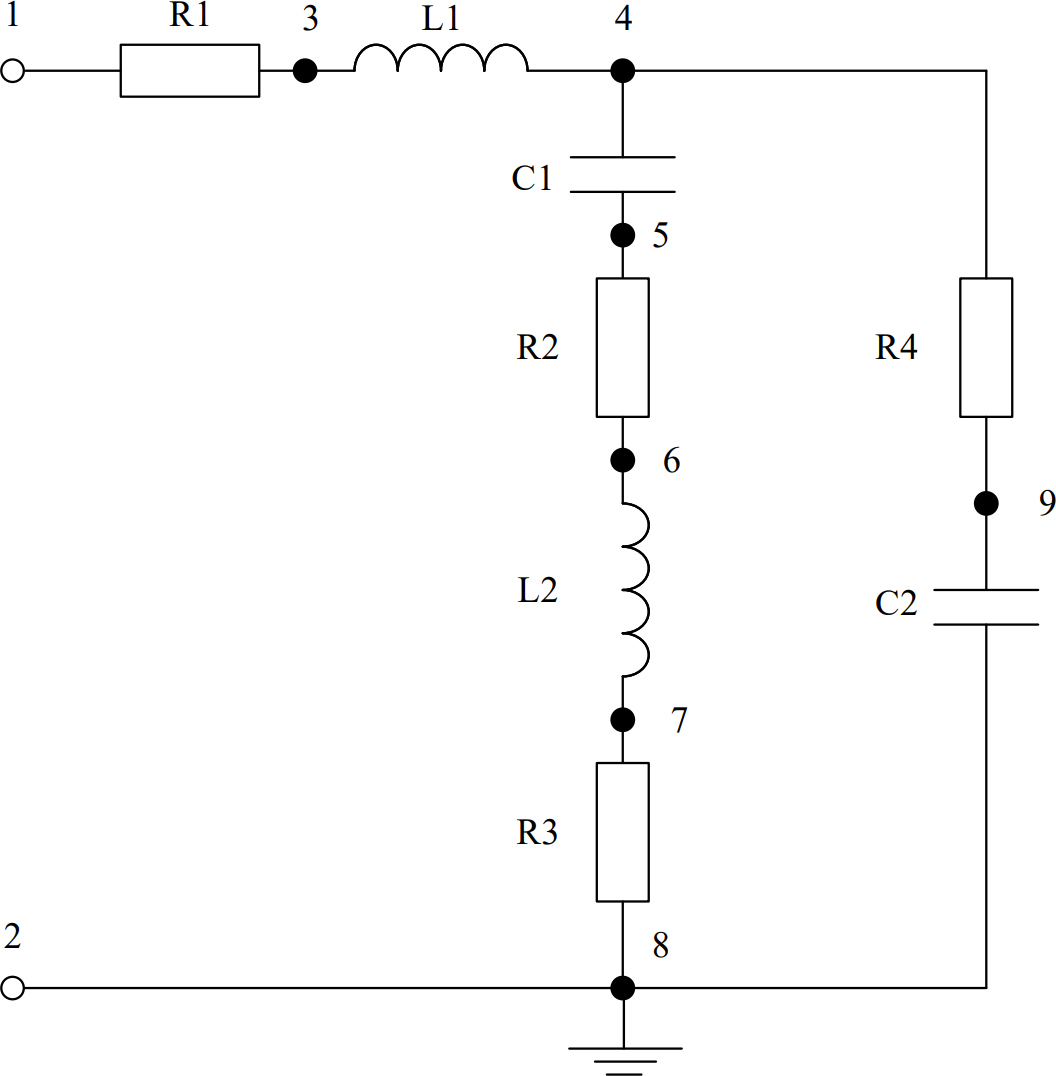
*Ampkcos*[*ωkt*]

Определить во сколько раз уменьшится амплитуда *k*-й гармоники на выхо- де четырехполюсника по сравнению с входным значением. В данном задании сигнал задается в виде массива из N элементов, который нужно взять из тек- стового файла. Каждый элемент массива соответствует определённому моменту времени *ti*. Интервал между соседними моментами времени также дается в за- дании 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 сиг- нала. 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 контак- тов четырехполюсника, массивы со входным сигналом можно найти в файле “FOIT\_IDZ3.xlsx”.

Отношение амплитуды k-й гармоники на выходе к амплитуде *k*-й гармонике на входе (*Ampk/Ampk*) нужно вычислить и записать в файл IDZ3.txt в пап- ке 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-12

Рис. 1: Исследуемый четырехполюсник

L1 = 13,2094490761272 Гн L2 = 0,834706894606612 Гн

C1 = 1,16365869528265E-05 Ф C2 = 1,300478410357E-05 Ф R1 = 114,535253054028 Ом R2 = 32,1696621442138 Ом R3 = 1052,22025365489 Ом R4 = 529,813173240116 Ом

N = 8192

*δt* = 0,0196349540849362

Контакты выхода - 7 и 8 Номер гармоники - 4

# Основные теоретические положения

Дискретное преобразование Фурье (ДПФ) — это метод анализа последова- тельности дискретных данных в частотной области. Если у вас есть последова- тельность чисел *x*0*, x*1*, ...xN−*1, ДПФ переводит ее из временной области в ча- стотную.

Допустим, у вас есть последовательность *xn* длины *N* , где *n* = 0*,* 1*,* 2*, ...N −* 1.

Тогда ДПФ определяется формулой:

*Xk* = Σ *N −* 1*n* = 0*xn*

*· e−*2*πikn*

*N*

В физике, особенно в контексте волновой теории и колебаний, термин "гармо- ника"относится к составляющим частям сложного колебательного или волново- го движения. Когда волна или колебание представлены как сумма нескольких частот или частотных компонент, каждая из этих частотных компонент назы- вается гармоникой. Гармоники являются кратными основной частоте и опреде- ляют структуру сложного сигнала.

Предположим, что у вас есть колебание или волна с основной частотой *f*1. Тогда гармоники будут иметь частоты *f*2 = 2*f*1*f*3 = 3*f*1*, f* 4 = 4*f*1 и так далее. Таким образом, *n −* гармоника имеет *fn* = *nf*1.

АЧХ (амплитудно-частотная характеристика) - это график, который отоб- ражает зависимость амплитуды сигнала от его частоты. АЧХ широко исполь- зуется в области сигнальной обработки, электроники, аудиотехники и других дисциплин.

Формально, АЧХ определяется как отношение амплитуды выходного сигнала к амплитуде входного сигнала в зависимости от частоты. Это измерение может предоставить информацию о том, как система реагирует на различные частоты входного сигнала.

АЧХ может быть представлена в виде графика, где по оси x отложены часто- ты, а по оси y - амплитуды. График позволяет визуально оценить, как систе- ма или устройство реагирует на различные частоты, выявляя преимущества и недостатки в передаче различных частот.

# Выполнение работы

Программа на Python успешно провела анализ сигнала, начиная с чтения данных из файла и построения графика зависимости сигнала от времени. С при- менением ДПФ был вычислен спектр сигнала, позволяя построить зависимость амплитуды от циклической частоты. Графически определена циклическая ча- стота первой гармоники.

С использованием расчетной функции ‘calculate\_h(omega)‘, программа эф- фективно определила передаточную функцию *H*(*iω*) для каждой циклической частоты в заданном диапазоне. Затем были построены графики зависимости сигнала от времени, амплитуды сигнала от циклической частоты, и график АЧХ.

В конце программы выведено значение передаточной функции для первой гармоники в консоль. Этот вывод позволяет оценить результаты анализа сигна- ла и полученные параметры системы.

# Приложение А

import matplotlib.pyplot as plt  
import numpy as np  
  
L1 = 13.2094490761272  
L2 = .834706894606612  
C1 = 1.16365869528265E-05  
C2 = 1.300478410357E-05  
R1 = 114.535253054028  
R2 = 32.1696621442138  
R3 = 1052.22025365489  
R4 = 529.813173240116  
N = 8192  
dt = 0.0196349540849362  
  
  
def read\_file():  
 with open('12.txt', 'r') as f:  
 data = [float(line) for line in f.readlines()[:N]]  
 return data  
  
def draw\_signal():  
 plt.title('s\_вход(t)')  
 plt.xlabel('t, c')  
 plt.ylabel('s\_вход')  
 plt.plot(time, signal, color='red')  
 plt.show()  
  
def draw\_amplitude():  
 plt.title('A\_вход(ω)')  
 plt.xlabel('ω, рад/c')  
 plt.ylabel('A\_вход')  
 plt.plot(frequency[:stop\_i], spectre\_module[:stop\_i], color='red')  
 plt.show()  
  
def calculate\_h(omega):  
 Z\_C1 = 1 / (1j \* omega \* C1)  
 Z\_C2 = 1 / (1j \* omega \* C2)  
 Z\_L1 = 1j \* omega \* L1  
 Z\_L2 = 1j \* omega \* L2  
 r\_output = R3 \* (R4 + Z\_C2) / (R4 + Z\_C2 + Z\_C1 + R2 + Z\_L2 + R3)  
 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 = next((i + 1 for i, val in enumerate(frequency) if 2 \* np.pi \* df \* i > 100), 0)  
 return stop\_i  
  
def draw\_afr():  
 plt.title('|H(iω)|')  
 plt.xlabel('ω, рад/c')  
 plt.ylabel('|H|')  
 plt.plot(frequency[1:stop\_i], H[:stop\_i-1], color='red')  
 plt.show()  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 j = 1j  
 df = 1 / (dt \* N)  
 signal = read\_file()  
 time = np.arange(0, dt \* N, dt)  
 spectre = np.fft.fft(signal)  
 spectre\_module = np.abs(spectre)   
 frequency = 2 \* np.pi \* np.arange(N) \* df   
 stop\_i = calculate\_stop\_i()  
 H = np.abs([calculate\_h(omega) for omega in frequency[1:]])  
 H\_ans = calculate\_h(40)   
 draw\_signal()  
 draw\_amplitude()  
 draw\_afr()  
 print(abs(H\_ans))