МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 4

**Субоптимальные методы расчёта КИХ-фильтров.**

Выполнил студент 595 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Лаптев

Проверил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ П.Н. Уланов

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2020

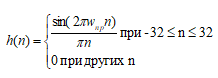
**Задание 1.**

Найти импульсную характеристику заданного идеального фильтра, построить её график (ограничив количество отсчётов заданным порядком фильтра).

**Решение:**

Импульсную характеристику фильтра можно рассчитать по формуле:

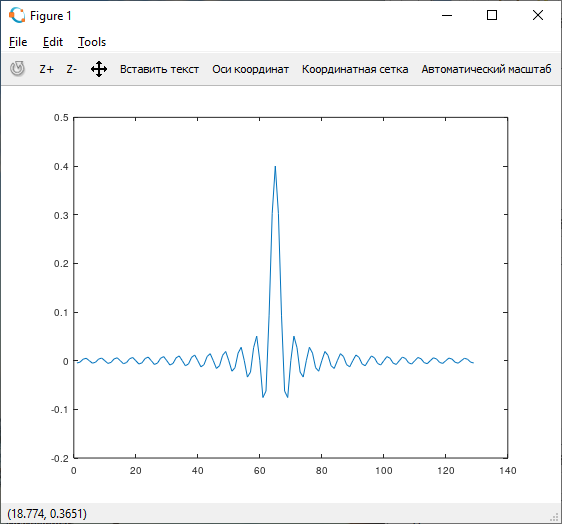


Ограничим количество отсчётов до конечного числа. Пусть их будет 128.

-64 ≤ n ≤ 64

При вычислении функции ИХФ напрямую может возникнуть неопределённость {0/0}

По свойству первого замечательного предела . 2w – значение функции в точке устранимого разрыва. Примем частоту пропускания w за 0.2 Гц.



*Рис.1. Импульсная характеристика идеального фильтра*

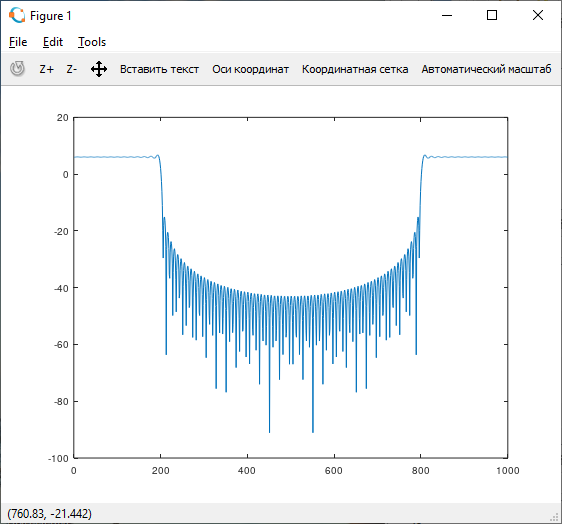
**Задание 2.**

Рассчитать АЧХ фильтра, полученного усечением импульсной характеристики и построить её график.

**Решение:**

Построим нерекурсивный фильтр, уравнение которого , где – ядро свёртки. Частоту возьмём за 0.2 Гц.

Для нахождения АЧХ подадим на вход программе единичный импульс и вычислим спектр отклика с помощью БПФ.



*Рис 2. АЧХ фильтра, полученного усечением импульсной характеристики*

**Задание 3.**

Найти коэффициенты фильтра, умножив импульсную характеристику на заданную «оконную» функцию.

**Решение:**

Для улучшения характеристики фильтра можно применить взвешивающее окно Хемминга, уравнение которого: .

Чтобы оптимизировать свойства фильтра нужно перемножить ядро свёртки и окно Хемминга между собой: . Рассчитаем новые коэффициенты и изобразим их графически.

Коэффициенты фильтра после применения окна Хемминга (от 1 до 129 слева направо):

-4.0700e-04 3.1020e-18 4.4344e-04 -5.0803e-18 -5.0598e-04 -7.4234e-22 5.9691e-04

-2.2417e-18 -7.1860e-04 5.4307e-18 8.7352e-04 3.7813e-18 -1.0643e-03 -5.7668e-19

1.2938e-03 -4.1852e-18 -1.5650e-03 1.0938e-17 1.8814e-03 -2.0165e-17 -2.2468e-03

-2.4027e-18 2.6655e-03 -7.0938e-18 -3.1425e-03 2.0019e-17 3.6840e-03 -8.7756e-18

-4.2972e-03 -6.8152e-18 4.9910e-03 -1.0525e-17 -5.7765e-03 3.3459e-17 6.6680e-03

-1.2274e-17 -7.6839e-03 -1.6168e-17 8.8486e-03 -1.3956e-17 -1.0195e-02 5.3655e-17

1.1770e-02 -1.5506e-17 -1.3637e-02 1.6212e-17 1.5895e-02 -1.6864e-17 -1.8692e-02

1.7455e-17 2.2269e-02 -1.7979e-17 -2.7043e-02 1.8432e-17 3.3806e-02 -1.8808e-17

-4.4250e-02 1.9104e-17 6.2784e-02 -1.9318e-17 -1.0558e-01 1.9447e-17 3.1813e-01

5.0000e-01 3.1813e-01 1.9447e-17 -1.0558e-01 -1.9318e-17 6.2784e-02 1.9104e-17

-4.4250e-02 -1.8808e-17 3.3806e-02 1.8432e-17 -2.7043e-02 -1.7979e-17 2.2269e-02

1.7455e-17 -1.8692e-02 -1.6864e-17 1.5895e-02 1.6212e-17 -1.3637e-02 -1.5506e-17

1.1770e-02 5.3655e-17 -1.0195e-02 -1.3956e-17 8.8486e-03 -1.6168e-17 -7.6839e-03

-1.2274e-17 6.6680e-03 3.3459e-17 -5.7765e-03 -1.0525e-17 4.9910e-03 -6.8152e-18

-4.2972e-03 -8.7756e-18 3.6840e-03 2.0019e-17 -3.1425e-03 -7.0938e-18 2.6655e-03

-2.4027e-18 -2.2468e-03 -2.0165e-17 1.8814e-03 1.0938e-17 -1.5650e-03 -4.1852e-18

1.2938e-03 -5.7668e-19 -1.0643e-03 3.7813e-18 8.7352e-04 5.4307e-18 -7.1860e-04

-2.2417e-18 5.9691e-04 -7.4234e-22 -5.0598e-04 -5.0803e-18 4.4344e-04 3.1020e-18

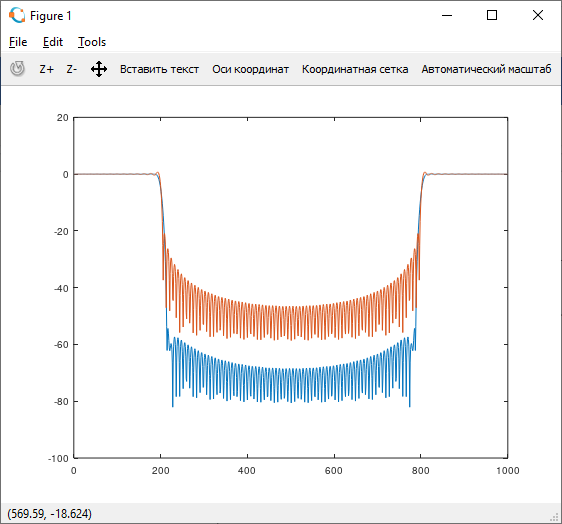
-4.0700e-04 -1.5592e-18 3.9448e-04

**Задание 4.**

Рассчитать АЧХ полученного фильтра. Построить графики АЧХ фильтра до взвешивания и после взвешивания в одних осях.

**Решение:**

Выполним все те же действия из задания 2, но уже для фильтра с окном Хэмминга.



*Рис. 4 АЧХ фильтра с окном (синий) и без него (красный)*

**Задание 5.**

Привести количественные результаты – как изменились характеристики фильтра (ширина полосы пропускания, пульсации АЧХ в полосах пропускания и задержки) после взвешивания?

**Решение:**

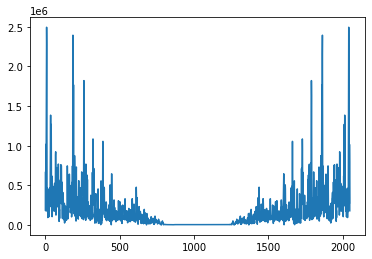
Взвешенный фильтр чуть лучше подавляет высокие частоты за пределами пропускания (амплитуды скачков практически не отличаются, они, лишь, несколько меньше, причём по мере увеличения кол-ва отсчётов эффективность подавления возрастает). При этом число пульсаций у взвешенного фильтра выше, чем у невзвешенного, но скачок амплитуды несколько меньше возле границы полосы пропускания. Наличие окна, кроме того, позволяет ещё немного уменьшить значения амплитуды на полосе пропускания.

**Задание 6.**

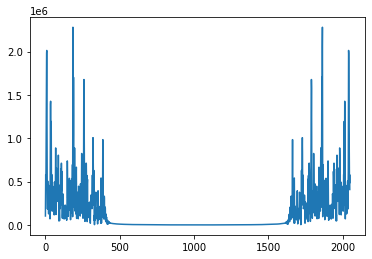
Проверить работу взвешенного фильтра как подавителя шума.

**Решение:**

Для решения возьмём фильтр с окном Хемминга. В качестве входного сигнала будет wav-файл. После пропускания через фильтр должно произойти подавление шума.



*Рис. 5 Входной сигнал*



*Рис. 6 Выходной сигнал, полученный после пропускания через фильтр*

На графиках видно, что максимальная амплитуда несколько уменьшилась (если бы сигнал содержал больше верхних частот, то изменения были бы более заметны), также полоса задержки (отрезок, где амплитуда близка к 0) на Рис. 6 заметно шире, чем подобный отрезок на Рис. 5.

**Вывод:** таким образом, применяя взвешивающие окна к реальным фильтрам можно добиться сглаживания графика спектра и уменьшения пульсаций полос, т.е. приблизить реальные фильтры к идеальным, улучшив свойства линейной системы.

**Приложение**

**Приложение к заданию 1**

w = 0.2;

N = 128;

for n = 1 : N + 1;

h(n) = 0;

if n == N / 2 + 1

h(n) = 2 \* w;

else

h(n) = sin(2 \* pi \* w \* (n - (N / 2 + 1))) / (pi \* (n - (N / 2 + 1)));

endif;

endfor;

plot(h);

**Приложение к заданию 2**

x = linspace(0, 0, 1000);

x(1, 1) = 1;

M = 1000;

w = 0.2;

N = 128;

y(m) = 0;

for m = 1 : M

for n = 1 : N + 1

if n == N / 2 + 1

h(n) = 2 \* w;

elseif

h(n) = (sin(2 \* pi \* w \* (n - (N / 2 + 1)))) / (pi \* (n - (N / 2 + 1)));

endif;

if (m - n) > 0

y(m) += x(m - n) \* h(n);

endif;

endfor;

endfor;

plot(20 \* log10(abs(fft(y))));

**Приложение к заданию 3**

w = 0.25;

L = 128;

for l = 1 : L + 1

if l - L / 2 == 0

b(l) = 2 \* w;

elseif

b(l) = sin(2 \* pi \* w \* (l - L / 2))/ (pi \* (l - L / 2)) \* (0.54 - 0.46\*cos(2 \* pi\*(l / L)));

endif;

endfor;

x = b(l);

**Приложение к заданию 4**

y = linspace(0, 0, 1000);

x = linspace(0,0,1000);

x(1, 1) = 1;

N = 512;

w = 0.2;

L = 128;

for l = 1 : L + 1

if l - L / 2 == 0

b(l) = 2 \* w;

elseif

b(l) = sin(2 \* pi \* w \* (l - L / 2))/ (pi \* (l - L / 2)) \* (0.54 - 0.46 \* cos(2 \* pi \* (l / L)));

endif;

endfor;

for n = 0 : N

for l = 1 : L + 1

if (n - l) > 0

y(n) = y(n) + x(n - l) \* b(l);

endif;

endfor;

endfor;

plot(20 \* log10(abs(fft(y))));

hold on

z = linspace(0, 0, 1000);

c = linspace(0,0,1000);

c(1, 1) = 1;

for d = 1 : L + 1

if d - L / 2 == 0

b(d) = 2 \* w;

elseif

b(d) = (sin(2 \* pi \* w \* (d - L / 2))) / (pi \* (d - L / 2));

endif;

endfor;

for n = 0 : N

for d = 1 : L + 1

if (n - d) > 0

z(n) = z(n) + c(n - d) \* b(d);

endif;

endfor;

endfor;

plot(20 \* log10(abs(fft(z))));

**Приложение к заданию 6**

import wave

import matplotlib.pyplot as plt

import numpy as np

import scipy as sp

import math

def split(arr,left,right):

l = len(arr)

for i in range(0,int(l\*0.5)):

lc.append(samples[2\*i])

rc.append(samples[2\*i+1])

def filters(lc):

M = len(lc)

w = 0.2

L = 128

y = []

h = []

for i in range (0, M):

y.append(0)

summa = 0

for l in range(0, L + 1):

h.append(0)

if l == L / 2:

h[l] = 2 \* w

else:

h[l] = math.sin(2 \* math.pi \* w \* (l - (L / 2))) / (math.pi \* (l - (L / 2))) \* (0.54 - 0.46 \* math.cos(2 \* math.pi \* (l / L)))

if (i - l) > 0:

summa += lc[i - l] \* h[l]

y[i] = summa

return(y)

fi = wave.open('test.wav', mode='rb')

qnt = fi.getnframes()

fr = fi.readframes(qnt)

samples = np.frombuffer(fr, dtype=np.int16)

lc = []

rc = []

split(samples,lc,rc)

spl = sp.fft.fft(lc)

plt.plot(abs(spl))

plt.show()

signal=filters(lc)

signa=sp.fft.fft(signal)

plt.plot(abs(signa))

plt.show()