



**«Московский государственный технический университет
имени Н.Э. Баумана»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ

Информатики и систем управления

КАФЕДРА

Проектирования и технологии производства ЭА

Се м и н а р № 3

**п о к у р с у « Ц и ф р о в а я о б р а б о т к а
с и г н а л о в »**

Студент

(Подпись, дата) Корчагин А.И.
(И.О.Фамилия)

Преподаватель

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

```
##### 1 БЕРЕМ СРЕДНЕЕ #####
```

```
clear;  
close all;
```

```
Fs = 97;  
ts = 0: 1/Fs : 16-1/Fs;  
N = length(ts);
```

```
% параметры случайного сигнала
```

```
a = -0.01;  
b = 0.1;
```

```
% Задаем случайные сигналы
```

```
%x = zeros(N);  
%x(5) = 1;  
x = (a + (b - a) * rand(1, N)).*sin(2*pi*0.5*ts);
```

```
figure;  
subplot(2,1,1);  
plot(x); grid on; title('Исходный сигнал');
```

```
y = zeros(1,N+6);  
for i = 6 : length(x)  
    y(i) = (x(i-1) + x(i-2) + x(i-3) + x(i-4) + x(i-5))*1/5;  
end
```

```
subplot(2,1,2);  
plot(y(1:100)); grid on; title('Усредненный сигнал');
```

```
%Функция Диракта
```

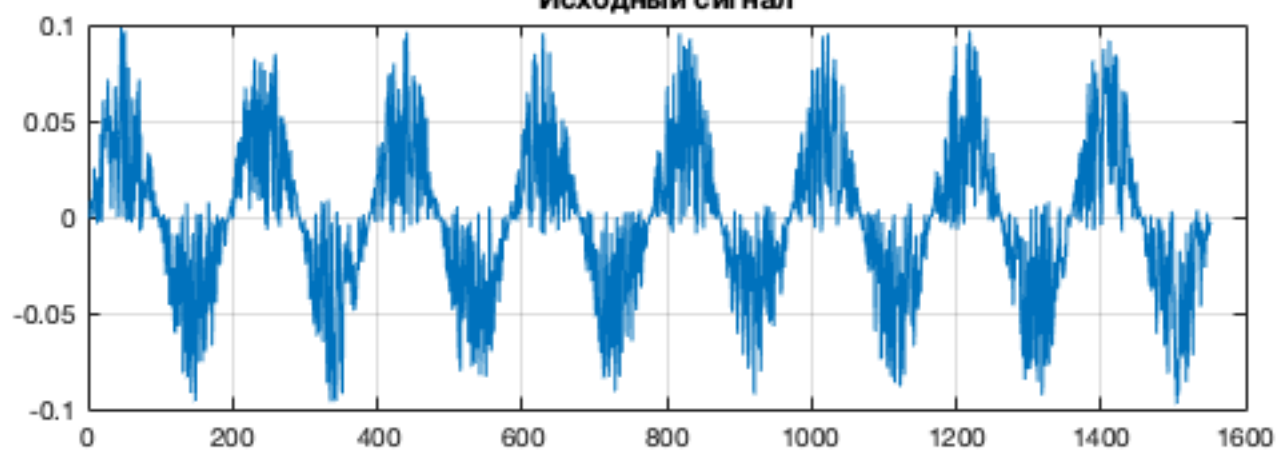
```
x = zeros(N);  
x(50) = 1;
```

```
figure;  
subplot(2, 1, 1);  
stem(x); grid on; title('Функция Диракта');  
xlabel('Время'); ylabel('Амплитуда');
```

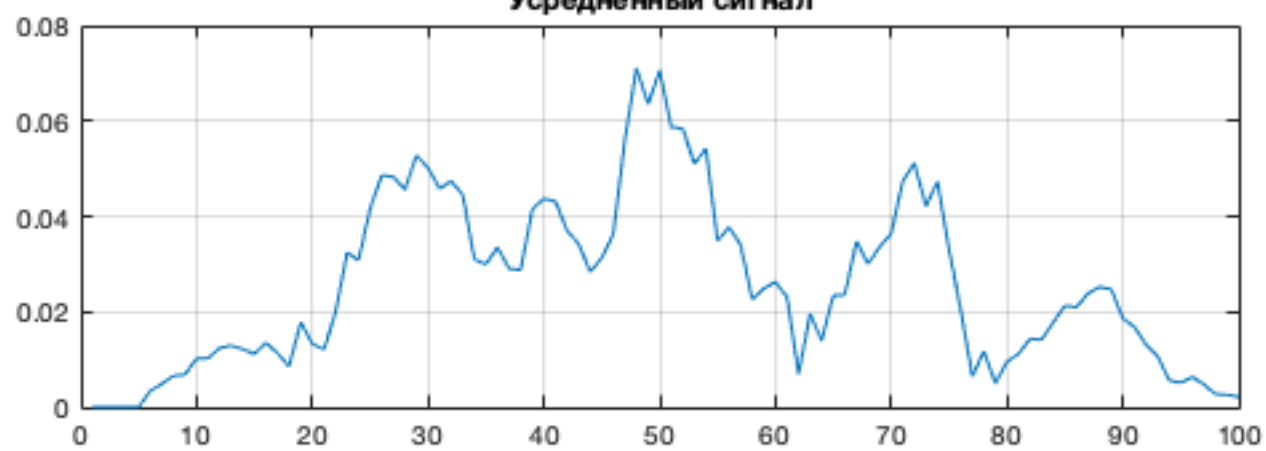
```
z = zeros(1, N + 6);  
for i = 6 : length(x)  
    y(i) = (x(i - 1) + x(i - 2) + x(i - 3) + x(i - 4) + x(i - 5)) / 5;  
end
```

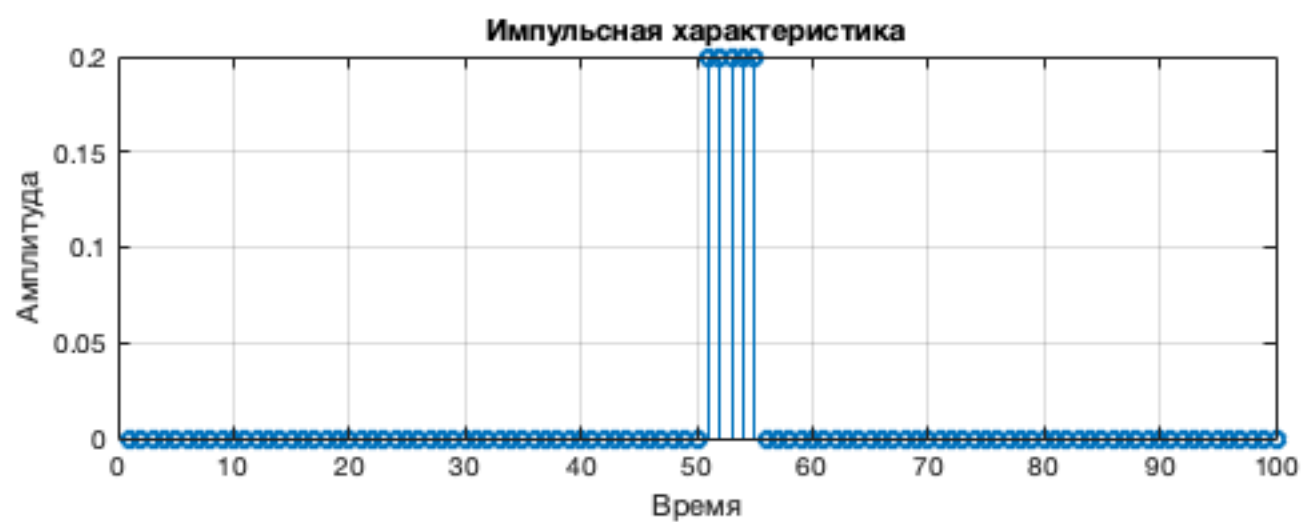
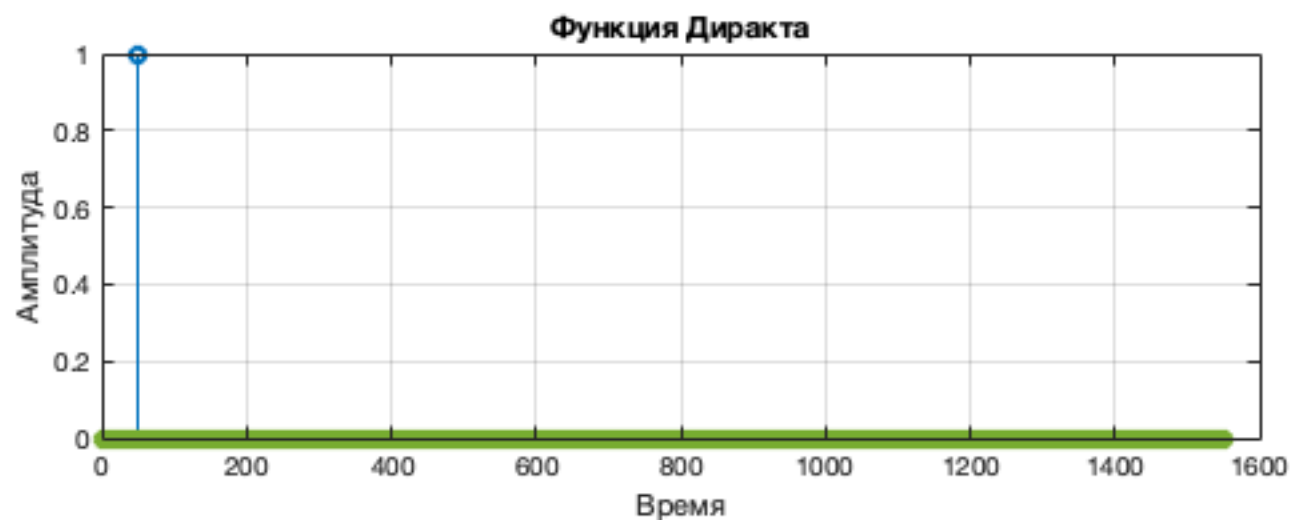
```
subplot(2, 1, 2);  
stem(y(1: 100)); grid on; title('Импульсная характеристика')  
xlabel('Время'); ylabel('Амплитуда');
```

Исходный сигнал



Усредненный сигнал





2 Филтър

function Hd = FIR1

%FIR1 Returns a discrete-time filter object.

% MATLAB Code

% Generated by MATLAB(R) 9.5 and DSP System Toolbox 9.7.

% Generated on: 14-Apr-2020 10:04:34

% Equiripple Lowpass filter designed using the FIRPM function.

% All frequency values are in Hz.

Fs = 125; % Sampling Frequency

Fpass = 40; % Passband Frequency

Fstop = 45; % Stopband Frequency

Dpass = 0.057501127785; % Passband Ripple

Dstop = 0.0001; % Stopband Attenuation

dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.

[N, Fo, Ao, W] = firpmord([Fpass, Fstop]/(Fs/2), [1 0], [Dpass, Dstop]);

% Calculate the coefficients using the FIRPM function.

b = firpm(N, Fo, Ao, W, {dens});

Hd = dfilt.dffir(b);

Ts = 0 : 1/Fs : 15 - 1/Fs;

N = length(Ts);

f1 = 30;

f2 = 40;

f3 = 50;

f4 = 60;

x = 0.5*sin(2*pi*f1*Ts) + ...
0.65*sin(2*pi*f2*Ts) + ...
0.8*sin(2*pi*f3*Ts) + ...
0.95*sin(2*pi*f4*Ts);

subplot(2, 2, 1);

plot(x); hold on; title('Исходный сигнал');

xlabel('Время'); ylabel('Амплитуда');

X = abs(fft(x));

Xm = 2*abs(X)/N;

subplot(2, 2, 2);

plot(X, Xm); grid on; title('БПФ Исходного сигнала');

xlabel('Частота'); ylabel('Амплитуда');

y = filter(Hd, x);

X = abs(fft(y));

```

subplot(2, 2, 3);
plot(y); grid on; title('Отфильтрованный сигнал');
xlabel('Время'); ylabel('Амплитуда');
Xm = 2 * abs(X) / N;
X = (0: N - 1) * Fs / N;
subplot(2, 2, 4);
plot(X, Xm); grid on; title('БПФ отфильтрованного сигнала');
xlabel('Частота'); ylabel('Амплитуда');

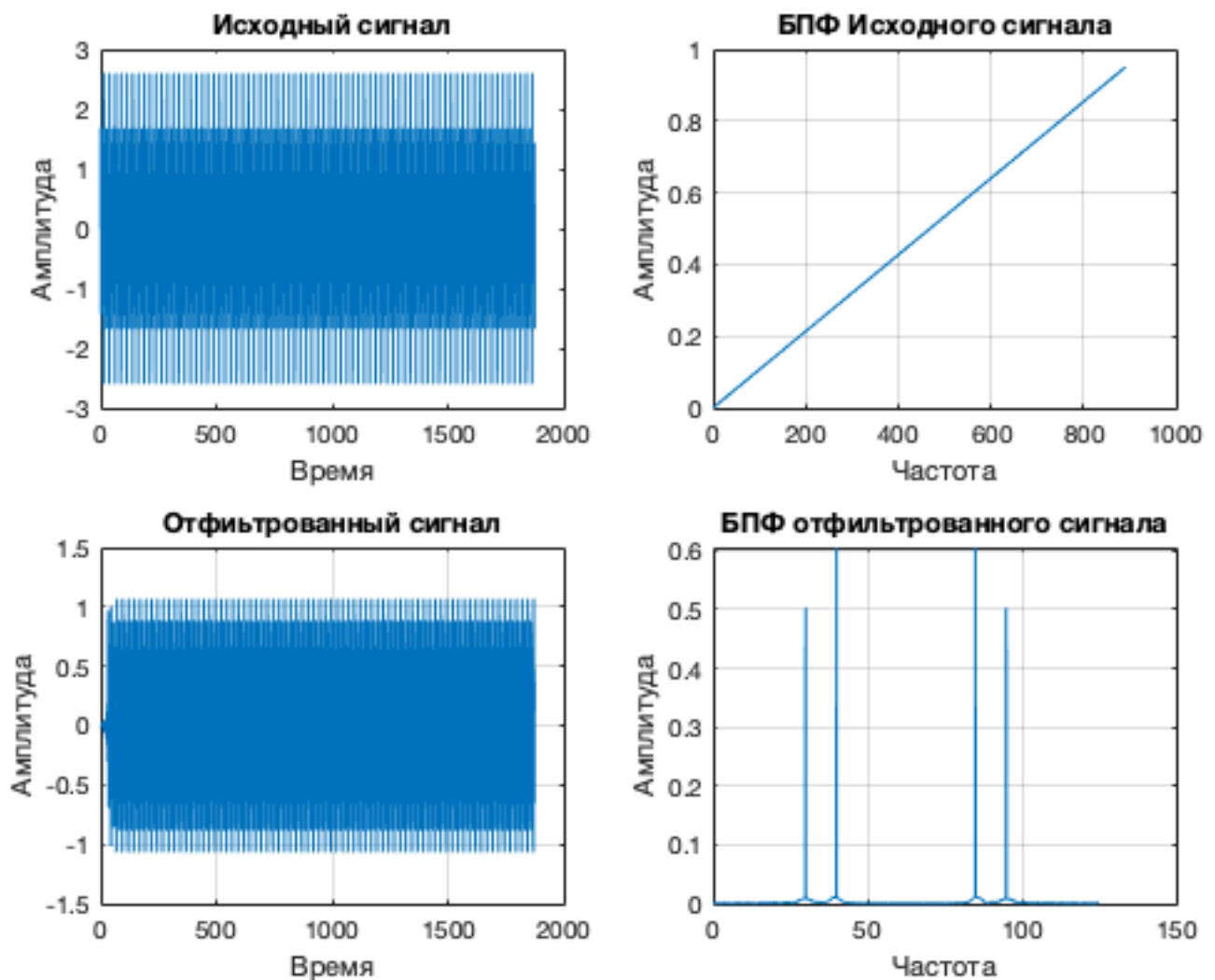
```

ans =

```

FilterStructure: 'Direct-Form FIR'
Arithmetic: 'double'
Numerator: [1x64 double]
PersistentMemory: false

```



```

#### 2 Фильтр WAV ####
function Hd = FIR_WAV
%FIR_WAV Returns a discrete-time filter object.

% MATLAB Code
% Generated by MATLAB(R) 9.5 and DSP System Toolbox 9.7.
% Generated on: 14-Apr-2020 10:28:58

% Equiripple Bandpass filter designed using the FIRPM function.

% All frequency values are in Hz.
Fs = 25000; % Sampling Frequency

Fstop1 = 20; % First Stopband Frequency
Fpass1 = 40; % First Passband Frequency
Fpass2 = 80; % Second Passband Frequency
Fstop2 = 200; % Second Stopband Frequency
Dstop1 = 0.001; % First Stopband Attenuation
Dpass = 0.057501127785; % Passband Ripple
Dstop2 = 0.0001; % Second Stopband Attenuation
dens = 20; % Density Factor

% Calculate the order from the parameters using FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fstop1 Fpass1 Fpass2 Fstop2]/(Fs/2), [0 1 ...
    0], [Dstop1 Dpass Dstop2]);

% Calculate the coefficients using the FIRPM function.
b = firpm(N, Fo, Ao, W, {dens});
Hd = dfilt.dffir(b);

% [EOF]

[x, fs] = audioread('rock.wav');

zone = x(:,1);
N = length(zone);

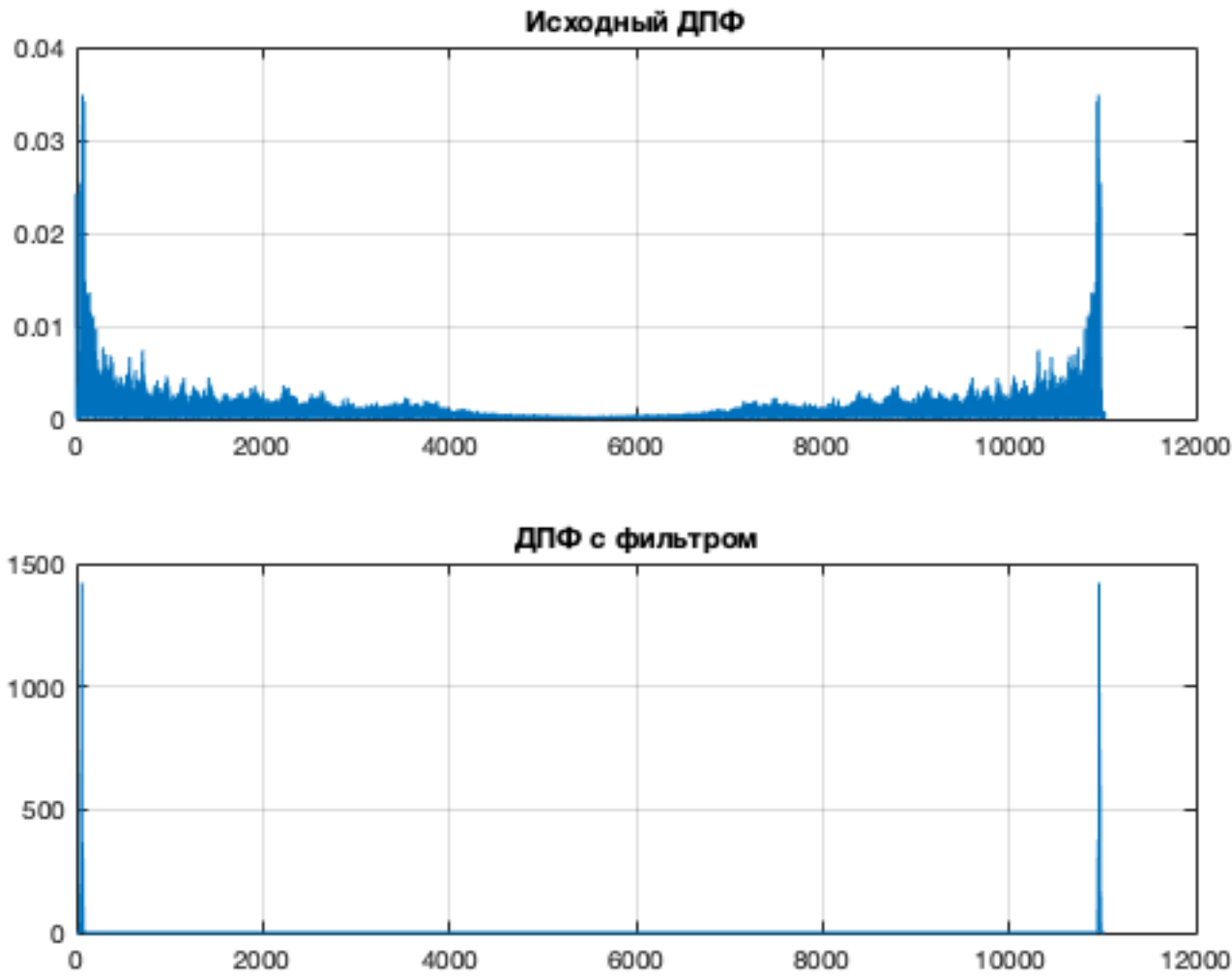
Xm = 2*(abs(fft(zone)))/N;
F = (0 : N - 1) * fs / N;
subplot(2, 1, 1);
plot(F, Xm); grid on; title('Исходный ДПФ');

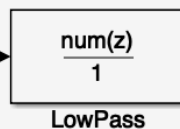
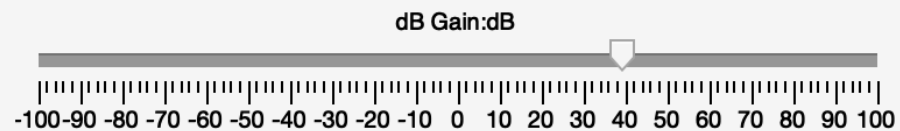
filtered_x = filter(Hd, zone);
xfft = abs(fft(filtered_x(:,1)));
Xm = 2*(xfft)/N;
F = (0 : N - 1) * fs / N;
subplot(2, 1, 2);
plot(F, Xm); grid on; title('ДПФ с фильтром');
sound(3*filtered_x, fs);

```

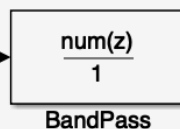
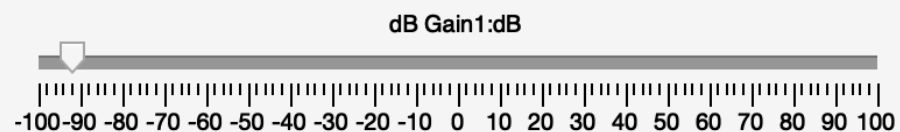
ans =

```
FilterStructure: 'Direct-Form FIR'  
Arithmetic: 'double'  
Numerator: [1x2468 double]  
PersistentMemory: false
```

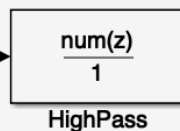
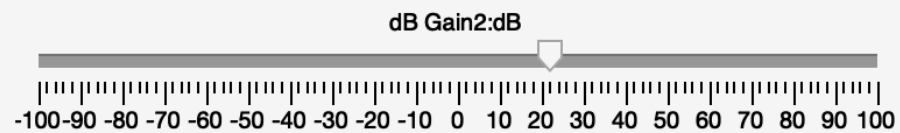




39.1667 dB
(90.8518)



-91.3889 dB
(2.69498e-05)



21.9444 dB
(12.509)

rock.wav
A: 11025 Hz, 8 bit, mono

