Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Фильтрация звукового сигнала.

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

Крайний срок сдачи: 05.12.23

.

Санкт-Петербург 2023

**Условие задания**

На входе приемника получен звуковой сигнал в двоичном коде (рис.1.). Необходимо перевести двоичный код в десятичный и затем провести над аналоговым сигналом процедуру фильтрации от высокочастотных помех. Для фильтрации необходимо использовать пассивные фильтры (фильтры без дополнительного источника питания), которые могут в себя включать, резисторы, конденсаторы и катушки индуктивности.

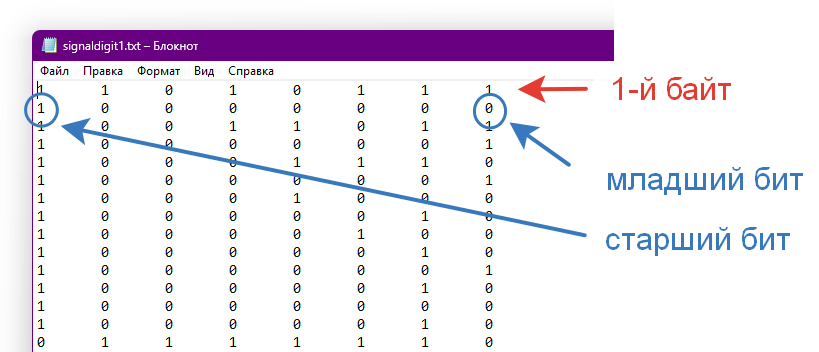


Рис.1. Структура данных в текстовом файле с сигналом

Исходные данные нужно взять в файле FOIT\_IDZ4.xlsx. В отчет нужно включить график сигнала во временной области и его спектр, схему фильтра и АЧХ его передаточной функции, спектр фильтрованного сигнала, а также график выходного сигнала во временной области. Файл IDZ4.txt должен содержать ответ на вопрос, который записан в звуковом сигнале.

Помимо текстового файла IDZ4.txt в папке IDZ4 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 IDZ4.txt:

25

**Исходные данные**

Вариант 9:

|  |  |
| --- | --- |
| длительность сигнала, с | Файл с сигналом |
| 3.5 | signaldigit9.txt |

**Теоретические сведения**

Для получения частотной характеристики аналогового сигнала необходимо применить быстрое преобразование Фурье к аналоговому сигналу, на основе полученных данных можно построить спектр сигнала.

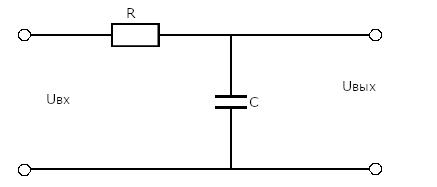
Для фильтрации сигнала от помех использовался фильтр низких частот.

Рис. 2 - Схема ФНЧ

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Файл **idz4.py**

import numpy as np  
import matplotlib.pyplot as plt  
from scipy.io.wavfile import write  
  
  
def write\_file(filename, signal):  
 write(filename, 44100, np.array(signal).astype(np.float32))  
  
  
def filter(omega, Uin):  
 R = 5000  
 C = 0.000001  
  
 def capacitor(omega, C):  
 cap = (1j \* omega \* C)  
 if cap == 0:  
 return np.inf  
 return 1 / cap  
  
 return (Uin \* capacitor(omega, C)) / (R + capacitor(omega, C))  
  
  
def main():  
 t = 3.5  
 digitalSignal = np.loadtxt("signaldigit9.txt", dtype=int)  
 signal = []  
  
 for row in digitalSignal:  
 signal.append(int(''.join(map(str, row)), 2))  
  
 dt = 3.5 / len(signal)  
  
 timeLine = np.arange(0, t, dt)  
 plt.plot(timeLine, signal)  
 plt.suptitle("Сигнал")  
 plt.show()  
  
 signalFourier = np.fft.fft(signal)  
 frequencies = [(i + 1) \* 2 \* np.pi / t for i in range(signalFourier.size)]  
 plt.plot(frequencies[1:len(frequencies) // 2], np.abs(signalFourier[1:signalFourier.size // 2]))  
 plt.suptitle('Исходный спектр')  
 plt.show()  
  
 hAbs = [abs(filter(frequencies[i], 1)) for i in range(signalFourier.size)]  
 plt.plot(frequencies[:500], hAbs[:500])  
 plt.suptitle('АЧХ')  
 plt.show()  
  
 singalFourierFiltered = []  
 for i in range(signalFourier.size - 1):  
 singalFourierFiltered.append(signalFourier[i + 1] \* hAbs[i + 1])  
  
 plt.plot(  
 frequencies[:len(singalFourierFiltered) // 2],  
 list(map(abs, singalFourierFiltered))[:len(singalFourierFiltered) // 2]  
 )  
 plt.suptitle("Спектр через фильтр")  
 plt.show()  
  
 filteredSignal = np.fft.ifft(singalFourierFiltered)  
 filteredSignal = [filteredSignal[i].real for i in range(filteredSignal.size)]  
 plt.plot(timeLine[1:], filteredSignal)  
 plt.suptitle('Сигнал через фильтр')  
 plt.show()  
  
 write\_file("filtered\_signal.wav", filteredSignal)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()