МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт цифровых технологий, электроники и физики (ИЦТЭФ)

Кафедра вычислительной техники и электроники (ВТиЭ)

Лабораторная работа № 03

**Однородный КИХ фильтр**

Выполнил студент 595 гр.

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

Проверил:

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

Лабораторная работа защищена

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

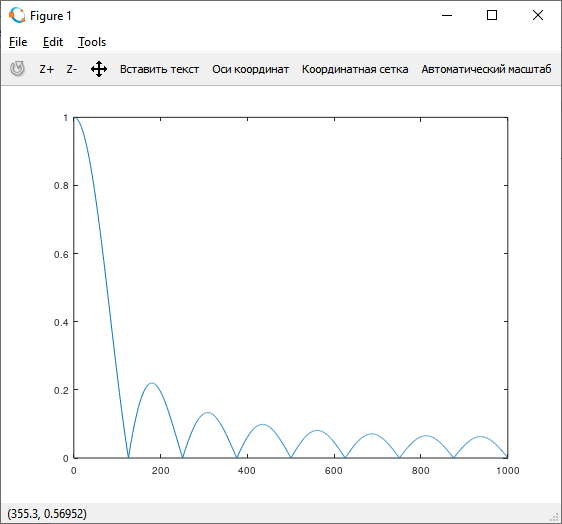
Барнаул 2020

**Задание 1.**

Постройте график 𝐴(𝑤)(0≤𝑤≤0.5), напишите программу, реализующую однородный фильтр, сравните АЧХ Вашего фильтра с рассчитанной теоретически.

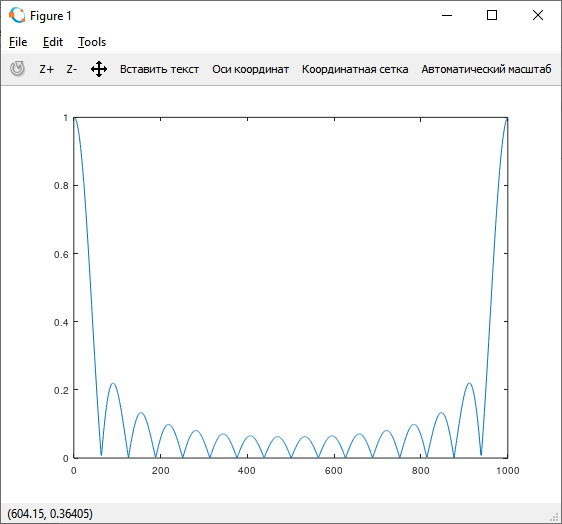
**Решение:**

Расчет теоретической АЧХ производим по формуле 𝐴(𝑤)= 1/𝐿(2𝜋𝑤𝑙-𝜋𝑤(𝐿−1)), где L - порядок фильтра, w - частота. Частота w будет принадлежать отрезку от 0Гц до 0.5Гц (0Гц ≤ w ≤ 0.5Гц ). Пусть порядок фильтра будет L = 16. Построим график.



*Рис.1 Модуль АЧХ однородного нерекурсивного фильтра, рассчитанный теоретически.*

Далее подаём на вход фильтра, реализованного по формуле 𝑦(𝑛)= 1/𝑁(𝑛−𝑙), сигнал, состоящий из одиночного импульса, все 1000 отсчётов сигнала равны 0, кроме первого, который равен 1, выходной сигнал обрабатываем с помощью ДПФ (fft) и построим график.



*Рис.2 Модуль экспериментальной АЧХ однородного нерекурсивного фильтра, на оси абсцисс номера гармоник.*

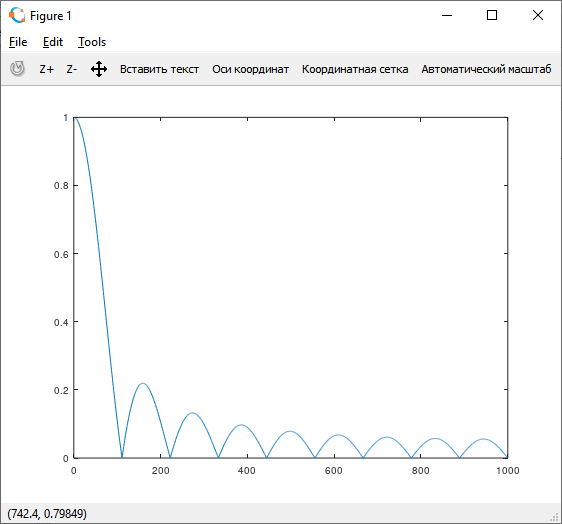
Таким образом, мы видим, что графики теоретически и экспериментально рассчитанных АЧХ одинаковые.

**Задание 2.**

Для другого числа L2=L1+2 создайте на основе нерекурсивного фильтра рекурсивный. Постройте график АЧХ, рассчитанной теоретически, а затем подайте на вход фильтра сигнал с равномерным спектром и сравните спектр выходного сигнала с АЧХ, рассчитанной теоретически.

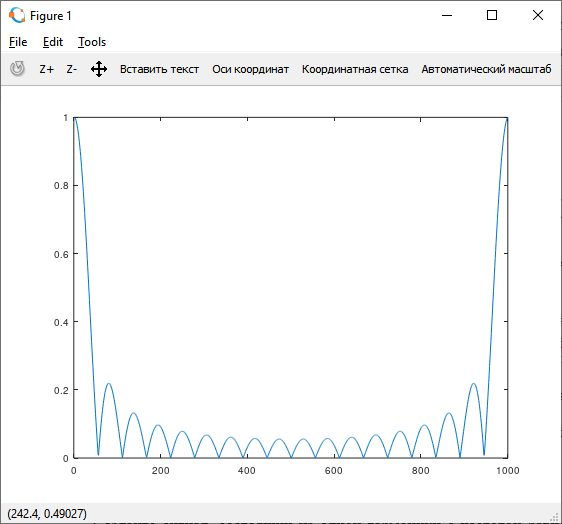
**Решение:**

Новый фильтр будет иметь восемнадцатый порядок (L2=16+2). Расчет теоретической АЧХ производим по формуле 𝐴(𝑤)=1/𝐿\*(sin(𝜋𝐿𝑤)/sin(𝜋𝑤)), w – частота(Гц), причем 0Гц ≤ w ≤ 0.5 Гц . Строим полученный модуль АЧХ.



*Рис.3 Модуль АЧХ однородного рекурсивного фильтра, рассчитанный теоретически.*

На основе нерекурсивного фильтра из первого задания 𝑦(𝑛)= 1/𝑁(𝑛-𝑙) можно построить линейный рекурсивный фильтр, уравнение которого *yn = yn-1 +* 1/ 𝑁*(xn - xn-N).* Подадим сигнал, состоящий из одиночного импульса, все 1000 отсчетов сигнала равны 0, кроме первого, который равен 1, выходной сигнал обрабатываем с помощью ДПФ (fft).



*Рис.4 Модуль экспериментальной АЧХ однородного рекурсивного фильтра.*

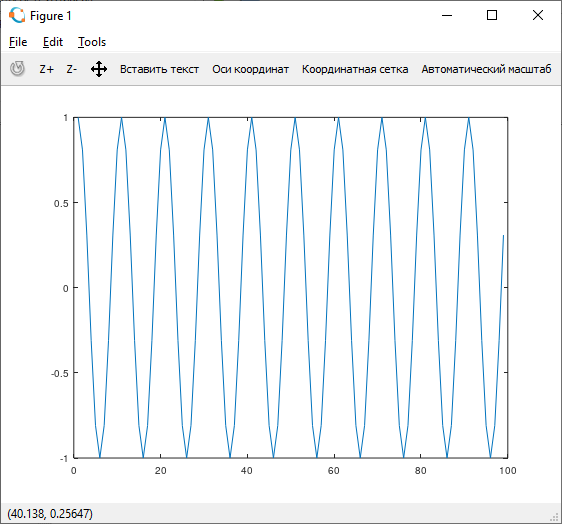
Таким образом, мы видим, что графики теоретически и экспериментально рассчитанных АЧХ также одинаковые.

**Задание 3.**

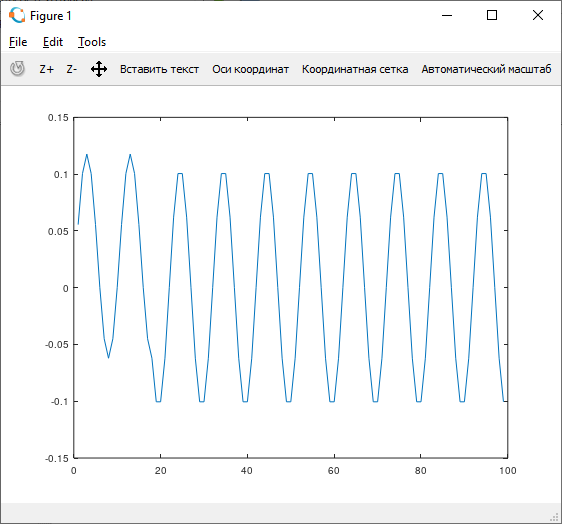
Создайте сигнал, состоящий из одной гармоники с частотой равной или близкой к той, которая полностью подавляется. Подайте этот сигнал на вход любого из Ваших фильтров. Постройте в одних и тех же осях графики входного и выходного сигналов

**Решение:**

Для решения данного пункта будем использовать фильтр 𝑦(𝑛)= 1/𝑁𝑛−1+(𝑥𝑛−𝑥𝑛−𝐿)/𝐿 – однородный рекурсивный фильтр восемнадцатого порядка. У него полностью подавляется частота W = 0.1 Гц). Подадим системе на вход сигнал cos(2𝜋𝑓𝑡), и посмотрим, как он изменится (рис. 5 - 6).



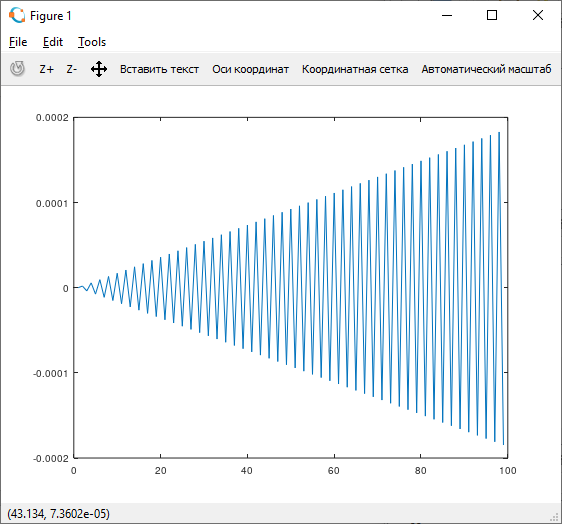
*Рис.5 Входной сигнал частотой 0.1 Гц*



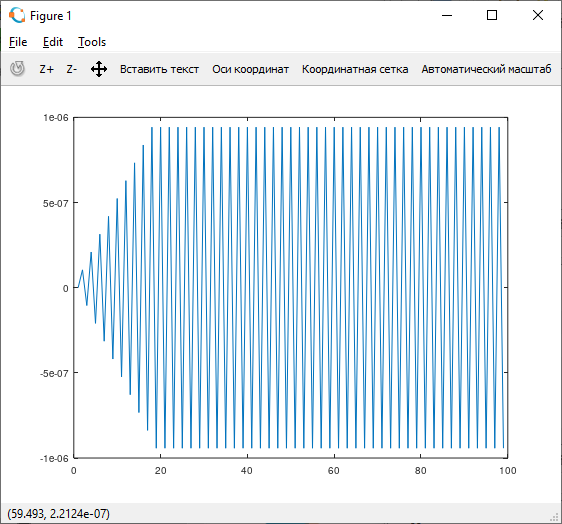
*Рис.6 Выходной сигнал частотой 0.1 Гц*

Таким образом, график отфильтрованного сигнала более прижат к оси абсцисс, чем исходный.

Далее подаём синусоидальный сигнал с единичной амплитудой sin(2𝜋𝑓𝑡) и подбираем такое занчение частоты, чтобы этот сигнал не проходил через фильтр, т.е. частота выходного сигнала не превышала 1е-6.



*Рис. 7 Входной синусоидальный сигнал с частотой 0.4999997 Гц*



*Рис. 8 Случай, при котором синусоидальный сигнал не проходит через фильтр (частота входного сигнала 0.4999997 Гц)*

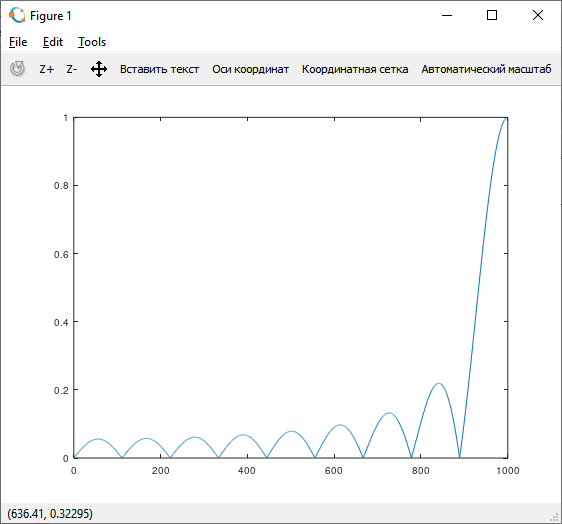
На Рис. 8 видно, что максимальная амплитуда при подобранном значении частоты не превышает значение 1е-6.

**Задание 4.**

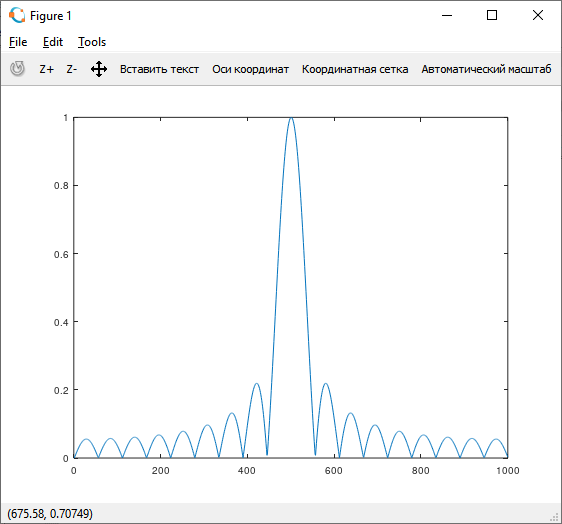
Создайте ФВЧ на основе одного из созданных однородных фильтров и постройте его АЧХ, подав на вход этого фильтра одиночный импульс

**Решение:**

Формула для нерекурсивного фильтра нижних частот 𝑌(𝑛)= 1/𝐿−1)𝑙∗(𝑥𝑛−𝑙) . Пусть w = w′ − 0.5: 𝐴(𝑤)=1/𝐿\*(sin(𝜋𝐿𝑤)/sin(𝜋𝑤)), таким образом, мы получили формулу для расчёта АЧХ нерекурсивного фильтра десятого порядка, выполняем расчёт АЧХ и строим его график.



*Рис. 9 График модуля теоретической АЧХ фильтра верхних частот.*



*Рис. 10 График модуля АЧХ фильтра высоких частот.*

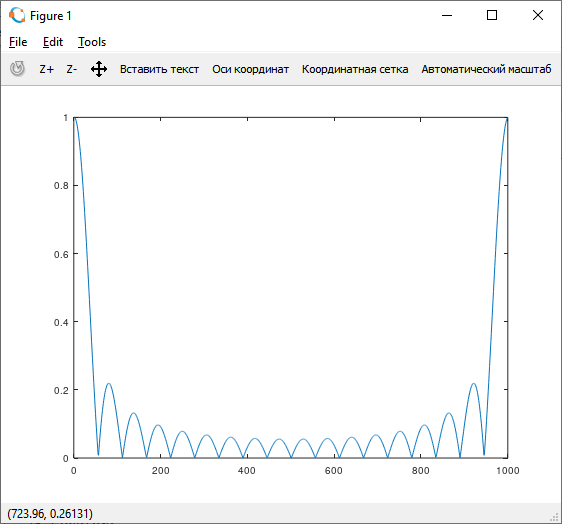
Таким образом, теоретическая АЧХ совпала с экспериментальной.

**Задание 5.**

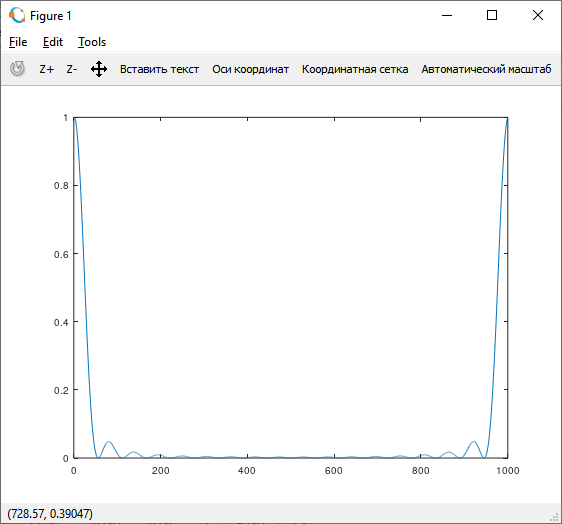
Исследуйте последовательное соединение фильтров. Для этого пропустите одиночный импульс дважды через один и тот же однородный фильтр (результатом будет триангулярный фильтр). Постройте импульсную характеристику триангулярного фильтра. Постройте в одних осях АЧХ триангулярного фильтра и однородного фильтра, из которого был создан триангулярный.

**Решение:**

Подключим 2 одинаковых однородных нерекурсивных фильтра 𝑌(𝑛)= 1/𝐿−1)𝑙∗(𝑥𝑛−𝑙). друг за другом, фильтруя второй системой выходной сигнал первой. Подадим такой системе единичный импульс. После этого мы сможем узнать АЧХ триангулярного фильтра и построить её график.



*Рис.11 Модуль АЧХ однородного фильтра.*



*Рис.12 Модуль АЧХ триангулярного фильтра*

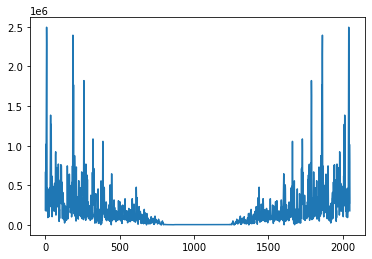
Таким образом, в результате последовательного соединения двух однородных фильтров, образуется триангулярный фильтр, имеющий более плавный график модуля АЧХ.

**Задание 6.**

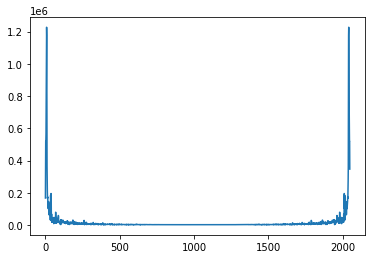
Проверьте работу однородного фильтра как подавителя шума.

**Решение:**

Для решения задачи возьмём рекурсивный фильтр (L = 131). В качестве входного сигнала будет wav-файл. После пропускания через фильтр должно произойти подавление шума.



*Рис. 13. Входной сигнал*



*Рис. 14. Выходной сигнал, полученный после пропускания через фильтр*

На графиках видно, что выходной сигнал имеет существенно меньше шума, чем входной сигнал (максимальная амплитуда уменьшилась примерно в 2 раза, а основная часть графика стала иметь более гладкие очертания).

**Вывод:**

В данной лабораторной работе мы познакомились с рекурсивными, не рекурсивными фильтрами и однородным фильтром. Научились использовать однородный фильтр как подавитель шума. Для построения графиков и реализации фильтра, я использовал программу Octave.

**Приложение**

**Приложения к пункту 1**

**1.1.** L = 16;

w = linspace(0, 0.5, 1000);

y = 0;

for l = 0:L-1

y += 1 / L \* cos((2 \* pi \* w \* l - pi \* w \* (L-1)));

endfor;

plot(abs(y));

**1.2.** L = 16;

N = 1000;

x = linspace(0, 0, N);

x(1, 1) = 1;

for n = 1 : N;

y(n) = 0;

for l = 0: L - 1;

if (n - l) > 0;

y(n) += x(n - l) / L;

endif;

endfor;

endfor;

plot(abs(fft(y)));

**Приложения к пункту 2**

**2.1.** L = 18;

for w = 1 : 1000;

y(w)= 1 / L \* sin(pi \* L \* w / 2000) / sin(pi \* w / 2000);

endfor;

plot(abs(y));

**2.2.** L = 18;

N = 1000;

x = linspace(0, 0, N);

x(1, 1) = 1;

for i = 1 : N;

if i==1;

y(i) = x(i) / L;

elseif (i > 1 && i <= L);

y(i) = y(i-1) + x(i) / L;

else;

y(i) = y(i-1) + (x(i) - x(i-L)) / L;

endif;

endfor;

plot(abs(fft(y)));

**Приложения к пункту 3**

**3.1.** L=18;

t=0:1:999;

x = cos(2\*pi\*0.1\*t);

plot(x(1.19:100));

for i = 1 : 1000;

if i==1;

y(i)=x(i)/L;

elseif (i > 1 && i <= L);

y(i)=y(i-1)+x(i)/L;

else;

y(i)=y(i-1)+(x(i)-x(i-L))/L;

endif;

endfor;

#plot(y(1.19:100));

**3.2.** L=18;

t=0:1:999;

x = sin(2\*pi\*0.4999997\*t);

plot(x(1.19:100));

for i = 1 : 1000;

if i==1;

y(i)=x(i)/L;

elseif (i > 1 && i <= L);

y(i)=y(i-1)+x(i)/L;

else;

y(i)=y(i-1)+(x(i)-x(i-L))/L;

endif;

endfor;

#plot(y(1.19:100));

**Приложения к пункту 4**

**4.1.** L =18;

for w = 1 : 1000;

a(w)= sin(pi \* L \* (w/2000-0.5))/sin( pi \* (w / 2000-0.5)) / L;

endfor;

plot(abs(a));

**4.2.** L = 18;

N = 1000;

x = linspace(0, 0, N);

x(1, 1) = 1;

for n = 1 : N;

y(n) = 0;

for l = 0: L - 1;

if (n - l) > 0;

y(n) += (-1)^l \* x(n - l) / L;

endif;

endfor;

endfor;

plot(abs(fft(y)));

**Приложения к пункту 5**

L = 18;

N = 1000;

x = linspace(0, 0, N);

x(1, 1) = 1;

for n = 1 : N;

if n == 1;

y(n) = x(n) / L;

elseif (n > 1 && n <= L);

y(n) = y(n - 1) + x(n) / L;

else;

y(n) = y(n - 1) + (x(n) - x(n - L)) / L;

endif;

endfor;

plot(abs(fft(y)));

for n = 1 : N ;

x(n) = y(n);

endfor;

for n = 1 : N;

if n == 1;

y(n) = x(n) / L;

elseif (n > 1 && n <= L);

y(n) = y(n - 1) + x(n) / L;

else;

y(n) = y(n - 1) + (x(n) - x(n - L)) / L;

endif

endfor;

#plot(abs(fft(y)));

**Приложения к пункту 6**

import wave

import matplotlib.pyplot as plt

import numpy as np

import scipy as sp

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):

L = 131

y=[]

k=len(lc)

for i in range (0,k):

y.append(0)

summa=0

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

if (i - l) > 0:

summa+=lc[i-l]/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()