Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Фильтрация звукового сигнала

|  |  |
| --- | --- |
| Фамилия И.О.: | Гирман А.В. |
| группа: | 1303 |
| Преподаватель: | Альтмарк А.М. |
| Итоговый балл: |  |
|  |  |

Крайний срок сдачи: 5.12.23

Санкт-Петербург 2023

Условие задания 4

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.

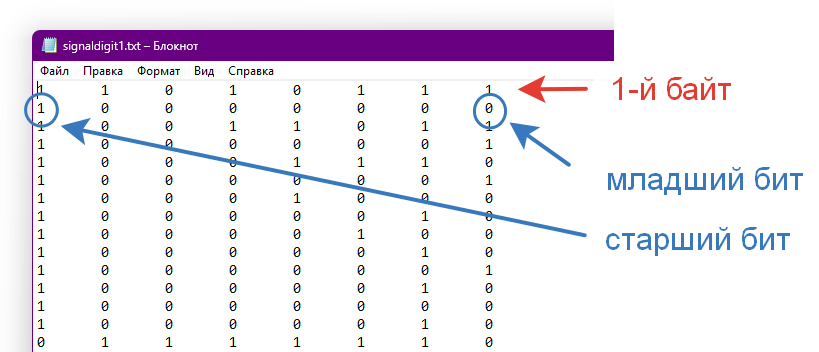


Рис.1. Структура данных в текстовом файле с сигналом

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

Вариант 22

Длительность сигнала 3.75с

Файл с сигналом: signaldigit22.txt

Теоретические положения

Для отчистки сигнала от помех используются фильтры высоких и низких частот.

В данной работе используется фильтр низких частот RC.

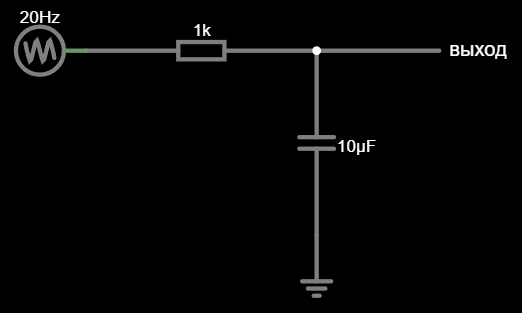


Рисунок 1. ФНЧ RC

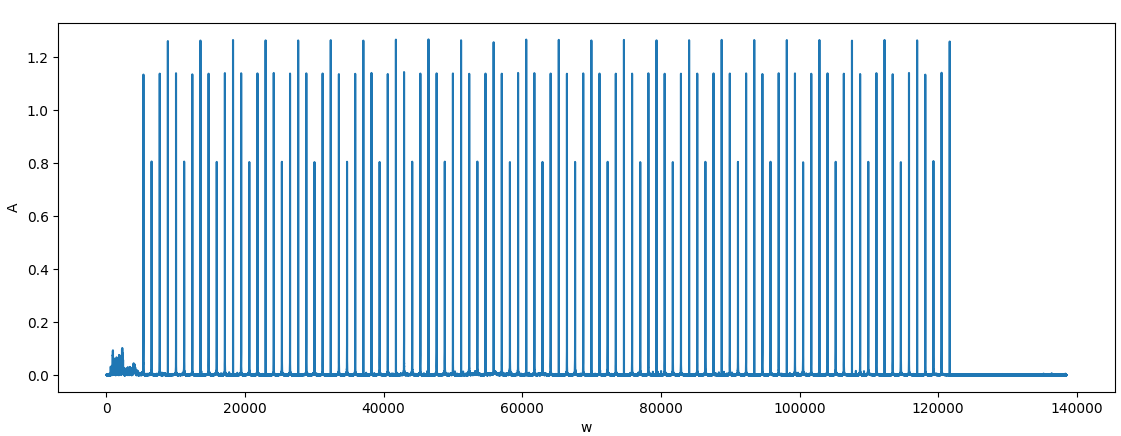


Рисунок 2. Спектр оригинального сигнала

После фильтрации сигнала получаем уже другой спектр.

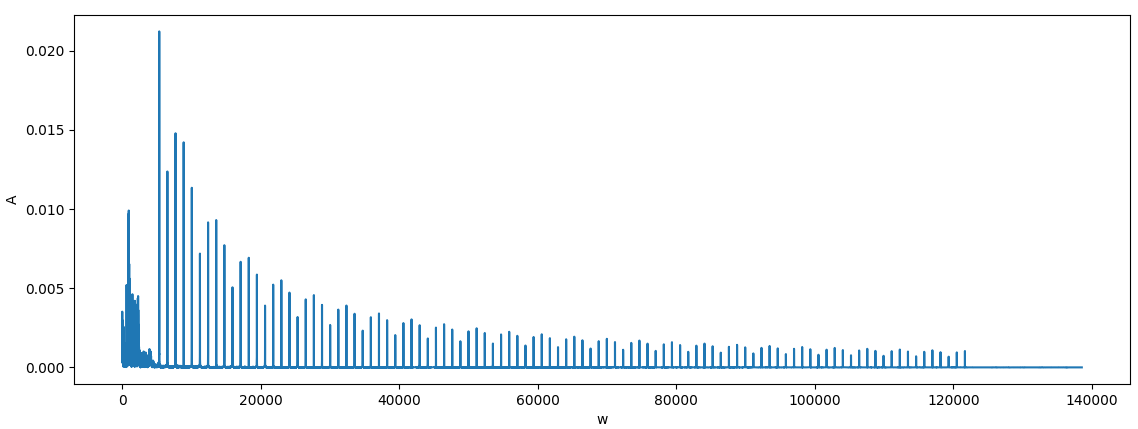


Рисунок 3. Спектр отфильтрованного сигнала

**ПРИЛОЖЕНИЕ А**

**ПРОГРАММА IDZ4.py**

import numpy as np

import matplotlib.pyplot as plt

from scipy.io.wavfile import write

C = 0.00001 # 10 мкФ

R = 1000 # 1 кОм

U = 5 # 5 В

def ZC(w, C):

if 1j \* w \* C == 0:

return np.inf

return 1 / (1j \* w \* C)

def R\_in(w):

if ZC(w, C) == np.inf:

return np.inf

return R + ZC(w, C)

def I\_in(w, U\_in):

return U\_in / R\_in(w)

# фильтр низких частот RC

def H(w, U\_in):

U\_out = I\_in(w, U\_in) \* ZC(w, C)

return U\_out / U\_in

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

# Аналоговый сигнал

analogue\_signal = []

with open('signaldigit22.txt', 'r') as file:

for line in file:

bits\_string = line.strip().split('\t')

bits = ''.join(bits\_string)

analogue\_signal.append(int(bits, 2))

scaled = np.float32(np.real(analogue\_signal) / np.max(np.abs(np.real(analogue\_signal))) \* 1.0)

# Запись звукового сигнала в файл

write('original22.wav', 44100, scaled)

N = len(analogue\_signal)

T = 3.75

dt = T / N

# Сигнал на входе

time\_array = np.arange(0, T, dt)

plt.plot(time\_array, analogue\_signal)

plt.xlabel('t')

plt.show()

# Спектр на входе

F = np.fft.fft(analogue\_signal)

A = [np.abs(i) \* 2 / N for i in F[1:]]

w = [i \* 2 \* np.pi / T for i in range(N)]

plt.plot(w[:int(len(w) / 2)], A[:int(len(A) / 2)])

plt.xlabel('w')

plt.ylabel('A')

plt.show()

# АЧХ

U = 5

H\_array = [np.abs(H(w\_elem, U)) for w\_elem in w]

plt.plot(w[:450], H\_array[:450])

plt.xlabel('Hz')

plt.ylabel('H')

plt.show()

# Выходной спектр

F\_new = []

for i in range(len(F)-1):

F\_new.append(F[i+1] \* H\_array[i+1])

A = [np.abs(i) \* 2 / N for i in F\_new]

plt.plot(w[:int(len(w) / 2)], A[:int(len(A) / 2)])

plt.xlabel('w')

plt.ylabel('A')

plt.show()

# Выходной сигнал

analogue\_signal\_out = np.fft.ifft(F\_new)

print(analogue\_signal\_out)

plt.plot(time\_array[1:], analogue\_signal\_out)

plt.xlabel('t')

plt.show()

scaled = np.float32(np.real(analogue\_signal\_out) / np.max(np.abs(np.real(analogue\_signal\_out))) \* 1.0)

# Запись выходного звукового сигнала в файл

write('filtered22.wav', 44100, scaled)