Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Фильтрация звукового сигнала.

|  |  |
| --- | --- |
| Фамилия И.О.: | Хулап. О. А. |
| 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.

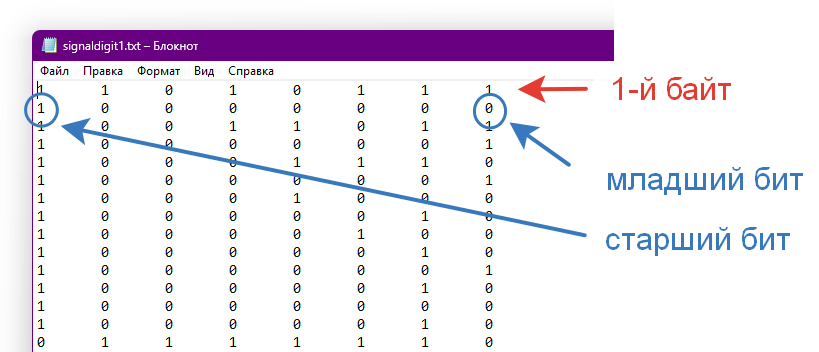


Рис.1. Структура данных в текстовом файле с сигналом

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 звуком 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 (11 вариант)**

|  |  |
| --- | --- |
| длительность сигнала, с | Файл с сигналом |
| 3.5 | signaldigit14.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.

Фильтры могут использоваться для удаления высокочастотных помех из аналогового сигнала. Фильтры могут включать в себя резисторы (R), конденсаторы (C) и индуктивности (L).

Одним из таких фильтров является фильтр Баттерворта, его схема имеет следующий вид:

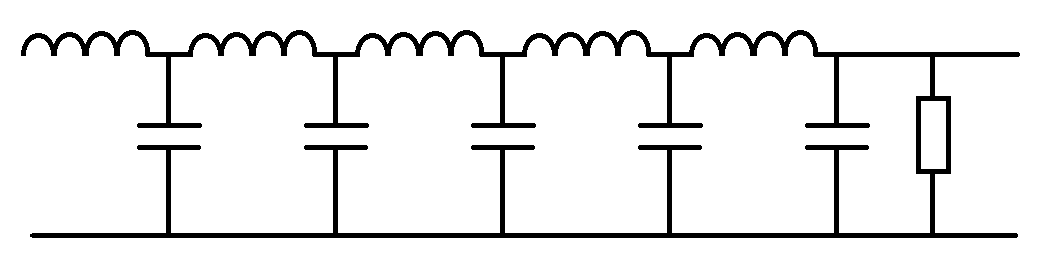


Рис.2 Схема фильтра

После задания параметров для данного фильтра и применения функции, описывающей передаточную функцию данного фильтра, можно найти спектр отфильтрованного сигнала.

Далее нужно произвести обратное преобразование Фурье отфильтрованного спектра и получить уже временной сигнал и в результате, с использованием специальной библиотеки, есть возможность произвести очищенный от помех сигнал.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД ПРОГРАММЫ**

Файл **main.py**

import numpy as np

import matplotlib.pyplot as plt

from scipy.io.wavfile import write

def writeToFile(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():

# Consts

t = 3.5

digitalSignal = np.loadtxt("signaldigit14.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()

singalFourier = np.fft.fft(signal)

frequences = [(i + 1) \* 2 \* np.pi/t for i in range(singalFourier.size)]

plt.plot(frequences[1:len(frequences)//2], np.abs(singalFourier[1:singalFourier.size//2]))

plt.suptitle('Спектр до фильтра')

plt.show()

hAbs = [abs(filter(frequences[i],1)) for i in range(singalFourier.size)]

plt.plot(frequences[:500], hAbs[:500])

plt.suptitle('АЧХ')

plt.show()

singalFourierAfterFilter = []

for i in range(singalFourier.size-1):

singalFourierAfterFilter.append(singalFourier[i + 1] \* hAbs[i+1])

plt.plot(

frequences[:len(singalFourierAfterFilter)//2],

list(map(abs,singalFourierAfterFilter))[:len(singalFourierAfterFilter)//2]

)

plt.suptitle("Спектр после фильтра")

plt.show()

signalAfterFilter = np.fft.ifft(singalFourierAfterFilter)

signalAfterFilter = [signalAfterFilter[i].real for i in range(signalAfterFilter.size)]

plt.plot(timeLine[1:], signalAfterFilter)

plt.suptitle('Сигнал после фильтра')

plt.show()

writeToFile("out.wav", signalAfterFilter)

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

main()