

Универзитет у Београду
Електротехнички факултет

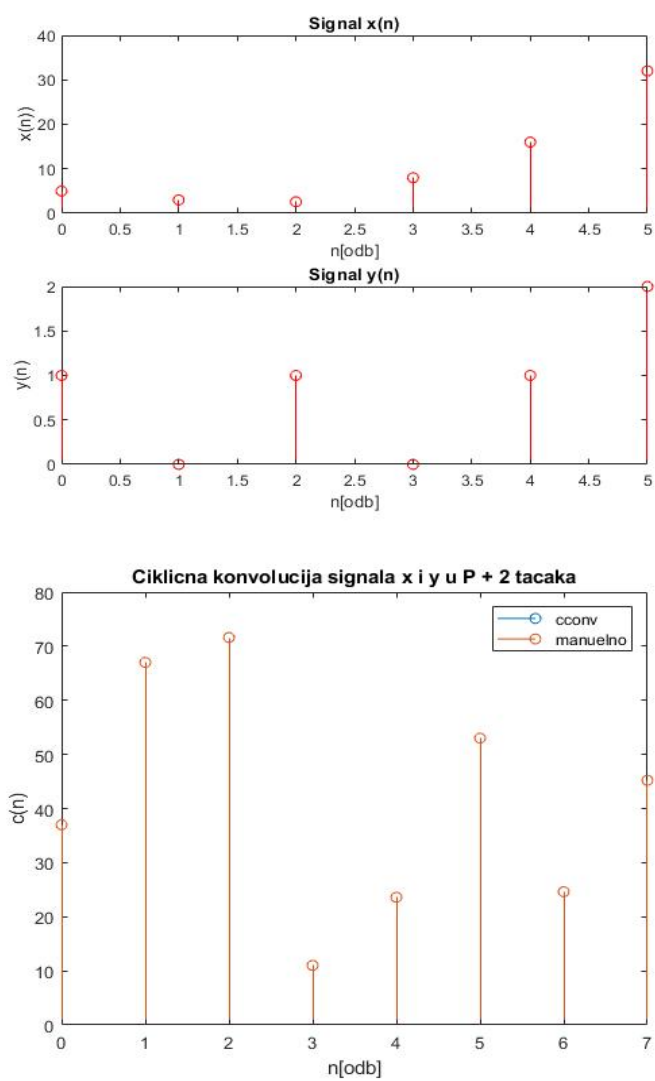
Домаћи задатак из Дигиталне обраде сигнала

Вредности параметара:
 $(P, Q, R, S) = (6, 2, 2, 2)$

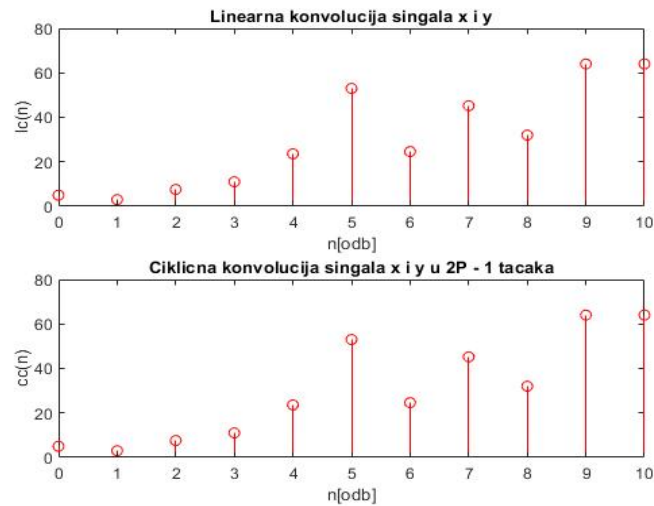
студент:
Александар Ђорђевић, 0086/2019, ОС

Децембар 2021.

1 Задатак 1



Да би циклична конволуција била једнака линеарној њу треба рачунати у $M + L - 1$ тачака где је M дужина прве секвенце, L дужина друге.



```
%Aleksandar Djordjevic 2019/0086

%% Zadatak 1 - ucitavanje signala

clear all;
close all;
clc

godina_upisa = 2019;
broj_indeksa = 86;

P = mod(broj_indeksa,4) + 4;

n = 0:1:(P - 1);

n1 = n(1:floor(P/2));
n2 = n(floor(P/2) + 1:P);

x1 = sin(n1) + 2*cos(2*n1) + P/2;
x2 = 2.^n2;

x = [x1 x2];

y1 = (-1).^n1 + mod(n1,2);
y2 = n2 - P/2;
```

```
y = [y1 y2];

%% Zadatak 1 pod a)

figure(1)

subplot(2,1,1)
stem(n,x,'r');
xlabel('n[odb]');
ylabel('x(n)');
title('Signal x(n)');

subplot(2,1,2)
stem(n,y,'r');
xlabel('n[odb]');
ylabel('y(n)');
title('Signal y(n)');

N = P + 2;

c = zeros(1,N);

add_zeros = zeros(1, N - length(x)); % niz nula koji
    treba dodati na signale

x_p2 = [x add_zeros];
y_p2 = [y add_zeros];

% racunanje ciklicne konvolucije
for k = 0:N-1
    c(k+1) = x_p2*y_p2(mod(k:-1:k-(N - 1),N)+1)';
end

n_p2 = 0:1:N-1;

c2 = cconv(x,y,N); % provera

figure(2)
stem(n_p2,c2);
hold on;
stem(n_p2,c);
hold off;
xlabel('n[odb]');
ylabel('c(n)');
title('Ciklicna konvolucija signala x i y u P + 2
```

```
    tacaka');
legend('cconv','manuelno');

%% Zadatak 1 pod b)

N = length(x) + length(y) - 1;

n_l = 0:1:N-1;

cc = zeros(1,N);

add_zeros = zeros(1, N - length(x));

x_2pm1 = [x add_zeros];
y_2pm1 = [y add_zeros];

% racunanje ciklicne konvolucije u 2P - 1 tacaka
for k = 0:N-1
    cc(k+1) = x_2pm1*y_2pm1(mod(k:-1:k-(N - 1),N)+1)';
end

lc = zeros(1,N);

% racunanje linearne konvolucije
for i = 1:N
    for j = 1:i
        lc(i) = lc(i) + x_2pm1(j)*y_2pm1(i - j + 1);
    end
end

figure(3)

subplot(2,1,1);
stem(n_l,lc,'r')
title('Linearna konvolucija singala x i y');
xlabel('n[odb]');
ylabel('lc(n)');

subplot(2,1,2)
stem(n_l,cc,'r')
title('Ciklicna konvolucija singala x i y u 2P - 1
    tacaka');
xlabel('n[odb]');
ylabel('cc(n)');

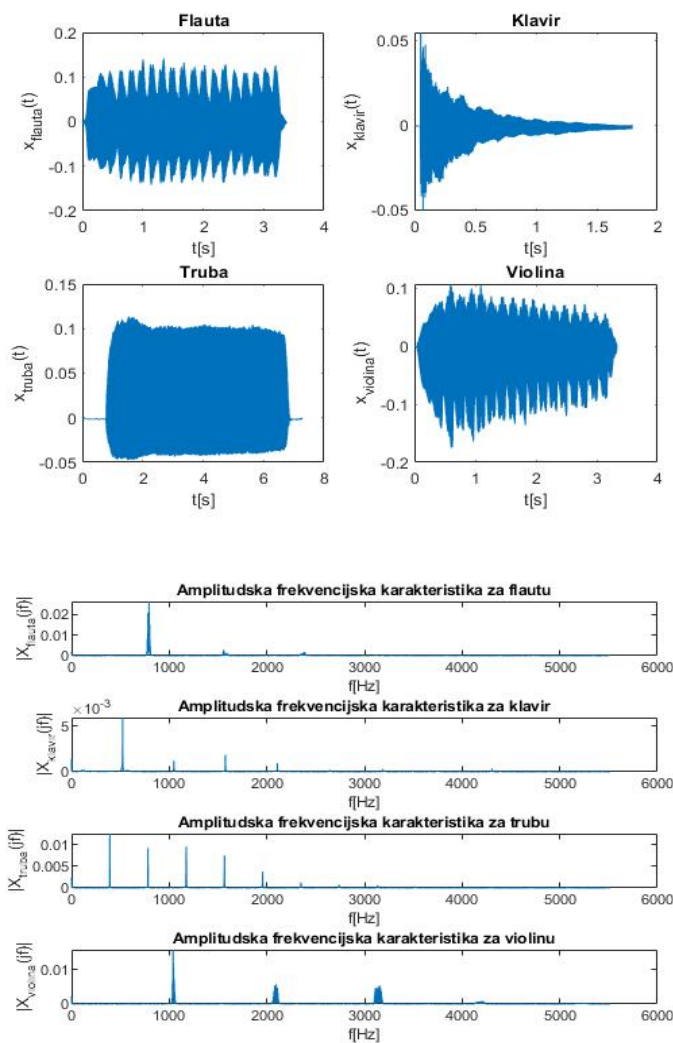
disp(lc == conv(x,y)); % proverava jednakosti ugradjene
```

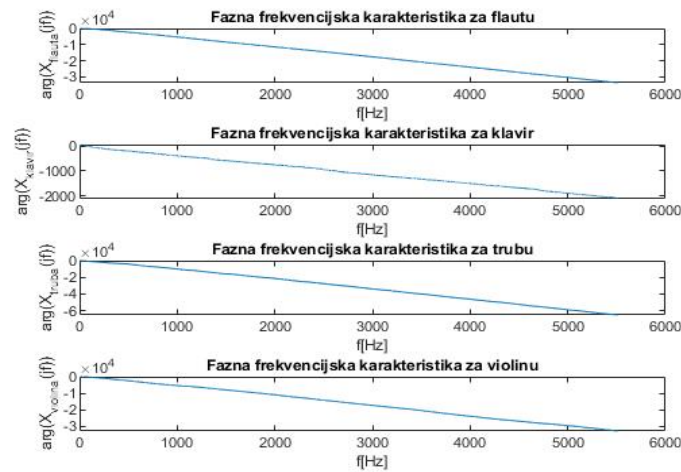
```

funkcije za lin konv i moje
disp(lc == cc); % provera jednakosti ciklicne i lin
konv

```

2 Задатак 2





Фреквенција првог пика за флауту је 793.7Hz , за клавир је 523.8625Hz , за трубу је 391.6351Hz и за виолину је 1039.6Hz . Из табеле која је дата у поставци задатка видимо да су то тонови G5(флаута), C5(клавир), G4(труба), C6(виолина).

Део под д) је урађен за трубу и добило се да је просечан размак између пикова 391.551Hz .

```
%Aleksandar Djordjevic 2019/0086

%% Ucitavanje signala

clear all
close all
clc

[x_flauta, Fs_flauta] = audioread('flauta_3.wav');
[x_klavir, Fs_klavir] = audioread('klavir_3.wav');
[x_truba, Fs_truba] = audioread('truba_3.wav');
[x_violina, Fs_violina] = audioread('violina_3.wav');

t_flauta = 0:1/Fs_flauta:(length(x_flauta) - 1)/
    Fs_flauta;
t_klavir = 0:1/Fs_klavir:(length(x_klavir) - 1)/
    Fs_klavir;
t_truba = 0:1/Fs_truba:(length(x_truba) - 1)/Fs_truba;
t_violina = 0:1/Fs_violina:(length(x_violina) - 1)/
    Fs_violina;

%% Zadatak 2 pod a)
```

```
figure(1)

subplot(2,2,1)
plot(t_flauta,x_flauta);
xlabel('t[s]');
ylabel('x_{flauta}(t)');
title('Flauta');

subplot(2,2,2)
plot(t_klavir,x_klavir);
xlabel('t[s]');
ylabel('x_{klavir}(t)');
title('Klavir');

subplot(2,2,3)
plot(t_truba,x_truba);
xlabel('t[s]');
ylabel('x_{truba}(t)');
title('Truba');

subplot(2,2,4)
plot(t_violina,x_violina);
xlabel('t[s]');
ylabel('x_{violina}(t)');
title('Violina');

%% Zadatak 2 pod b)

N_flauta = 2^nextpow2(length(x_flauta));
N_klavir = 2^nextpow2(length(x_klavir));
N_truba = 2^nextpow2(length(x_truba));
N_violina = 2^nextpow2(length(x_violina));

X_flauta = fft(x_flauta,N_flauta)/length(x_flauta);
X_klavir = fft(x_klavir,N_klavir)/length(x_klavir);
X_truba = fft(x_truba,N_truba)/length(x_truba);
X_violina = fft(x_violina,N_violina)/length(x_violina)
;

f_flauta = 0:Fs_flauta/N_flauta:Fs_flauta/2;
f_klavir = 0:Fs_klavir/N_klavir:Fs_klavir/2;
f_truba = 0:Fs_truba/N_truba:Fs_truba/2;
f_violina = 0:Fs_violina/N_violina:Fs_violina/2;

amp_flauta = abs(X_flauta(1:N_flauta/2+1));
```



```
amp_flauta(2:N_flauta/2+1) = 2*amp_flauta(2:N_flauta/2+1);

amp_klavir = abs(X_klavir(1:N_klavir/2+1));
amp_klavir(2:N_klavir/2+1) = 2*amp_klavir(2:N_klavir/2+1);

amp_truba = abs(X_truba(1:N_truba/2+1));
amp_truba(2:N_truba/2+1) = 2*amp_truba(2:N_truba/2+1);

amp_violina = abs(X_violina(1:N_violina/2+1));
amp_violina(2:N_violina/2+1) = 2*amp_violina(2:N_violina/2+1);

figure(5)

subplot(4,1,1)
plot(f_flauta,amp_flauta);
xlabel('f[Hz]');
ylabel('|X_{flauta}(jf)|');
title('Amplitudska frekvencijska karakteristika za flautu');

subplot(4,1,2)
plot(f_klavir,amp_klavir);
xlabel('f[Hz]');
ylabel('|X_{klavir}(jf)|');
title('Amplitudska frekvencijska karakteristika za klavir');

subplot(4,1,3)
plot(f_truba,amp_truba);
xlabel('f[Hz]');
ylabel('|X_{truba}(jf)|');
title('Amplitudska frekvencijska karakteristika za trubu');

subplot(4,1,4)
plot(f_violina,amp_violina);
xlabel('f[Hz]');
ylabel('|X_{violina}(jf)|');
title('Amplitudska frekvencijska karakteristika za violinu');

ph_flauta = unwrap(angle(X_flauta(1:N_flauta/2+1)));
ph_klavir = unwrap(angle(X_klavir(1:N_klavir/2+1)));
```

```
ph_truba = unwrap(angle(X_truba(1:N_truba/2+1)));
ph_violina = unwrap(angle(X_violina(1:N_violina/2+1)));
;

figure(6)

subplot(4,1,1)
plot(f_flauta,ph_flauta);
xlabel('f[Hz]');
ylabel('arg(X_{flauta}(jf))');
title('Fazna frekvencijska karakteristika za flautu');

subplot(4,1,2)
plot(f_klavir,ph_klavir);
xlabel('f[Hz]');
ylabel('arg(X_{klavir}(jf))');
title('Fazna frekvencijska karakteristika za klavir');

subplot(4,1,3)
plot(f_truba,ph_truba);
xlabel('f[Hz]');
ylabel('arg(X_{truba}(jf))');
title('Fazna frekvencijska karakteristika za trubu');

subplot(4,1,4)
plot(f_violina,ph_violina);
xlabel('f[Hz]');
ylabel('arg(X_{violina}(jf))');
title('Fazna frekvencijska karakteristika za violinu')
;

%% Zadatak 2 pod c) i pod d)

%FLAUTA

[pik_vred,pik_frekv] = findpeaks(amp_flauta);

pik_frekv = pik_frekv(pik_vred > 0.05*max(pik_vred));
pik_vred = pik_vred(find(pik_vred > 0.05*max(pik_vred)
));

pik_vred = pik_vred(3:length(pik_vred));

pik_frekv = pik_frekv*Fs_flauta/N_flauta;
pik_frekv = pik_frekv(3:length(pik_frekv));
```

```
i = 1;
count = 1;

while(i < length(pik_frekv))
    j = i + 1;
    while(j < length(pik_frekv) && abs(pik_frekv(i) -
        pik_frekv(j)) < 100)
        j = j + 1;
    end
    frekvencije_flauta(count) = pik_frekv(find(
        pik_vred == max(pik_vred(i:j))));
    count = count + 1;
    i = j + 1;
end

frekvencije_flauta;
ton_flauta = frekvencije_flauta(1);
frekvencije_flauta = diff(frekvencije_flauta);
prosecan_razmak_flauta = mean(frekvencije_flauta);

% KLAVIR

[pik_vred,pik_frekv] = findpeaks(amp_klavir);

pik_frekv = pik_frekv(find(pik_vred > 0.05*max(
    pik_vred)));
pik_vred = pik_vred(find(pik_vred > 0.05*max(pik_vred)
    ));

pik_vred = pik_vred(3:length(pik_vred));

pik_frekv = pik_frekv*Fs_klavir/N_klavir;
pik_frekv = pik_frekv(3:length(pik_frekv));

i = 1;
count = 1;

while(i < length(pik_frekv))
    j = i + 1;
    while(j < length(pik_frekv) && abs(pik_frekv(i) -
        pik_frekv(j)) < 100)
        j = j + 1;
    end
    frekvencije_klavir(count) = pik_frekv(find(
```

```
        pik_vred == max(pik_vred(i:j))));
count = count + 1;
i = j + 1;
end

frekvencije_klavir;
ton_klavir = frekvencije_klavir(1);
frekvencije_klavir = diff(frekvencije_klavir);
prosecan_razmak_klavir = mean(frekvencije_klavir);

%TRUBA

[pik_vred,pik_frekv] = findpeaks(amp_truba);

pik_frekv = pik_frekv(find(pik_vred > 0.05*max(
    pik_vred))));
pik_vred = pik_vred(find(pik_vred > 0.05*max(pik_vred)
    ));

pik_vred = pik_vred(3:length(pik_vred));

pik_frekv = pik_frekv*Fs_truba/N_truba;
pik_frekv = pik_frekv(3:length(pik_frekv));

i = 1;
count = 1;

while(i < length(pik_frekv))
    j = i + 1;
    while(j < length(pik_frekv) && abs(pik_frekv(i) -
        pik_frekv(j)) < 100)
        j = j + 1;
    end
    frekvencije_truba(count) = pik_frekv(find(pik_vred
        == max(pik_vred(i:j))));
    count = count + 1;
    i = j + 1;
end

frekvencije_truba;
ton_truba= frekvencije_truba(1);
frekvencije_truba = diff(frekvencije_truba);
prosecan_razmak_truba = mean(frekvencije_truba);

%VIOLINA
```

```
[pik_vred,pik_frekv] = findpeaks(amp_violina);

pik_frekv = pik_frekv(find(pik_vred > 0.05*max(
    pik_vred)));
pik_vred = pik_vred(find(pik_vred > 0.05*max(pik_vred)
    ));

pik_vred = pik_vred(3:length(pik_vred));

pik_frekv = pik_frekv*Fs_violina/N_violina;
pik_frekv = pik_frekv(3:length(pik_frekv));

i = 1;
count = 1;

while(i < length(pik_frekv))
    j = i + 1;
    while(j < length(pik_frekv) && abs(pik_frekv(i) -
        pik_frekv(j)) < 100)
        j = j + 1;
    end
    frekvencije_violina(count) = pik_frekv(find(
        pik_vred == max(pik_vred(i:j))));
    count = count + 1;
    i = j + 1;
end

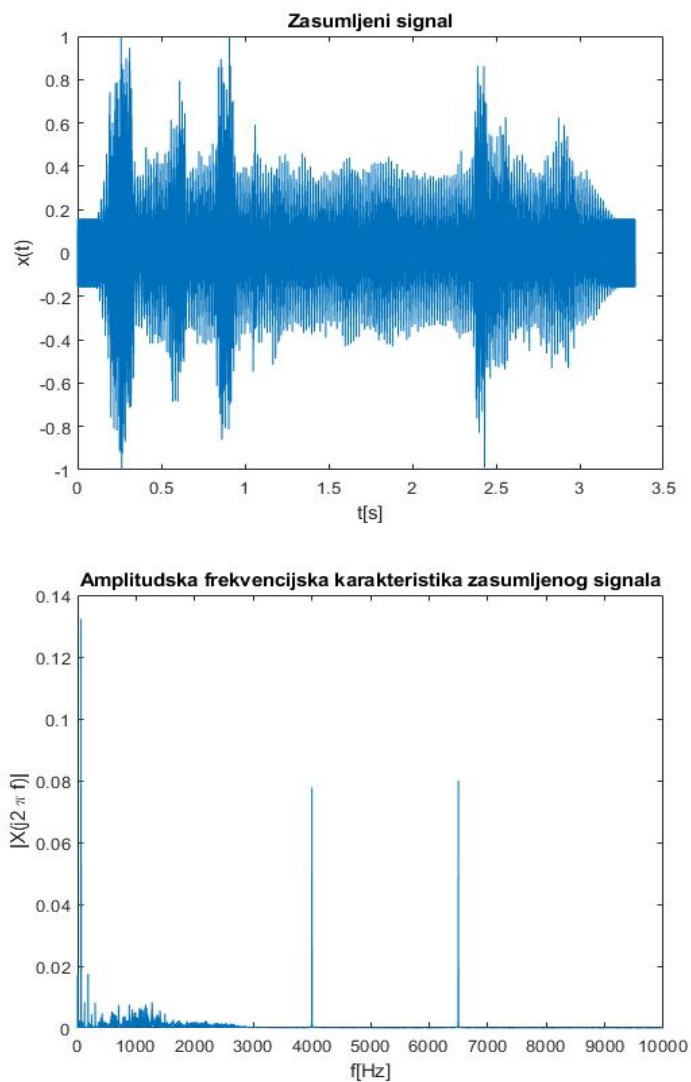
frekvencije_violina;
ton_violina = frekvencije_violina(1);
frekvencije_violina = diff(frekvencije_violina);
prosecan_razmak_violina = mean(frekvencije_violina);

disp(ton_flauta);
disp(ton_klavir);
disp(ton_truba);
disp(ton_violina);

disp(prosecan_razmak_truba);
```

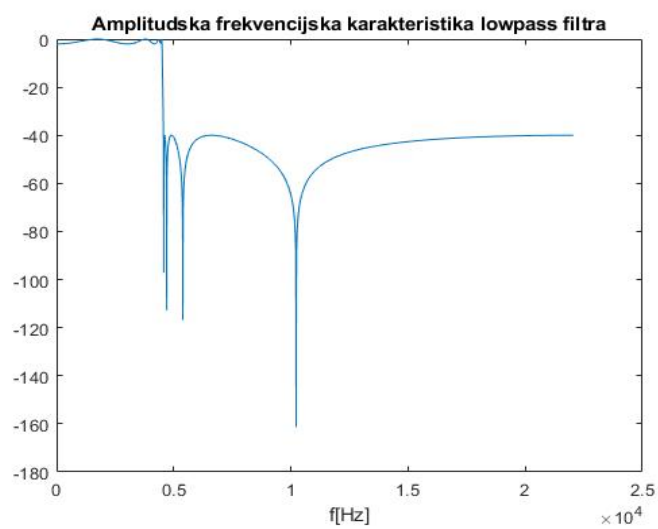
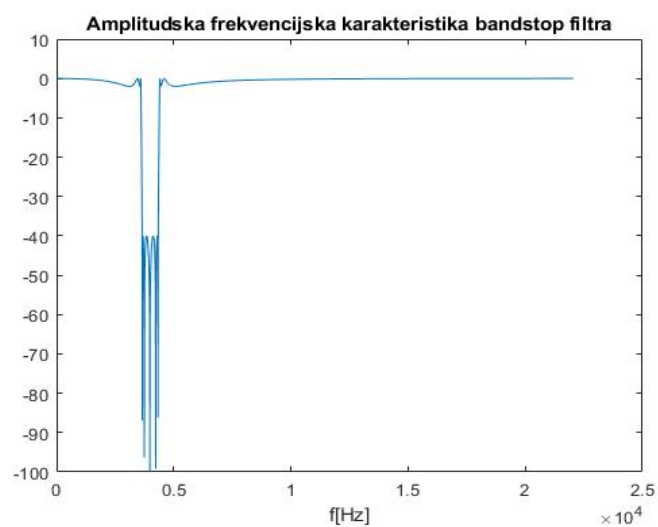
3 Задатак 3

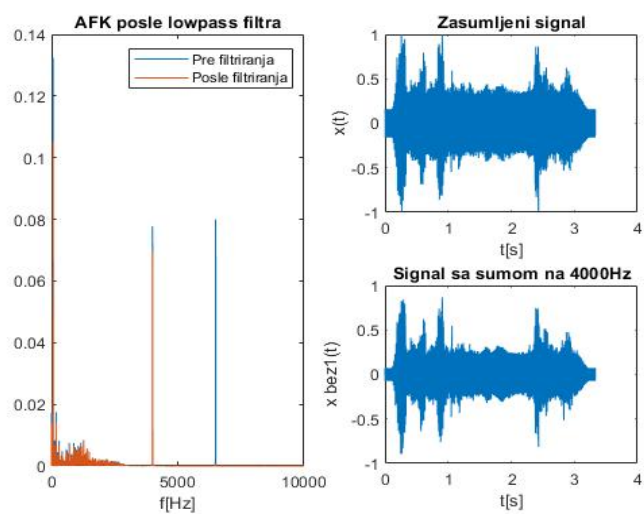
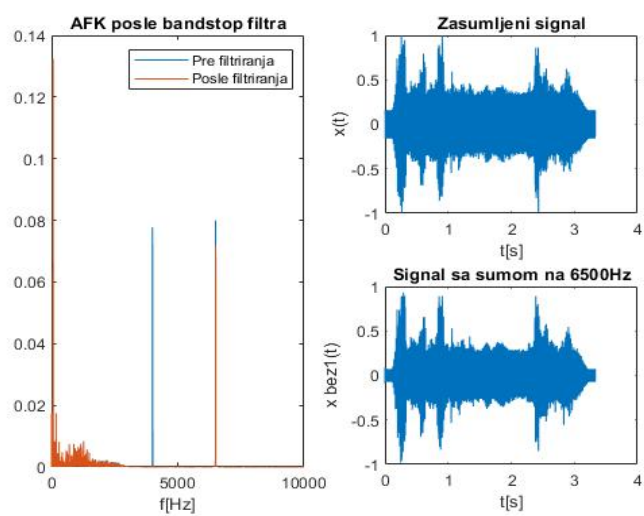
3.1 Prvi zasumljeni

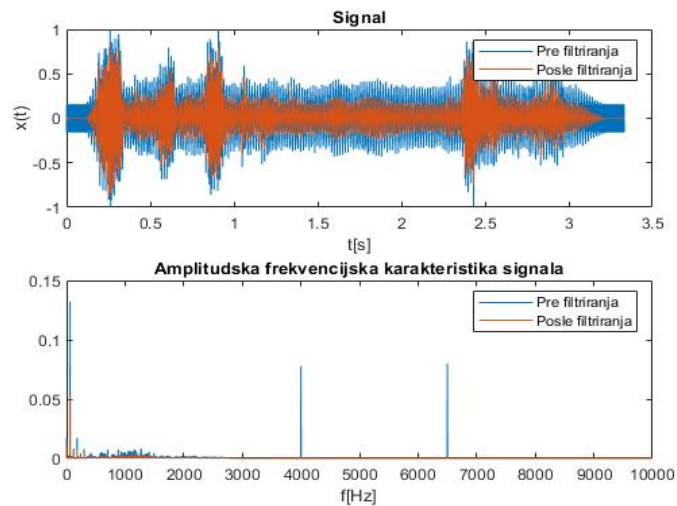


Као што се види синусоидални шумови се налазе на фреквенцијама $4000Hz$ и $6500Hz$.

Конструисана су 2 елиптична филтра(један bandstop који уклања шум на $4000Hz$ и један lowpass који уклања шум на $6500Hz$).







```
%Aleksandar Djordjevic 2019/0086

clear all;
close all
clc

%% Ucitavanje signala

[x, Fs] = audioread('star_wars_zasumljen1.wav');

t = 0:1/Fs:(length(x) - 1)/Fs;

%% Vremenski domen i amplitudska karakteristika

figure(1)
plot(t,x);
xlabel('t[s]');
ylabel('x(t)');
title('Zasumljeni signal');

N = 2^nextpow2(length(x));
f1 = 0:Fs/N:Fs/2;
X = fft(x,N)/length(x);
amp_x = abs(X(1:N/2+1));
amp_x(2:N/2+1) = 2*amp_x(2:N/2+1);

figure(2)
plot(f1,amp_x);
```

```
xlim([0 10^4]);
xlabel('f[Hz]');
ylabel('|X(j2 \pi f)|');
title('Amplitudska frekvencijska karakteristika
      zasumljenog signala');

%% Uklanjanje frekvencij na 4000Hz

% Digitalni elipticki filter(bandstop)

Wp = [3600 4400]/(Fs/2);
Ws = [3500 4500]/(Fs/2);
Rp = 2;
Rs = 40;

[n, Wn] = ellipord(Wp,Ws,Rp,Rs);
[b, a] = ellip(n,Rp,Rs,Wn, 'stop');
[h, f]= freqz(b, a, N/2 + 1, Fs);

figure(3)

plot(f, 20*log10(abs(h)));
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika prvog
      filtra');
ylim([-100 10]);

% Filtriranje signala

x_bez1 = filter(b,a,x);
X_bez1 = fft(x_bez1,N)/length(x_bez1);
amp_x_bez1 = abs(X_bez1(1:N/2+1));
amp_x_bez1(2:N/2+1) = 2*amp_x_bez1(2:N/2+1);

figure(4)

subplot(2,2,[1 3]);
plot(f1,amp_x);
hold on;
plot(f1,amp_x_bez1);
hold off;
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika
      signala');
legend('Pre filtriranja', 'Posle filtriranja');
xlim([0 10^4])
```

```
subplot(2,2,2);
plot(t,x);
xlabel('t[s]');
ylabel('x(t)');
title('Zasumljeni signal');

subplot(2,2,4)
plot(t,x_bez1);
xlabel('t[s]');
ylabel('x_bez1(t)');
title('Signal sa sumom na 6500Hz');

% Uklanjanje suma na frekveniciji 6500Hz

% Digitalni elipticki filter(lowpass)

Wp = 4500/(Fs/2);
Ws = 4600/(Fs/2);
Rp = 2;
Rs = 40;

[n, Wn] = ellipord(Wp,Ws,Rp,Rs);
[b2, a2] = ellip(n, Rp, Rs, Wp);
[h, f] = freqz(b2,a2,N/2+1,Fs);

figure(5)
plot(f, 20*log10(abs(h)));
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika drugog
      filtra')

% Filtriranje signala

x_bez2 = filter(b2,a2,x);
X_bez2 = fft(x_bez2,N)/length(x_bez2);
amp_x_bez2 = abs(X_bez2(1:N/2+1));
amp_x_bez2(2:N/2+1) = 2*amp_x_bez2(2:N/2+1);

figure(6)

subplot(2,2,[1 3]);
plot(f1,amp_x);
hold on;
plot(f1,amp_x_bez2);
hold off;
```

```
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika
      signala');
legend('Pre filtriranja', 'Posle filtriranja');
xlim([0 10^4])

subplot(2,2,2);
plot(t,x);
xlabel('t[s]');
ylabel('x(t)');
title('Zasumljeni signal');

subplot(2,2,4)
plot(t,x_bez2);
xlabel('t[s]');
ylabel('x_ bez1(t)');
title('Signal sa sumom na 4000Hz');

%% Potpuno uklanjanje suma

x_bez_suma = filter(b,a,x);
x_bez_suma = filter(b2,a2,x_bez_suma);

X_bez_suma = fft(x_bez_suma,N)/length(x_bez_suma);
amp_x_bez_suma = abs(X_bez_suma(1:N/2+1));
amp_x_bez_suma(2:N/2+1) = amp_x_bez_suma(2:N/2+1);

figure(7)

subplot(2,2,[1 3]);
plot(f1,amp_x);
hold on;
plot(f1,amp_x_bez_suma);
hold off;
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika
      signala');
legend('Pre filtriranja', 'Posle filtriranja');
xlim([0 10^4])

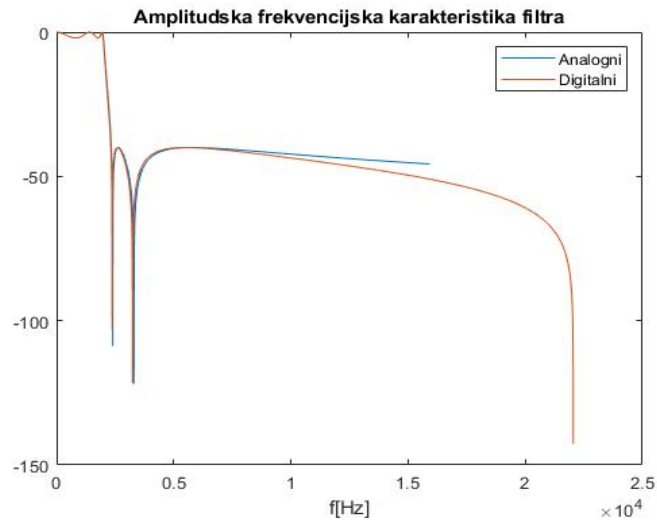
subplot(2,2,2);
plot(t,x);
xlabel('t[s]');
ylabel('x(t)');
title('Zasumljeni signal');
```

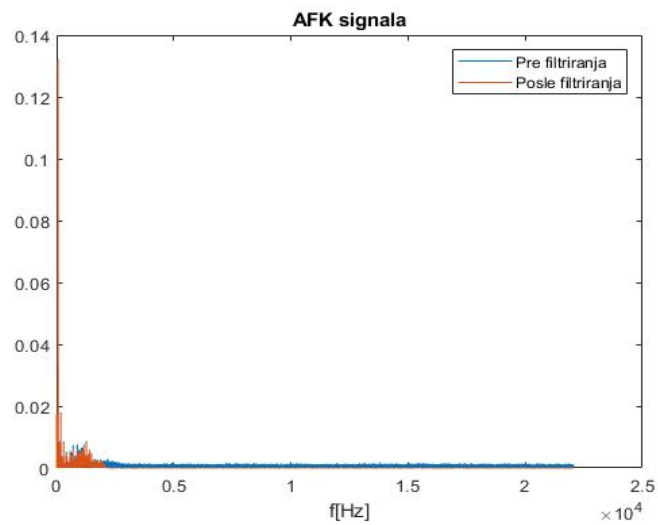
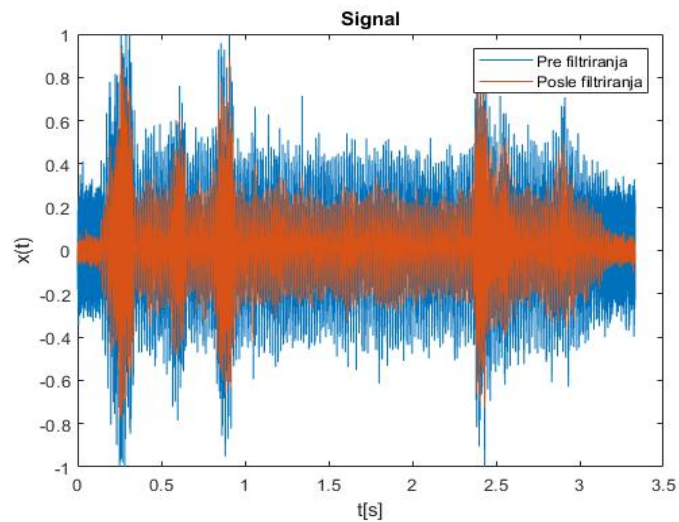
```
subplot(2,2,4)
plot(t,x_bez2);
xlabel('t[s]');
ylabel('x_ bez1(t)');
title('Signal bez suma');

audiowrite('isfiltriran1.wav', x_bez_suma,Fs);
```

3.2 Drugi zasumljeni

Како је Гаусов шум углавном на вишим фреквенцијама користимо lowpass филтар(елиптични). Правимо прво аналогни, а затим га билинеарном методом дигитализујемо.





```
%Aleksandar Djordjevic 2019/0086

%% Ucitavanje signala, vremenski domen i amp
   karakteristika
clear all;
close all;
clc

[x, Fs] = audioread('star_wars_zasumljen2.wav');
```

```
t=0:1/Fs:(length(x) - 1)/Fs;

N = 2^nextpow2(length(x));
f1 = 0:Fs/N:Fs/2;
X = fft(x,N)/length(x);
amp_x = abs(X(1:N/2+1));
amp_x(2:N/2+1) = 2*amp_x(2:N/2+1);

figure(1)

subplot(2,1,1)
plot(t,x)
xlabel('t[s]');
ylabel('x(t)');
title('Zasumljeni signal');

subplot(2,1,2);
plot(f1,amp_x);
xlim([0 0.25*10^4])
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika
      zasumljenog signala');

%% Uklanjanje suma

% Analogni elipticni filter(lowpass)
Wp = 2000*2*pi;
Ws = 2500*2*pi;
Rp = 2;
Rs = 40;

[n,Wn] = ellipord(Wp,Ws,Rp,Rs,'s');
[b, a] = ellip(n,Rp,Rs,Wp,'s');

[h, w] = freqs(b,a,N/2+1);

[bz,az] = bilinear(b,a,Fs);

[hz, fz] = freqz(bz,az,N/2+1,Fs);

figure(2)
plot(w/(2*pi),20*log10(abs(h)));
hold on;
plot(fz,20*log10(abs(hz)));
hold off;
```

```
xlabel('f[Hz]');
title('Amplitudska frekvencijska karakteristika filtra');
legend('Analogni','Digitalni');

% Filtriranje signala
x_bez_suma = filter(bz,az,x);
X_bez_suma = fft(x_bez_suma,N)/length(x_bez_suma);
amp_x_bez_suma = abs(X_bez_suma(1:N/2+1));
amp_x_bez_suma(2:N/2+1) = 2*amp_x_bez_suma(2:N/2+1);

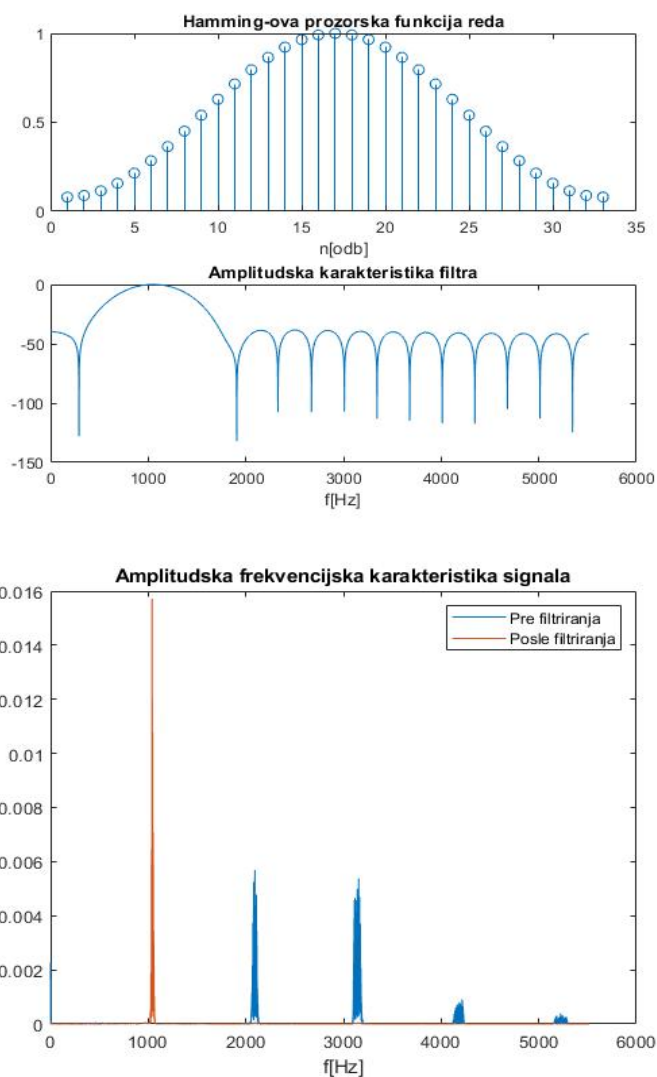
figure(3);
plot(f1,amp_x);
hold on;
plot(f1,amp_x_bez_suma);
hold off;
title('AFK signala');
xlabel('f[Hz]');
legend('Pre filtriranja','Posle filtriranja')

figure(4)
plot(t,x);
hold on;
plot(t,x_bez_suma);
hold off;
title('Signal');
xlabel('t[s]');
ylabel('x(t)');
legend('Pre filtriranja','Posle filtriranja');

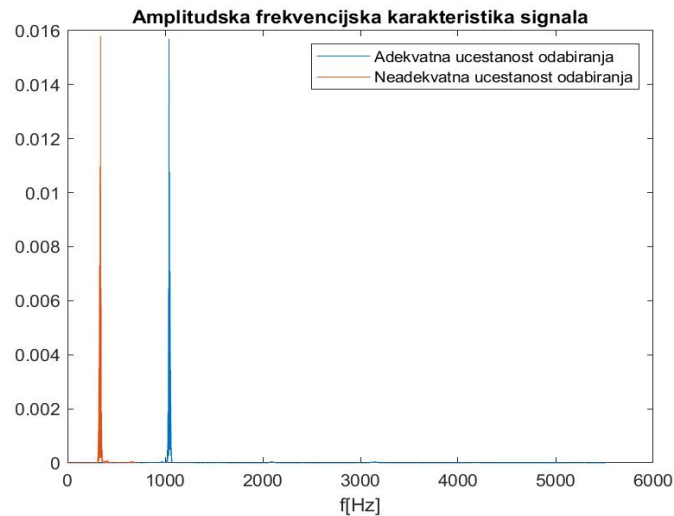
audiowrite('isfiltriran2.wav',x_bez_suma,Fs);
```

4 Задатак 4

Како је размак између пикова релативно велики није било потребе узимати превелики ред FIR филтра. У овом задатку је узет ред филтра 32 и филтар је реализован помоћу Hamming-ове прозорске функције.



Иницијално је фреквенција одабирања била 11025Hz , како је пик на фреквенцији $1039,6\text{Hz}$ то су адекватне учестаности оне веће од 2079.2Hz . Из филтрираног сигнала је узет сваки 8 одабирак и на тај начин је фреквенција одабирања спуштена на осмину полазне вредности односно 1378.125Hz што је очигледно неадекватна учестаност.



Као што видимо пик се видно померио тј. наступила је замена учестаности. Замењена учестаност износи око 340Hz што се поклапа са теоретским предвиђањима.

```
%Aleksandar Djordjevic 2019/0086

%% Ucitavanje signala
clear all;
close all;
clc;

[x, Fs] = audioread('violina_3.wav');

%% Vremenski domen i amplitudska karakteristika
t = 0:1/Fs:(length(x)-1)/Fs;

N = 2^nextpow2(length(x));
f1 = 0:Fs/N:Fs/2;
X = fft(x,N)/length(x);
amp_x = abs(X(1:N/2+1));
amp_x(2:N/2+1) = 2*amp_x(2:N/2+1);

figure(1)

subplot(2,1,1)
plot(t,x);
title('Signal u vremenskom domenu');
xlabel('t[s]');
ylabel('x(t)');
```

```
subplot(2,1,2)
plot(f1,amp_x);
title('Amplitudska frekvencijska karakteristika
      signala');
xlabel('f[Hz]');

%% Izdvajanje prvog pika

% FIR filter sa hamming-ovim prozorom

n = 32;
window = hamming(n+1);
Wn = [1000 1100]/(Fs/2);
b = fir1(n,Wn,window);
a = 1;
[h, f] = freqz(b,a,N/2+1,Fs);

figure(2)

subplot(2,1,1)
stem(window);
xlabel('n[odb]');
title('Hamming-ova prozorska funkcija reda');

subplot(2,1,2)
plot(f, 20*log10(abs(h)));
xlabel('f[Hz]');
title('Amplitudska karakteristika filtra')

% Filtriranje signala

x_fil = filter(b,a,x);
X_fil = fft(x_fil,N)/length(x_fil);
amp_x_fil = abs(X_fil(1:N/2+1));
amp_x_fil(2:N/2+1) = 2*amp_x_fil(2:N/2+1);

figure(3)

plot(f1,amp_x);
hold on;
plot(f1,amp_x_fil);
title('Amplitudska frekvencijska karakteristika
      signala');
xlabel('f[Hz]');
legend('Pre filtriranja','Posle filtriranja');
```

```
audiowrite('violina_3_fil.wav',x_fil,Fs);

%% Zadatak4 pod b)

m = 8;

x_fil_zam = zeros(1,round(length(x_fil)/m));
count = 1;

for i =1:length(x_fil)
    if(mod(i,m) == 0)
        x_fil_zam(count) = x_fil(i);
        count = count + 1;
    end
end

X_fil_zam = fft(x_fil_zam,N)/length(x_fil_zam);
amp_x_fil_zam = abs(X_fil_zam(1:N/2+1));
amp_x_fil_zam(2:N/2+1) = 2*amp_x_fil_zam(2:N/2+1);

f2 = linspace(0,Fs/16,N/2+1);

figure(4)
plot(f1,amp_x_fil);
hold on;
plot(f2,amp_x_fil_zam);
hold off;
title('Amplitudska frekvencijska karakteristika
       signala');
xlabel('f[Hz]');
legend('Adekvatna ucestanost odabiranja', 'Neadekvatna
       ucestanost odabiranja');

audiowrite('violina_3_zam.wav',x_fil_zam,Fs);
```