Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Проектирования и технологии производства ЭА

ОТЧЕТ

По семинару №3

по курсу Киберфизические системы

Студент Фан К.
(Подпись, дата) (И.О.Фамилия)

Руководитель задания Леонидов В. В.
(Подпись, дата) (И.О.Фамилия)

Москва, 2023

Задание 1: Скользящее среднее в среде MATLAB

```
clear, clc, close all
fs = 100;
ts = 0 : 1/fs : 2 - 1/fs;
N = length(ts);

a = 1;
b = 0.2;

x = (a+(b-a)*rand(1,N)).*sin(2*pi*5*ts);

subplot(2,1,1);
plot(x), grid on
y = zeros(1, N+7);
for i = 7: N
y(i) = (x(i-1) + x(i-2) + x(i-3) + x(i-4) + x(i-5)+ x(i-6)) * 1/6;
end
subplot(2,1,2);
plot(y(1:200)), grid on
```

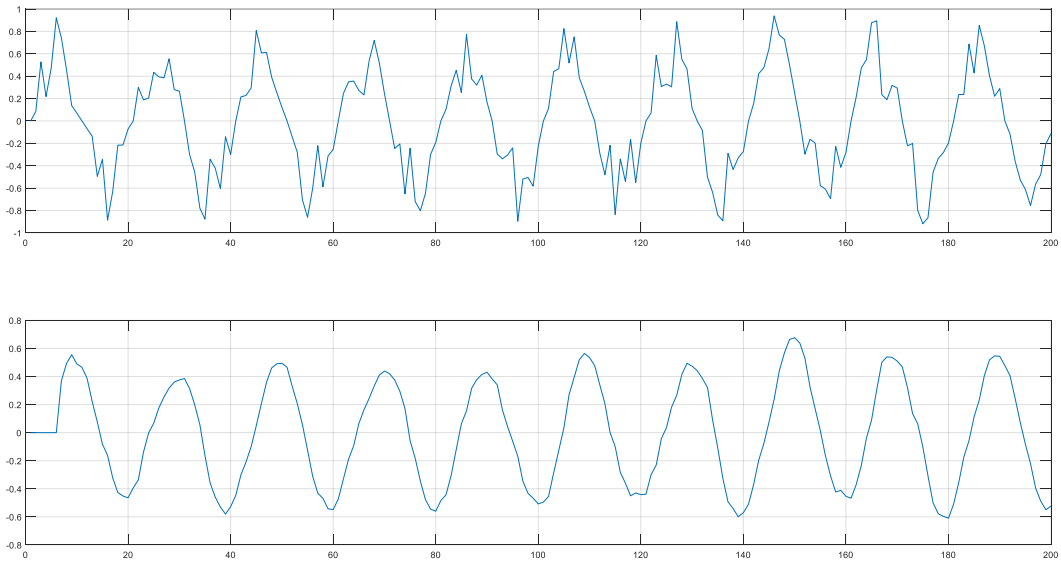


Рисунок 1: Скользящее среднее

Вывод:

Путем применения фильтра скользящего среднего мы получаем сигнал, похожий на исходный сигнал, но с некоторой задержкой и более гладкой формой.

Задание 2: Скользящее среднее функции Дирака в среде MATLAB

```
clear, clc, close all
fs = 100;
ts = 0 : 1/fs : 2 - 1/fs;
N = length(ts);

% Dirac func
x = zeros(1,N);
x(8) = 1;

subplot(3,1,1);
stem(x), grid on, title('Dirac func')
y = zeros(1, N+7);
for i = 7: N
y(i) = (x(i-1) + x(i-2) + x(i-3) + x(i-4) + x(i-5)+ x(i-6)) * 1/6;
end
subplot(3,1,2);
stem(y(1:200)), grid on, title('Impulse Response') % Импульсная характеристика

X = 2*abs(fft(y,N))/N;
f = (0:N-1)*fs/N;

subplot(3,1,3);
stem(f,X), grid on, title('Frequency Response')
```

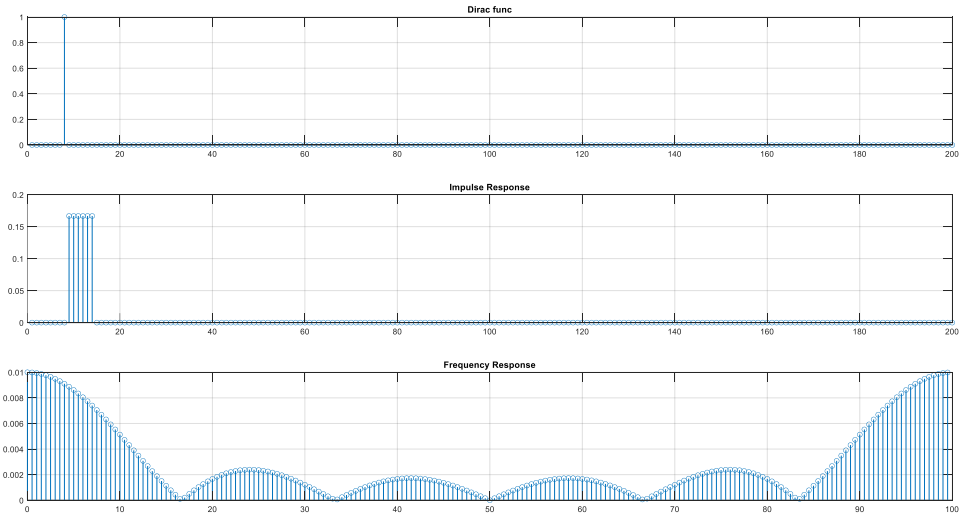


Рисунок 2: Скользящее среднее функции Дирака

Вывод:

В данном задании было использовано скользящее среднее для фильтрации функции Дирака. Результатом стала импульсная характеристика фильтра. Из рисунка видно, что импульсная характеристика фильтра представляет собой 6 отсчётов амплитудой 0.17 (или 1/6). Эти значении также коэффициенты фильтра.

Задание 3: КИХ-фильтр руками в среде MATLAB

```
clear, clc, close all
fs = 1000;
ts = 0:1/fs:1-1/fs;
N = length(ts);

x =zeros(N,1);
x(1:300)=1;
% ОДПФ(IDFT)
y = real(ifftshift(ifft(x)));

subplot(2,1,1);
plot(x), grid on, title('Expected Frequency Response')

subplot(2,1,2)
plot(N/2-150:N/2+150-1,y(N/2-150:N/2+150-1)), grid on , title('Impulse Response')

a = zeros(N,1);
a(1) = 1;
figure

Nf1 = 50;
af1 = filter(y(N/2-Nf1/2:N/2+Nf1/2-1),1,a);
af2 = filter(y(N/2-Nf1/2:N/2+Nf1/2-1).*hanning(Nf1),1,a);

subplot(2,2,1)
plot(abs(fft(af1))), grid on ,title('Frequency Response , N=50')
subplot(2,2,2)
plot(abs(fft(af2))), grid on ,title('Frequency Response with Hanning window , N=50')

Nf2 = 100;
af3 = filter(y(N/2-Nf2/2:N/2+Nf2/2-1),1,a);
af4 = filter(y(N/2-Nf2/2:N/2+Nf2/2-1).*hanning(Nf2),1,a);

subplot(2,2,3)
plot(abs(fft(af3))), grid on ,title('Frequency Response , N=100')
subplot(2,2,4)
plot(abs(fft(af4))), grid on ,title('Frequency Response with Hanning window , N=100')
```

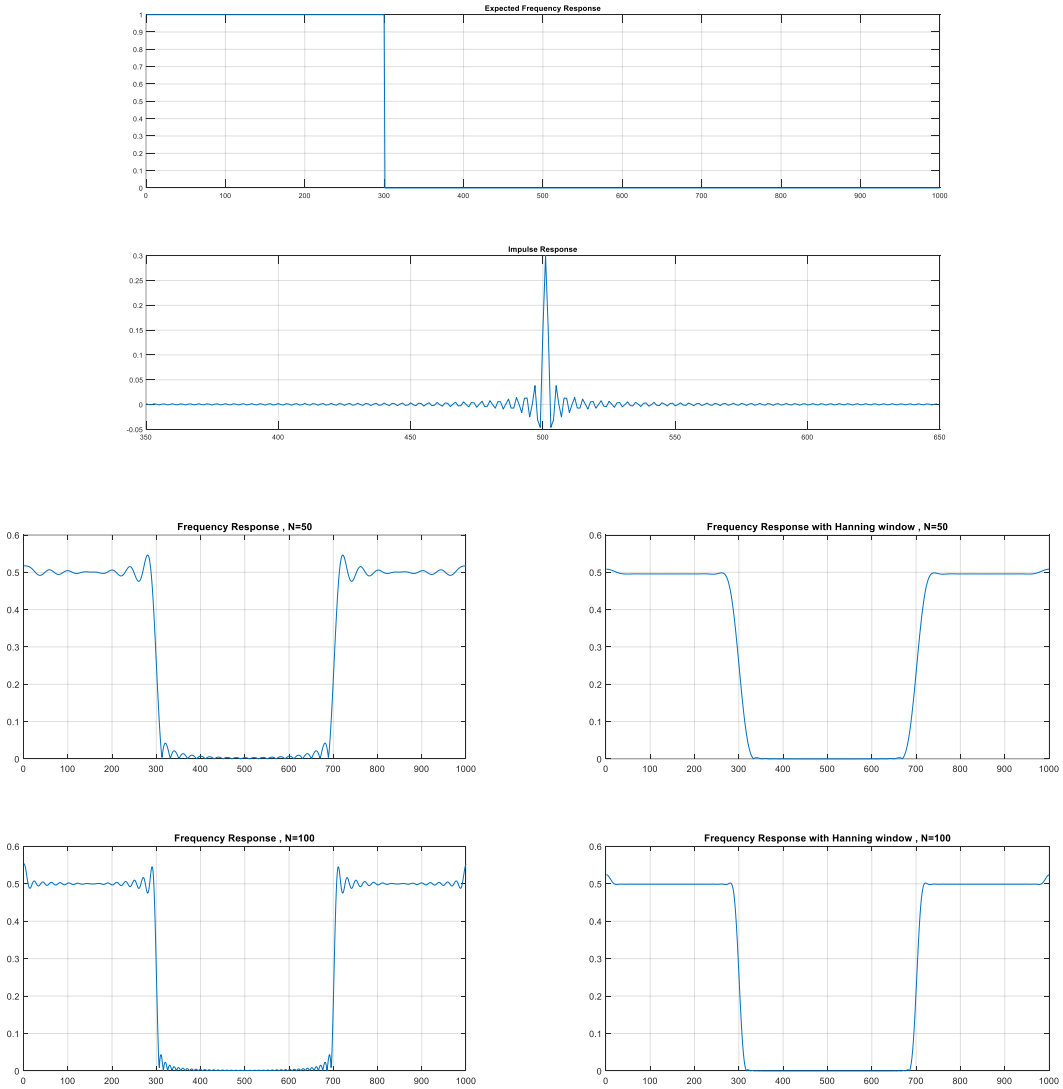


Рисунок 3: КИХ-фильтр

Вывод:

В данном задании от идеальной АЧХ фильтра делаем обратное ДПФ и получаем набор коэффициентов во временной области. Из рис.3 видно, что чем больше отсчётов фильтра, тем более его характеристика становится похожа на идеальную. Окно помогает убрать пульсации Гиббса из АЧХ.

Задание 4: Применение КИХ-фильтр к сигналу в среде MATLAB

```
clear, clc, close all
[x, fs] = audioread('rock.wav');
N = length(x);
X = 2*abs(fft(x))/N;
subplot(3,1,1);
plot(X(1:10000)), grid on, title('DFT input signal');

x1 = zeros(N,1);
x1(1:5000) = 1;
y = real(ifftshift(ifft(x1)));

a = zeros(N,1);
a(1) =1; % Dirac
Nf1 = 2500;
af1 = filter(y(N/2-Nf1/2:N/2+Nf1/2-1).*hanning(Nf1),1,a);
af2 = filter(y(N/2-Nf1/2:N/2+Nf1/2-1).*hanning(Nf1),1,x);
subplot(3,1,2);
plot(abs(fft(af1))), xlim([0 10000]), grid on , title('Frequency Response low pass
filter')
subplot(3,1,3);
plot(abs(fft(af2))), xlim([0 10000]), grid on , title('DFT filtered signal')
```

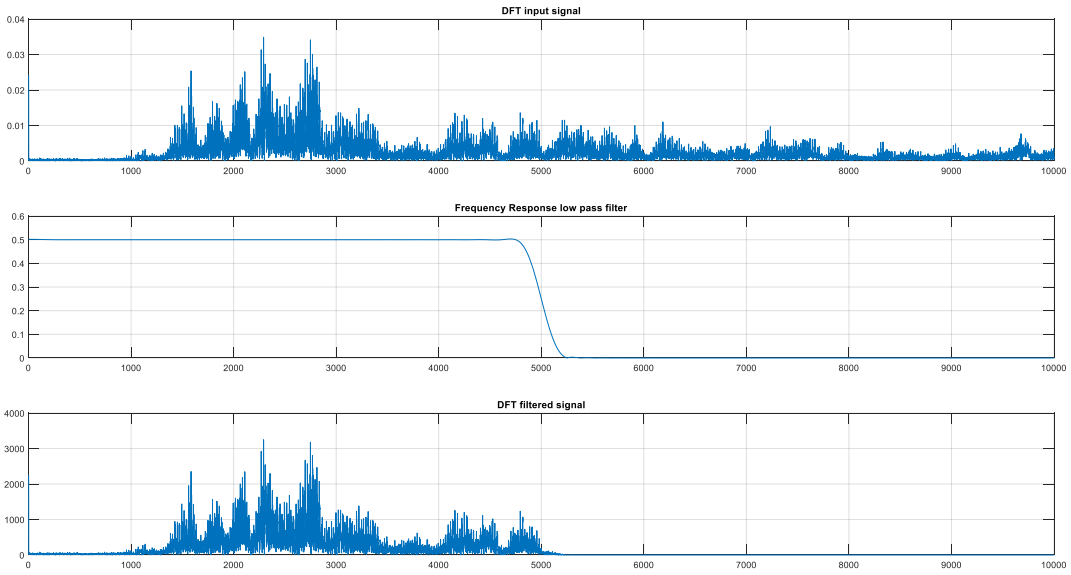


Рисунок 4: Применение КИХ-фильтр к сигналу

Вывод:

КИХ-фильтр ФНЧ помогает удалить высокие частоты из сигнала, а низкочастотные сигналы сохраняются

Задание 5: КИХ-фильтр при помощи filterDesigner в среде MATLAB

```
clear, clc, close all;
fs = 20000;
ts = 0:1/fs:1-1/fs;
N = length(ts);
x = 0.15*sin(2*pi*1500*ts) + sin(2*pi*5000*ts+3*pi/4)+ 0.15*cos(2*pi*8500*ts);
subplot(2,2,1)
plot(x),xlim([100 200]), grid on, title('Input signal')
subplot(2,2,2)
stem(2*abs(fft(x))/N), xlim([0 N/2]) ,grid on, title('DFT input signal')
fir = bpf;
y = filter(fir.Numerator, 1,x);
subplot(2,2,3)
plot(y),xlim([100 200]), grid on, title('Filtered signal')
subplot(2,2,4)
stem(2*abs(fft(y))/N), xlim([0 N/2]) , grid on, title('DFT filtered signal')
```

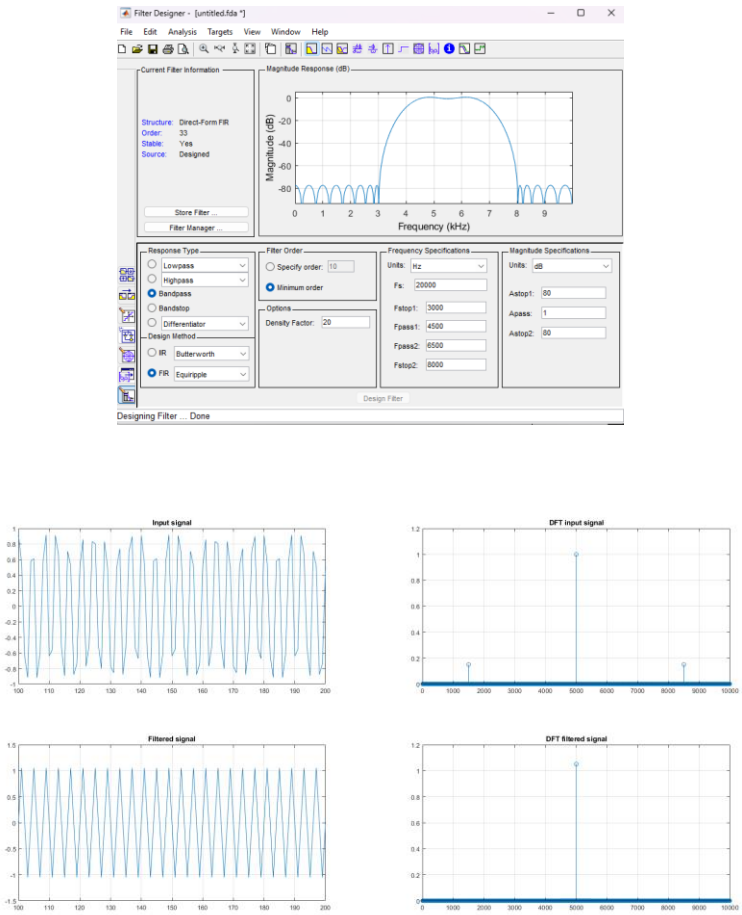


Рисунок 5: КИХ-фильтр при помощи filterDesigner

Вывод:

В этом задании мы используем инструмент FilterDesigner для создания Bandpass КИХ-фильтра. Этот фильтр помогает нам удалить из сигнала частотный шум 1500 Гц и 8500 Гц.

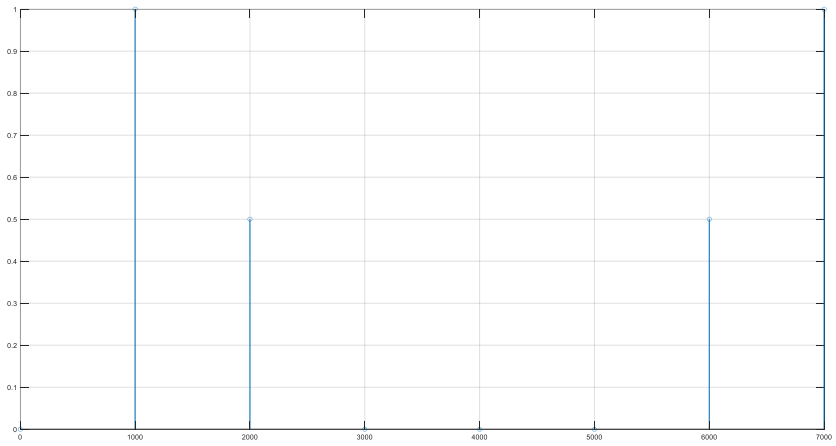
Задание 6: Реализация алгоритма Гёрцеля в среде MATLAB

```
clear,clc, close all
fs =8000;
ts =0:1/fs:0.001-1/fs;
N =length(ts);
x = sin(2*pi*1000*ts)+0.5*sin(2*pi*2000*ts+3*pi/4);
X = 2*abs(fft(x))/N;
f = 0: fs/N:fs-fs/N;
stem(f,X), grid on

fg = 2000;
m = fg/fs*N+1;
Xf = 0;
for n = 1:N
    Xf = Xf + x(n) * (cos(2*pi*(n-1)*(m-1)/N)-1i*sin(2*pi*(n-1)*(m-1)/N));
end

Xm= 2*abs(Xf)/N;
disp('way 1: DFT at 1 sample');
disp(Xm);
u1 =0;
u2 =0;
w =2*pi*(m-1)/N;
for n=1:N
    u0 = 2*cos(w)*u1-u2+x(n);
    u2=u1;
    u1=u0;
end
y = u0 - exp(-1i*w)*u2;
Y = 2*abs(y)/N;
disp('way 2: Use Goertzel algorithm');
disp(Y);

H = goertzel(x,m);
Hm =2*abs(H)/N;
disp('way 3: Use func goertzel Matlab');
disp(Hm)
```




```
way 1: DFT at 1 sample
    0.5000

way 2: Use Goertzel algorithm
    0.5000

way 3: Use func goertzel Matlab
    0.5000
```

Рисунок 6: Реализация алгоритма Гёрцеля

Вывод:

В этом задании мы вычисляем амплитуду сигнала на частоте 2 кГц, используя несколько методов, включая ДПФ, алгоритм Гёрцеля и функцию goertzel в Matlab. Все три метода дают одинаковый результат: 0.5.

Задание 7: Эквалайзер на базе БИХ-фильтров в среде MATLAB

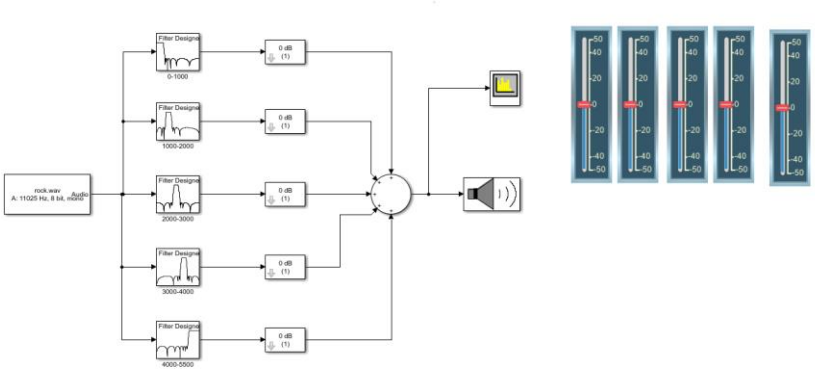


Рисунок 7.1 – Эквалайзер на базе БИХ-фильтров

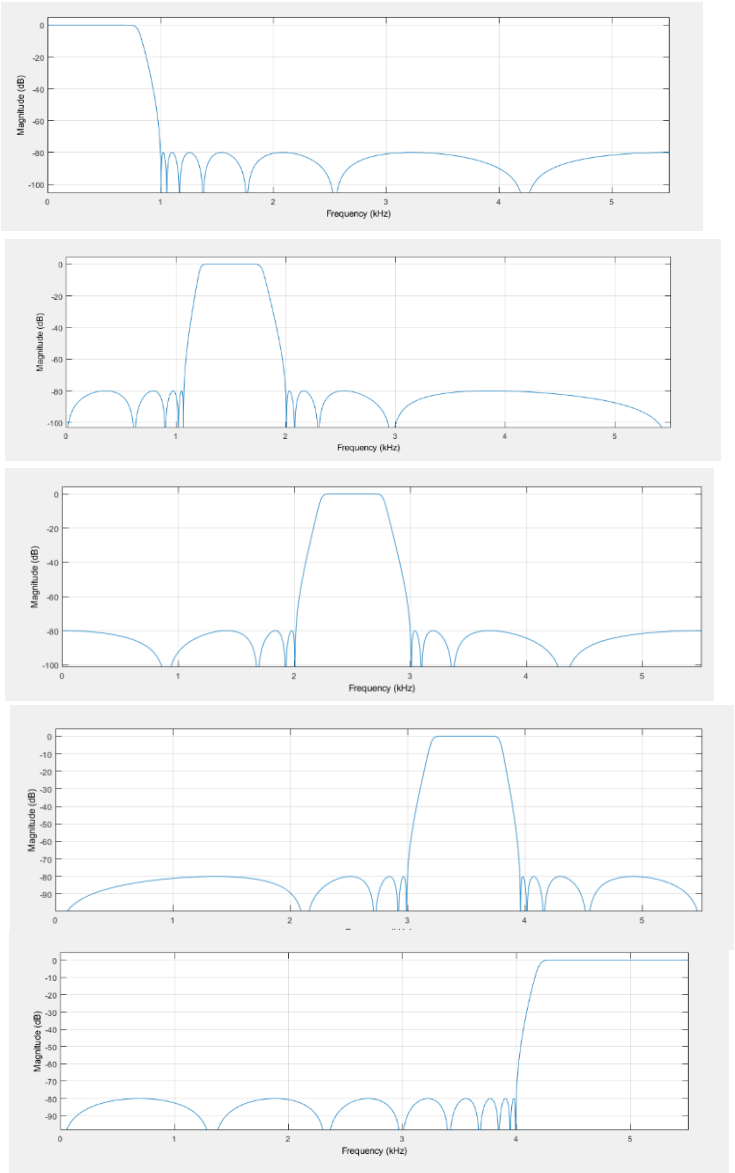


Рисунок 7.2 – Характеристика фильтров

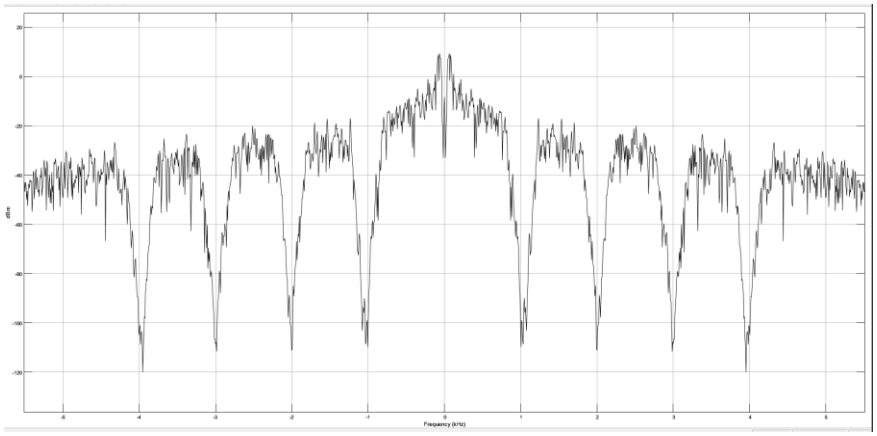


Рисунок 7.3 – Сигнал без фильтрации

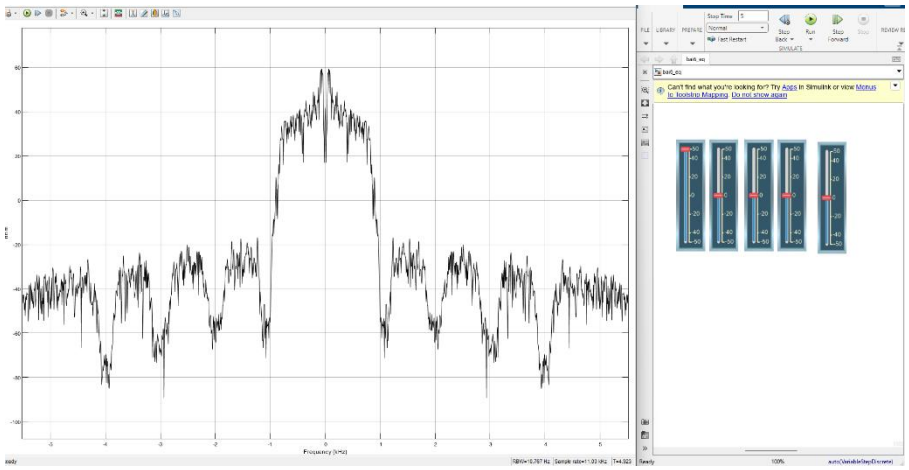


Рисунок 7.4 – Усиление 0-1000 Гц

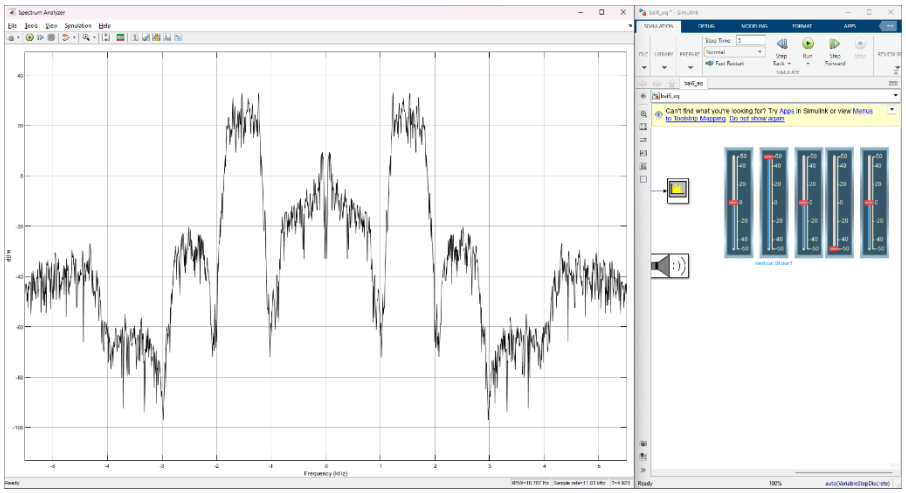


Рисунок 7.4 – Усиление 1000-2000 Гц, подавление диапазона 3000-4000Гц

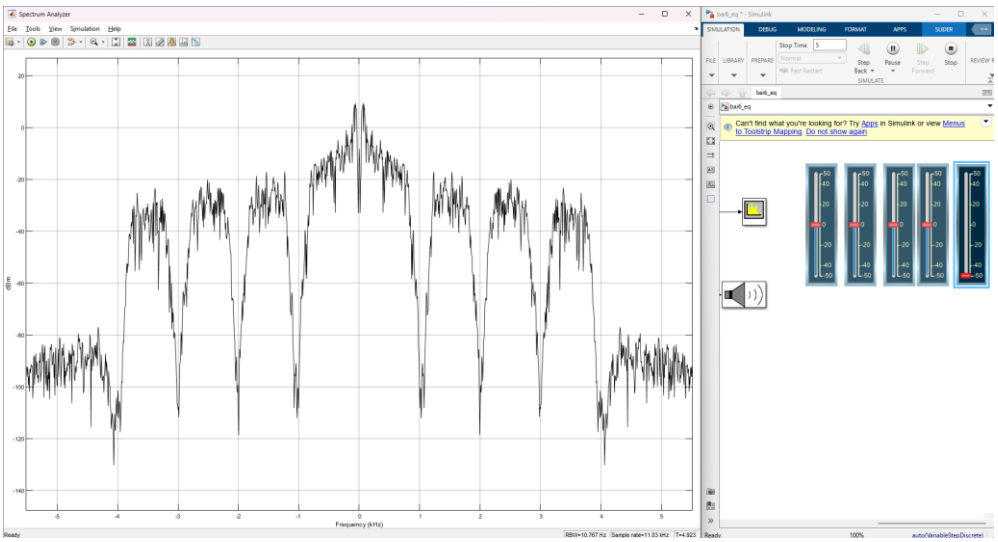


Рисунок 7.5 – Подавление высоких частот

Вывод:

В этом задании мы проектируем эквалайзер на базе БИХ-фильтров в пакете Matlab Simulink. Эквалайзер -это устройство, который увеличивает или уменьшает громкость разных частот в звуковом сигнале.