

кафедра физики

Задание №4 по дисциплине

"Физические основы информационных технологий"

Название: Фильтрация звукового сигнала

Фамилия И.О.:

Чубан Д.В.

группа:

1303

Преподаватель:

Альтмарк А.М.

Итоговый балл:

Крайний срок сдачи:

5.12.23

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.

signaldigit1.txt – Блокнот

Файл	Правка	Формат	Вид	Справка				
1	1	0	1	0	1	1	1	0
1	0	0	0	0	0	0	0	1
1	0	0	1	1	0	1	1	0
1	0	0	0	0	0	0	0	1
1	0	0	0	1	1	1	1	0
1	0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0
0	1	1	1	1	1	1	1	0

1-й байт

младший бит

старший бит

Рис.1. Структура данных в текстовом файле с сигналом

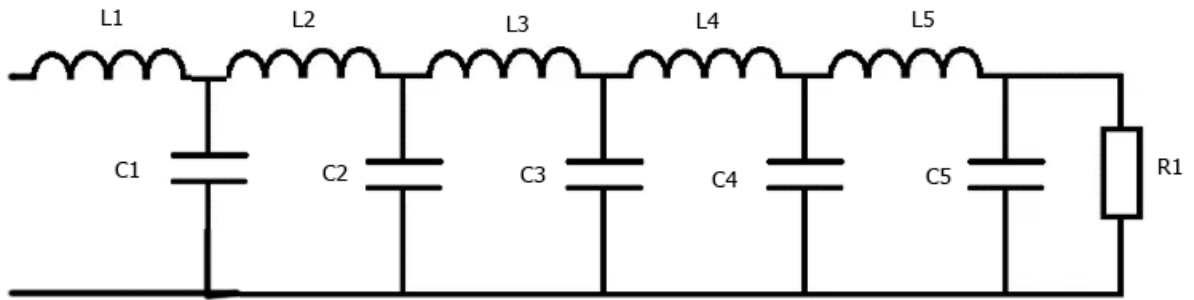
Исходные данные нужно взять в файле FOIT_IDZ4.xlsx. В отчет нужно включить график сигнала во временной области и его спектр, схему фильтра и АЧХ его передаточной функции, спектр фильтрованного сигнала, а также график выходного сигнала во временной области. Файл IDZ4.txt должен содержать ответ на вопрос, 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:

Теоретические положения

Для очистки 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.

Чтобы посчитать импеданс фильтра нужно:

1. Найти импеданс параллельного соединения $R1$, $C5$ (обозн. $Z_{\text{par}5}$)
2. Найти импеданс параллельного соединения $C4$ с последовательным соединением $L5$ и $Z_{\text{par}5}$ (обозн. $Z_{\text{par}4}$)
3. Найти импеданс параллельного соединения $C3$ с последовательным соединением $L4$ и $Z_{\text{par}4}$ (обозн. $Z_{\text{par}3}$)
4. Найти импеданс параллельного соединения $C2$ с последовательным соединением $L3$ и $Z_{\text{par}3}$ (обозн. $Z_{\text{par}2}$)
5. Найти импеданс параллельного соединения $C1$ с последовательным соединением $L2$ и $Z_{\text{par}2}$ (обозн. Z_{par}) – импеданс всего фильтра

График входного сигнала:

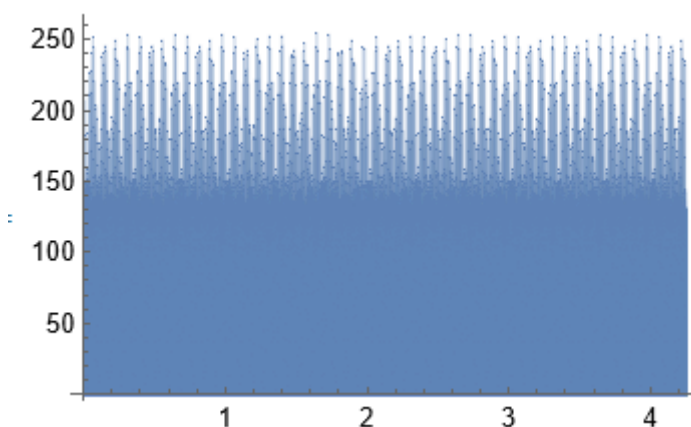


График спектра сигнала:

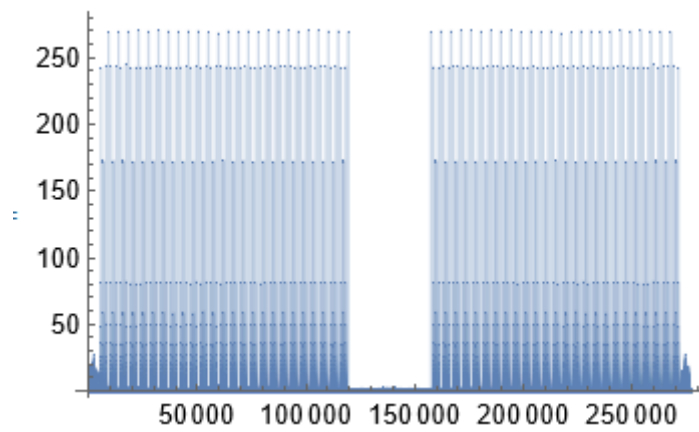


График АЧХ:

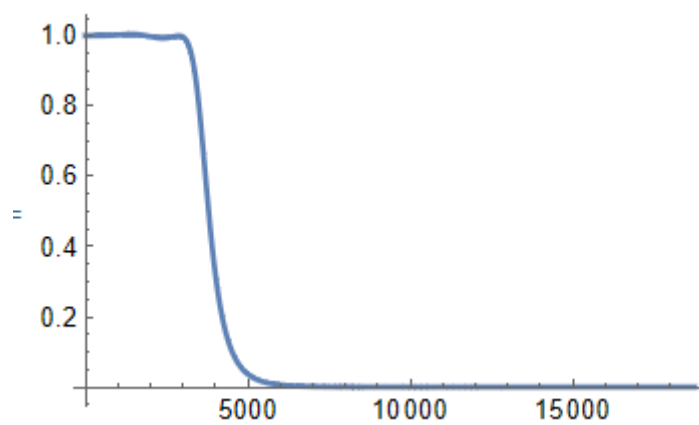
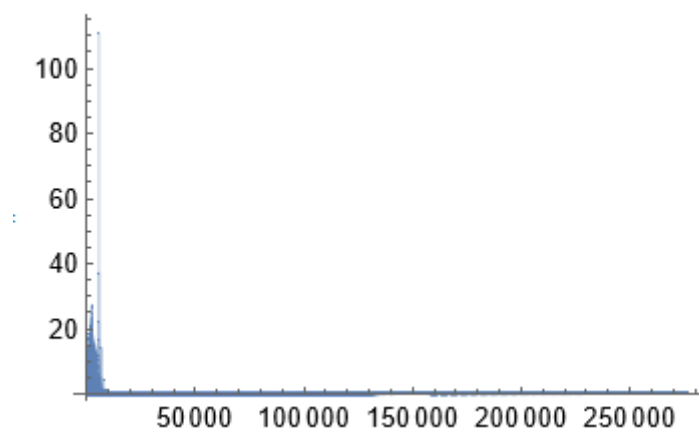
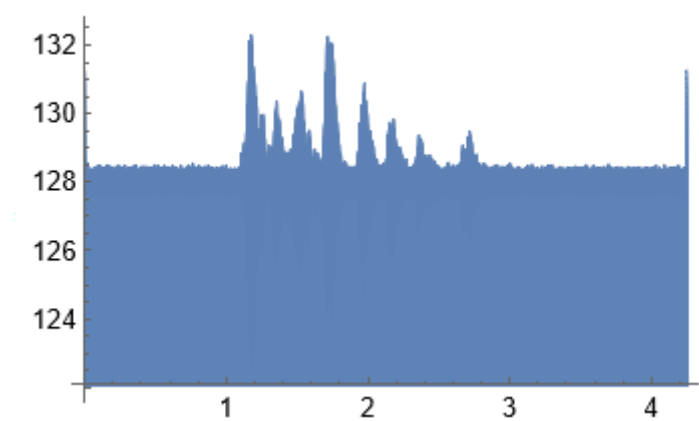


График спектра отфильтрованного сигнала:



Отфильтрованный сигнал:



ПРИЛОЖЕНИЕ А
ПРОГРАММА IDZ4.NB

```
t = 4.25;
discretSignal =
  Import["C:\\Users\\metro\\Downloads\\signaldigit16.txt", "Data"];
analogSignal =
  Table[FromDigits[discretSignal[[i]], 2], {i, 1,
    Length@discretSignal}];
dt = t/Length@discretSignal;
(*Входной сигнал*)
ListPlay[analogSignal, SampleRate -> 44100]

ForPlot =
  Table[{(i - 1)*dt, analogSignal[[i]]}, {i, 1,
    Length@discretSignal}];
ListPlot[ForPlot, Filling -> Axis, PlotRange -> Full]

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]
coef = 1.75;
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]
FourNew = Table[{2 \[Pi] df (i - 1), Abs@Fnew[[i]]}, {i, 2, outN}];
ListPlot[FourNew, Filling -> Axis, PlotRange -> Full]
filteredFourier = InverseFourier[Fnew];
FilteredSignal =
  Table[{(i - 1)*dt, Re@filteredFourier[[i]]}, {i, 1,
    Length@discretSignal}];
```

```
ListPlot[FilteredSignal, Filling -> Axis, PlotRange -> Full]  
ListPlay[Re@filteredFourier, SampleRate -> 44100]
```