Санкт-Петербургский Государственный Электротехнический Университет "ЛЭТИ"

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Фильтрация звукового сигнала

|  |  |
| --- | --- |
| Фамилия И.О.: | Королева П.А |
| 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.

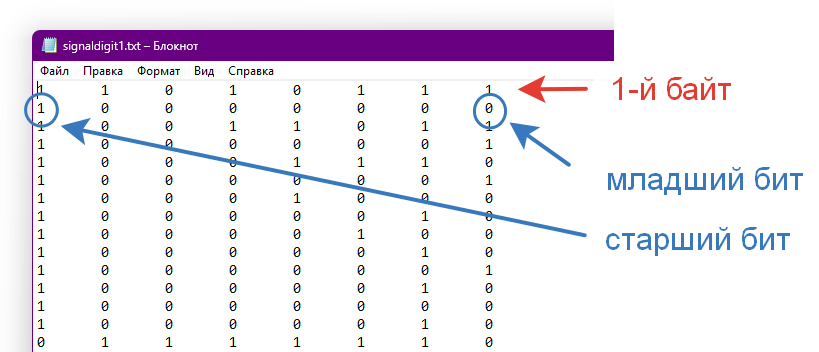


Рис.1. Структура данных в текстовом файле с сигналом

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”.

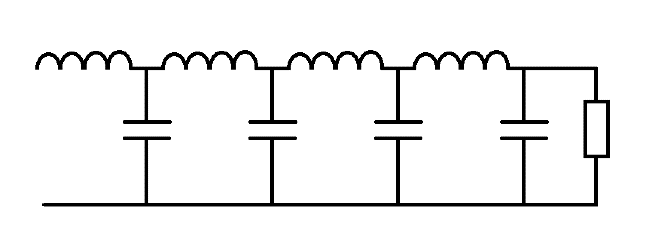
Пример содержания файла IDZ4.txt:

25

Вариант 21 Длительность сигнала 3.25с

Теоретические положения

Для отчистки сигнала от помех используются фильтры.

В данной работе используется фильтр Баттерворта.

АЧХ его передаточной функции имеет вид ступеньки, что позволяет достаточно точно обрезать высокие частоты шума, почти не ослабив основной сигнал. Схема фильтра представлена на рисунке.

**ПРИЛОЖЕНИЕ А**

**ПРОГРАММА MAIN.NB**

(\*считываем битовый сигнал, переводим в аналог\*)

t = 3.25;

discretSignal= ReadList["C:\\Users\\Полина\\Desktop\\учеба\\foit\\idz4\\signaldigit21.txt",{Number, Number, Number, Number, Number, Number, Number, Number}];

analogSignal = Table[FromDigits[discretSignal[[i]], 2], {i, 1, Length@discretSignal}]

dt = t/ Length@discretSignal;

ListPlay[analogSignal, SampleRate -> 44100]

(\*строю график сигнала\*)

ForPlot = Table[{(i - 1)\*dt, analogSignal[[i]]}, {i, 1, Length@discretSignal}];

ListPlot[ForPlot, Filling -> Axis, PlotRange -> Full]

(\*нахожу спектр\*)

Fsig = Fourier[analogSignal];

outN = Length@Fsig;

df = 1/t;

FourAbs = Table[{2 \[Pi] df (i - 1), Abs@Fsig[[i]]}, {i, 2, outN}];

ListPlot[FourAbs, Filling -> Axis, PlotRange -> Full]

(\*пытаюсь сделать фильтр Батерворта\*)

coef = 2.25;

R1 = 50;

C1 = 0.0000059\*coef;

C2 = 0.0000054\*coef;

C3 = 0.0000041\*coef;

C4 = 0.0000024\*coef;

C5 = 497.9 \* 10^-9\*coef;

L1 = 0.0124\*coef;

L2 = 0.0144\*coef;

L3 = 0.012\*coef;

L4 = 0.0083\*coef;

L5 = 0.0037\*coef;

Zpar5[w\_] = 1/(I w C5 + 1/R1);

Zpar4[w\_] = 1/(I w C4 + 1/(I w L5 + Zpar5[w]));

Zpar3[w\_] = 1/(I w C3 + 1/(I w L4 + Zpar4[w]));

Zpar2[w\_] = 1/(I w C2 + 1/(I w L3 + Zpar3[w]));

Zpar[w\_] = 1/(I w C1 + 1/(I w L2 + Zpar2[w])); (\*общий\*)

I1[w\_] = Uin/(I w L1 + Zpar[w]);

Upar[w\_] = I1[w]\*Zpar[w];

I2[w\_] = Upar[w]/(I w L2 + Zpar2[w]);

Upar2[w\_] = I2[w] \* Zpar2[w];

I3[w\_] = Upar2[w]/(I w L3 + Zpar3[w]);

Upar3[w\_] = I3[w] \* Zpar3[w];

I4[w\_] = Upar3[w]/(I w L4 + Zpar4[w]);

Upar4[w\_] = I4[w] \* Zpar4[w];

I5[w\_] = Upar4[w]/(I w L5 + Zpar5[w]);

Upar5[w\_] = I5[w] \* Zpar5[w];

Uout[w\_] = Upar5[w];

H[w\_] = Uout[w]/Uin;

Hlist = Table[Abs@H[i], {i, 1, outN}];

Fnew = Fsig\*Hlist;

Plot[Abs@H[w], {w, 1, outN/10}, PlotRange -> Full]

(\*смотрю преобразованный спектр\*)

FnewTable = Table[{2 \[Pi] df (i - 1), Abs@Fnew[[i]]}, {i, 2, outN}];

ListPlot[FnewTable, Filling -> Axis, PlotRange -> Full]

(\*получаю обратно сигнал\*)

ChangedSignal = InverseFourier[Fnew];

FilteredSignal = Table[{( i - 1)\*dt, Re@ChangedSignal[[i]]}, {i, 1, Length@discretSignal}];

ListPlot[FilteredSignal, Filling -> Axis, PlotRange -> Full]

ListPlay[Re@ChangedSignal, SampleRate -> 44100]