



Univerzitet u Beogradu -Elektrotehnički fakultet
Katedra za Signale i sisteme



13E053DOS - Digitalna obrada signala

Domaći zadatak

STUDENT

Tijana Aleksić

BROJ INDEKSA

2018/0455

PARAMETRI

$P = 3 \quad Q = 2 \quad R = 1 \quad S = 0$

Decembar 2020.

Zadatak 1 • Parametar $P = 3$

- Analitički oblik signala $x[n]$ i $y[n]$ i broj tačaka N

$$x[n] = \begin{cases} n & , 0 \leq n \leq \frac{N}{2} - 1 \\ 1 - n - \frac{N}{2} & , \frac{N}{2} \leq n \leq N - 1 \end{cases}$$

$$y[n] = \begin{cases} 2\cos((P+1)n + \frac{\pi}{4}) & , 0 \leq n \leq \frac{N}{2} - 1 \\ 0 & , \text{inače} \end{cases}$$

$$N = 10 * (3 + 1) = 40$$

- Prikaz signala $x[n]$ i $y[n]$

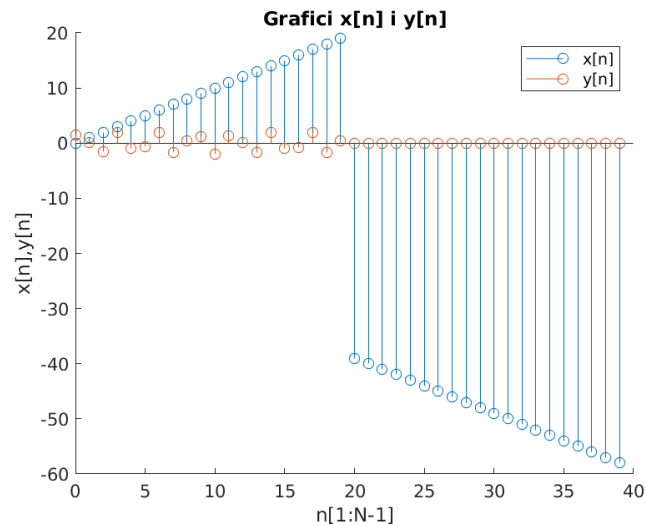


Figure 1: Prikaz signala $x[n]$ i $y[n]$

- Prikaz rezultata linearne konvolucije i rezultata funkcije conv

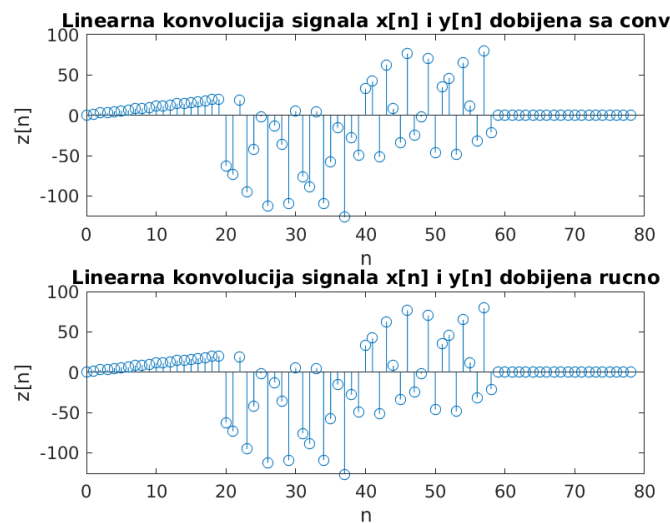


Figure 2: Linearna konvolucija signala $x[n]$ i $y[n]$

- *Prikaz rezultata ciklične konvolucije i rezultata funkcije cconv*

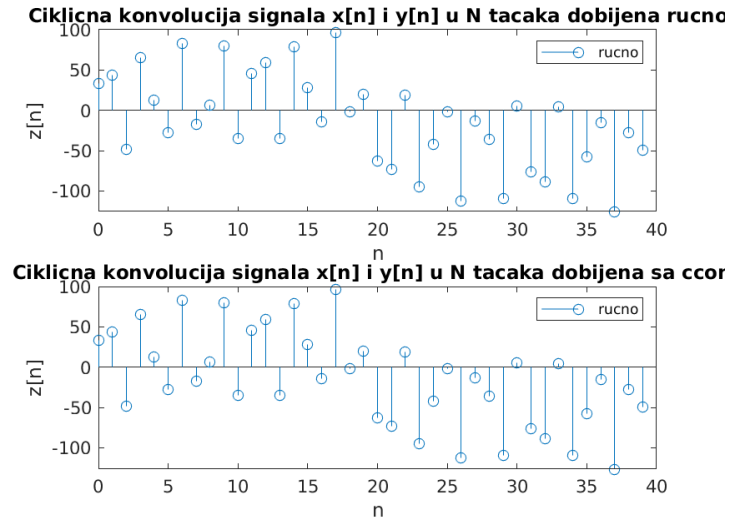


Figure 3: Ciklična konvolucija signala $x[n]$ i $y[n]$

- *Linearna i ciklična konvolucija imaju iste vrednosti u sledećim odbircima*

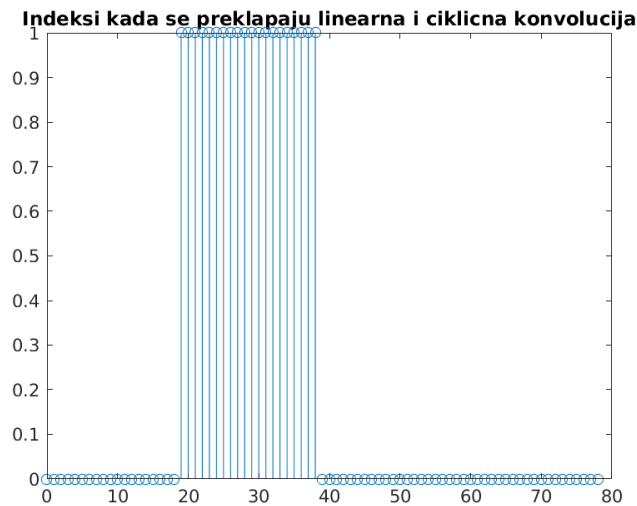


Figure 4: Preklapanja linearne i ciklične konvolucije signala $x[n]$ i $y[n]$ pri čemu je ciklična konvolucija produžena do kraja sa vrednostima ciklične konvolucije

- *Programski kôd*

```

1      P=3;
2      N=(P+1)*10;
3
4      %kreiranje niza x
5      x=0:N-1;
6      for i=N/2:N-1
7          x(i+1)=1-i-N/2;
8      end

```

```

9  %kreiranje niza y[n]
10 t=0:N-1;
11 y=zeros(1,N);
12 for i=0:N-1;
13     if(i<N/2)
14         y(i+1)=2*cos((P+1)*i*pi/4);
15     end
16 end
17
18
19 figure(1),
20 subplot(2,1,1);
21 stem(t,x);
22 ylabel('x[n]'), xlabel('n[1:N-1]'), title('Grafici x[n] i y[n]');
23 subplot(2,1,2);
24 stem(t,y);
25 ylabel('y[n]'), xlabel('n[1:N-1]'), title('Grafici x[n] i y[n]');
26 saveas(figure(1), 'zad.1.xy.png');
27
28
29 %linearna kovolucija
30
31
32 %rucno racunata
33 xp=cat(2,x,zeros(1,length(x)));
34 yp=cat(2,y,zeros(1,length(y)));
35 lin_rucno=zeros(1,length(x)+length(y)-1);
36 for n=1:length(lin_rucno)
37     for k=1:n
38         lin_rucno(n)=lin_rucno(n)+xp(k)*yp(n-k+1);
39     end
40 end
41
42
43 lin_conv=conv(x,y);
44 tlin=0:length(lin_conv)-1;
45 figure(2);
46 hold all;
47
48 stem(tlin,lin_conv);
49 ylabel('x[n],y[n]'), xlabel('n[1:N-1]'), title('Grafici x[n] i y[n]');
50 legend('x[n]','y[n]');
51 stem(tlin,lin_rucno);
52 ylabel('z[n]'), xlabel('n'), title('Linearna konvolucija signala x[n] i y[n]');
53 saveas(figure(2), 'zad.1.lin.konv.png');
54 legend('ugradjeno','rucno')
55
56
57 %ciklicna konvolucija
58
59 %rucno kucana;
60 cikl_rucno=zeros(1,length(x));
61 for n=1:length(cikl_rucno)
62     for k=1:length(cikl_rucno)
63         cikl_rucno(n)=cikl_rucno(n)+x(k)*y(mod(n-k,N)+1);
64     end
65 end
66
67
68 cikl_conv=cconv(x,y,N);
69 tcik=0:length(cikl_conv)-1;
70 figure(3);
71 hold all;
72 stem(tcik,cikl_rucno);
73 stem(tcik,cikl_conv);
74 ylabel('z[n]'), xlabel('n'), title('Ciklicna konvolucija signala x[n] i y[n] u N tacaka');
75 legend('rucno','ugradjeno')
76 saveas(figure(3), 'zad.1.cik.konv.png')

```

```
77
78 %preklapanje ciklicne i linearne konvolucije
79 for i=1:max(length(lin_conv),length(cikl_conv))
80     if(i<min(length(lin_conv),length(cikl_conv)))
81         cc_to_compare(i)=cikl_conv(i);
82     else
83         cc_to_compare(i)=cikl_conv(i-min(length(lin_conv),length(cikl_conv))+1);
84     end
85 end
86
87 preklapanja=(cc_to_compare-lin_conv<0.00001);
88 for i=1:length(preklapanja)
89     if(preklapanja(i)==1)
90         fprintf('Preklapaju se u %i \n',i);
91     end
92 end
93 yosa=0:78;
94 figure(4)
95 stem(yosa,preklapanja);
96 title('Indeksi kada se preklapaju linearna i ciklicna konvolucija');
97 saveas(figure(4),'ZAD.1.preklapanja.png');
```

Zadatak 2 • Parametar $Q = 2$

- Prikaz originalnog signala i signala dobijenog odabiranjem sa f_s iz tabele

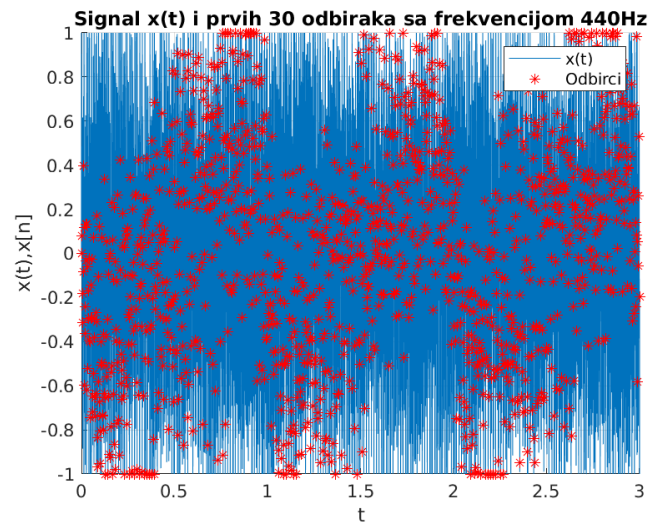


Figure 5: Prikaz signala $x(t)$ i odbirci signala sa zadatom frekvencijom na celom intervalu

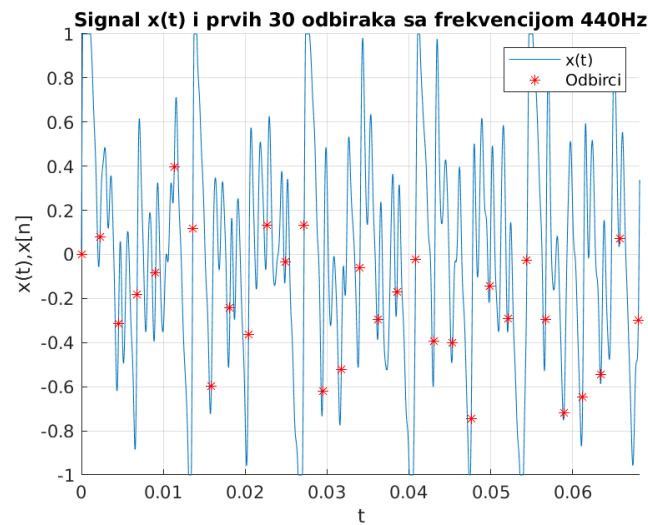


Figure 6: Prikaz signala $x(t)$ i odbirci signala sa zadatom frekvencijom na manjem intervalu

- *Amplitudska frekvencijska karakteristika originalnog signala*



Figure 7: Amplitudska frekvencijska karakteristika originalnog signala

- *Signal se sastoji od sledećih tonova*

Data je frekvencijska karakteristika od 200 do 450 Hz. Sa date slike mozemo primetiti da postoje tri tona jer je ton na 220 Hz isti kao i na 440Hz jer je celobrojni umnozak pa samim tim odredjuje boju zvuka a ne i njegovu visinu.

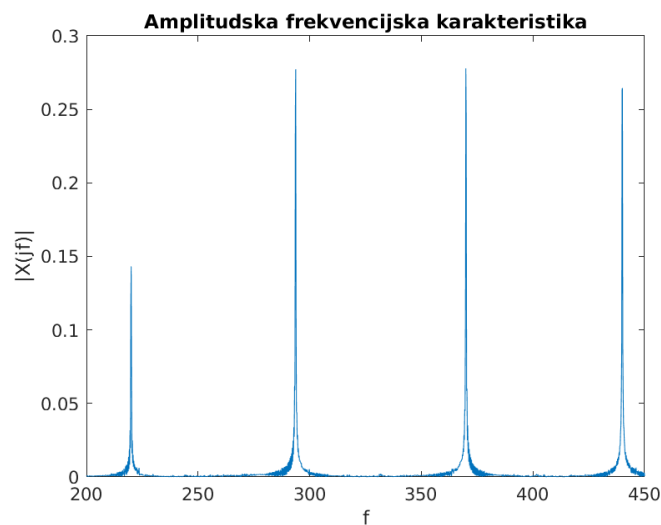


Figure 8: Prikaz signala $x[n]$ i $y[n]$

Izlaz koda koji se pokrece je:

Ton koji je odsviran je **d**

Ton koji je odsviran je **fis**

Ton koji je odsviran je **a**

- Programski kôd

```

1      clear all;
2      close all;
3      clc;
4
5      %parametar Q=2
6
7      filename='audio2.wav';
8      [x,Fs]=audioread(filename);
9      t=0:1/Fs:(length(x)-1)/Fs;
10     %plot(t,x);
11     %sound(x,Fs);
12     Fs1=440;
13     %aproksimirano koji je korak odabira za sempling od 440Hz
14     apr=round(Fs/Fs1);
15     j=1;
16     for i=1:apr:length(x)
17         y(j)=x(i);
18         ty(j)=t(i);
19         j=j+1;
20     end
21
22
23     figure(1);
24     hold all;
25     plot(t,x);
26     scatter(ty,y,'r*');
27     grid on;
28     xlim([0 30/Fs1]);
29     xlabel('t'),ylabel('x(t),x[n]'),title('Signal x(t) i prvih 30 odbiraka sa ...
        frekvencijom 440Hz'),legend('x(t)', 'Odbirci')
30     saveas(figure(1), 'ZAD.2-signal.i.odabran.png')
31
32     figure(4);
33     hold all;
34     plot(t,x);
35     scatter(ty,y,'r*');
36     grid on;
37     xlabel('t'),ylabel('x(t),x[n]'),title('Signal x(t) i prvih 30 odbiraka sa ...
        frekvencijom 440Hz'),legend('x(t)', 'Odbirci')
38     saveas(figure(4), 'ZAD.2-signal.i.odabran.ceo.png')
39
40     %amplitudska karakteristika
41
42     N=2^nextpow2(length(x));
43     X=fft(x,N)/length(x);
44     X1=abs(X(1:N/2+1));
45     X1(2:N/2+1)=2*X1(2:N/2+1);
46     f1=0:Fs/N:Fs/2;
47
48     figure(2);
49     plot(f1,X1)
50     xlabel('f'),ylabel('|X(jf)|'),title('Amplitudska frekvencijska karakteristika');
51     saveas(figure(2), 'ZAD.2-amp.karakteristika.png')
52
53     %deo zadatka sa pronalazenjem frekvencija
54
55     for i=1:12
56         mult(i)=2^(i/12);
57         freq(i)=220*mult(i);
58     end
59
60     %
61     tones=["b", "h", "c", "cis", "d", "dis", "e", "f", "fis", "g", "gis", "a"];
62     %vrednost od 0.17 odredjena je na osnovu spektra u opsegu 220 do 440 Hz gde
63     %se nalazi jedna cela oktava od a do a1(od a u velikoj do a u maloj oktavi)

```



```
64 finds=find(X1>0.17);
65 for i=1:2:length(finds)
66     for j=1:length(freq)
67         if((f1(finds(i))<freq(j)) && (f1(finds(i+1))>freq(j)))
68             fprintf('Ton koji je odsviran je %s \n',tones(j));
69         end
70     end
71 end
72 figure(3)
73 plot(f1,X1)
74 xlabel('f'),ylabel('|X(jf)|'),title('Amplitudska frekvencijska karakteristika');
75 xlim([200 450])
76 saveas(figure(3),'ZAD_2_amp_od_200_do_450Hz.png')
77 %tonovi su a d fis
```

Zadatak 3 • Parametar $R = 1$

- EKG signal u vremenskom domenu

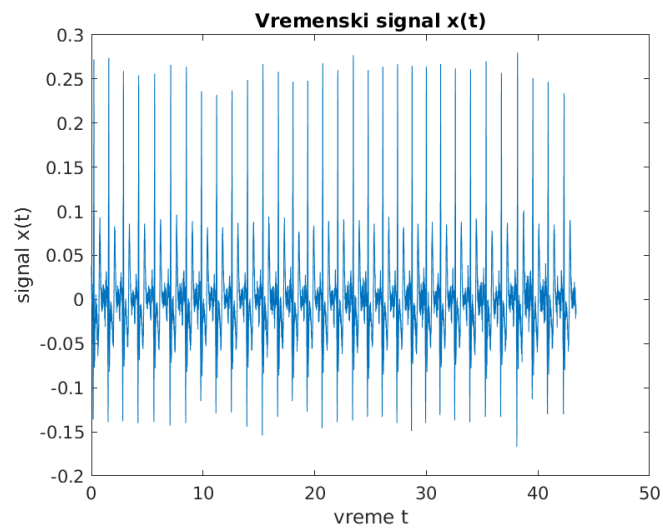


Figure 9: Prikaz signala $x(t)$

- Amplitudska frekvencijska karakteristika EKG signala

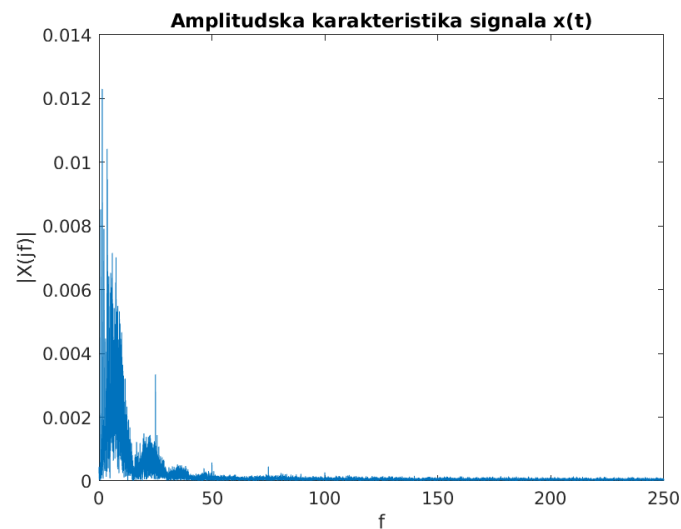


Figure 10: Amplitudska frekvencijska karakteristika EKG signala

- Amplitudske frekvencijske karakteristika analognih i digitalnih filtara

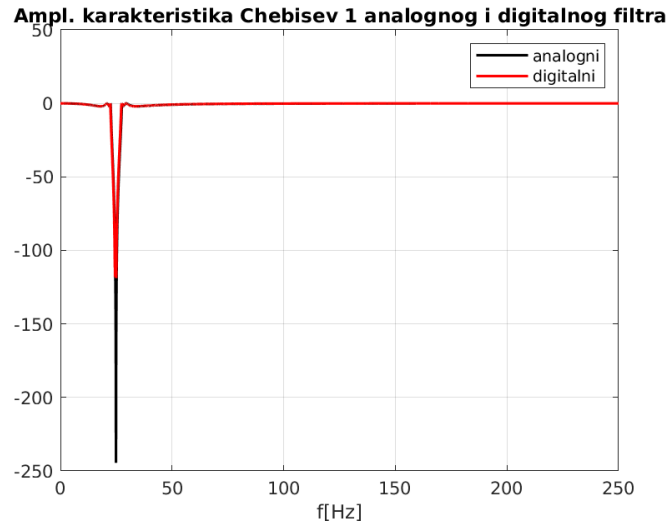


Figure 11: Amplitudske frekvencijske karakteristike analognog i digitalnog Chebisev 1 filtra

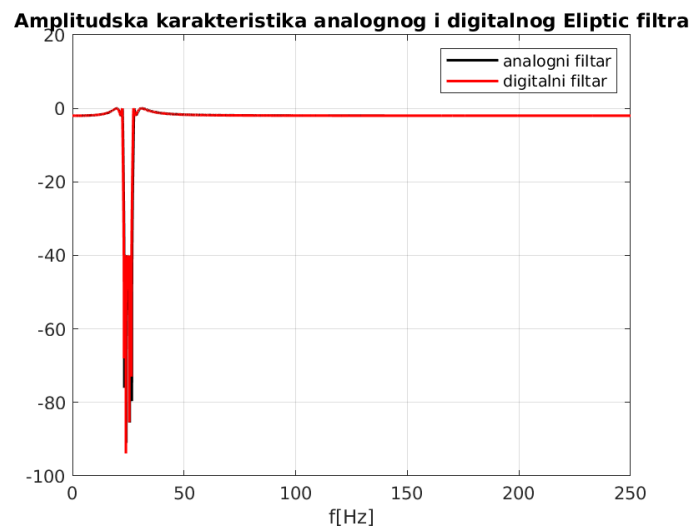


Figure 12: Amplitudske frekvencijske karakteristike analognog i digitalnog Eliptic filtra

- *Originalni i filtrirani signali u vremenskom domenu*

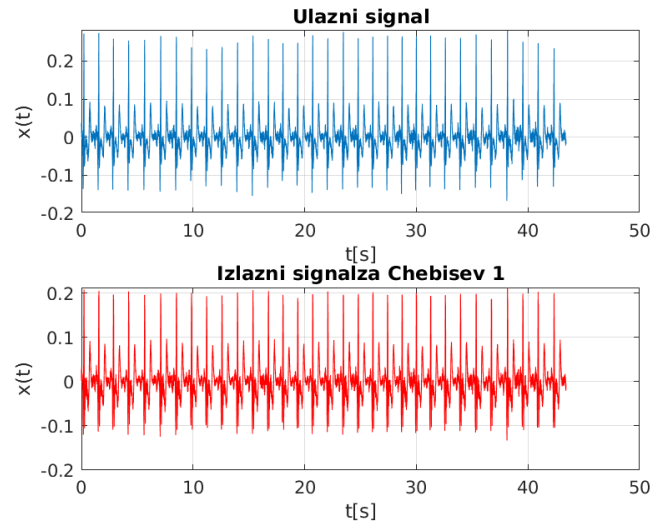


Figure 13: Originalni i filtriran signal u vremenskom domenu nakon Chebisev 1 filtra

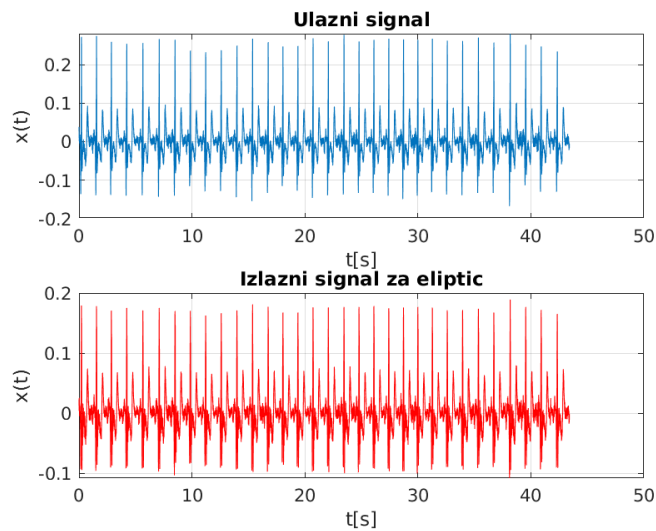


Figure 14: Originalni i filtriran signal u vremenskom domenu nakon Eliptic filtra

- Amplitudske frekvencijske karakteristika originalnog i filtriranih signala

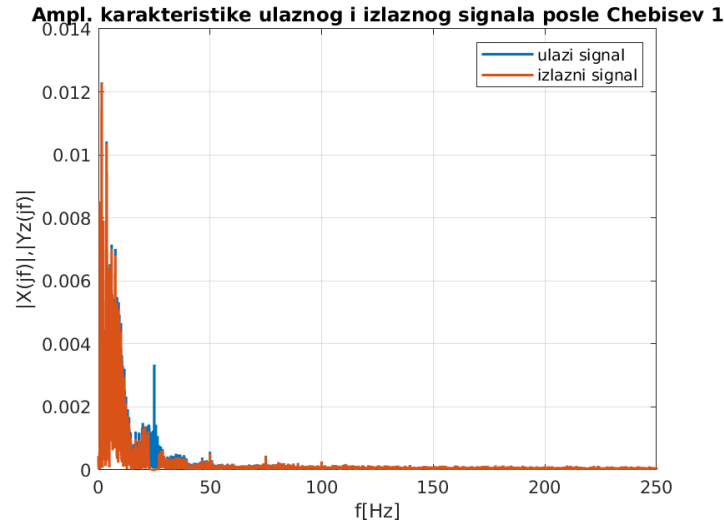


Figure 15: Amplitudske frekvencijske karakteristike originalnog i filtriranog signala nakon Chebisev 1 filtra

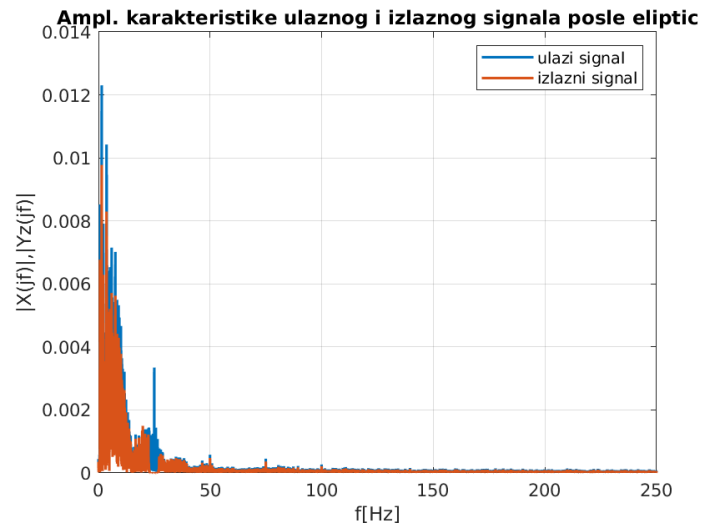


Figure 16: Amplitudske frekvencijske karakteristike originalnog i filtriranog signala nakon Chebisev 1 filtra

- Programski kôd

```

1      %Eliptic i Chebishev 1
2      clear all;close all;clc;
3
4      %parametar za koji se radi zadatak
5      R=1;
6      %ucitavanje signala koji se obradjuje
7      data=load(['ekg' num2str(R) '.mat']);
8      x=data.x;
9      fs=data.fs;
10     %vremenski domen
11     t=0:1/fs:(length(x)-1)/fs;
12     figure(1);

```

```

13 plot(t,x);
14 %xlim([0 4])
15 xlabel('vreme t'),ylabel('signal x(t)'),title('Vremenski signal x(t)');
16 saveas('figure(1)','ZAD.3.vremenski_signal.png');
17
18 %frekvencijski domen
19
20 N=2^nextpow2(length(x));
21 X=fft(x,N)/length(x);
22 X1=abs(X(1:N/2+1));
23 X1(2:N/2+1)=2*X1(2:N/2+1);
24 f1=0:fs/N:fs/2;
25
26
27 figure(2);
28 plot(f1,X1);
29 xlabel('f'),ylabel('|X(jf)|'),title('Amplitudska karakteristika signala x(t)');
30 saveas('figure(2)','ZAD.3.amplitudska-karakteristika.png');
31
32
33
34 %chebishev type 1;
35
36
37 %frekvencija koju treba da ne propustimo:
38 Frekv=25;%[Hz]
39
40 %pravljenje parametara za Cheby filter
41
42 %opsezi u kojima ce biti zakrivljenje filtra
43 Wp1=[90*Frekv/100 110*Frekv/100]*2*pi;
44 Ws1=[80*Frekv/100 120*Frekv/100]*2*pi;
45 %maksimalno slabljenje u propusnom opsegu
46 Rp1=2;%[dB]
47 %minimalno slabljenje u nepropusnom opsegu
48 Rs1=40;%[dB]
49
50 %pravljenje analognog filtra i izracunavanje reda n1 i opsega Wn1
51 [n1, Wn1]=cheblord(Wp1,Ws1,Rp1,Rs1,'s');
52 %izracunavanje parametara filtra a i b koji se prosledjuju funkciji filter
53 [b1, a1] = cheby1(n1,Rp1,Wp1,'stop','s');
54
55 %diskretizacija bilinearnom metodom
56 fp=10;
57 [bz,az]=bilinear(b1,a1,fs,fp);
58
59 %odziv analognog filtra
60 [h1, w1]=freqs(b1,a1,N/2+1);
61 %odziv digitalnog filtra
62 [hz,fz]=freqz(bz,az,N/2+1,fs);
63
64
65 %amplitudska frekvencijska karakteristika filtra
66 figure(3)
67 plot(w1/(2*pi),20*log10(abs(h1)),'k-','Linewidth',1.5);hold on;
68 plot(fz,20*log10(abs(hz)),'r','LineWidth',1.5);
69 xlabel('f[Hz]');grid on;
70 title('Ampl. karakteristika Chebishev 1 analognog i digitalnog filtra');
71 legend('analogni','digitalni')
72 saveas('figure(3)','ZAD.3.afk-aid-cheb1.png')
73
74 %filtriranje x
75
76
77 yz=filter(bz,az,x);
78 ya=filter(b1,a1,x);
79 %digitalan filter;
80 Yz=fft(yz,N)/length(yz);

```

```

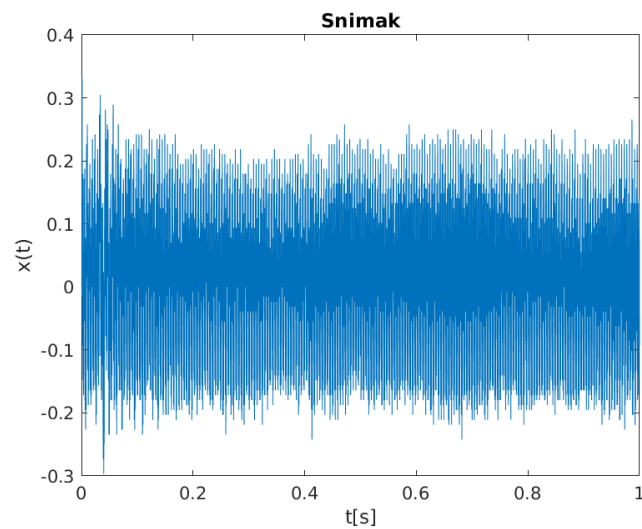
81 Yz1=abs(Yz(1:N/2+1));
82 Yz1(2:N/2+1)=2*Yz1(2:N/2+1);
83 %analogni filter
84 Y=fft(ya,N)/length(ya);
85 Ya1=abs(Y(1:N/2+1));
86 Ya1(2:N/2+1)=2*Ya1(2:N/2+1);
87
88 %crtanje karakteristika ulaznog i izlaznog
89 figure(4)
90 plot(f1,X1,'Linewidth',1.5);hold on;
91 plot(f1,Yz1,'Linewidth',1.5);
92 xlabel('f[Hz]');ylabel('|X(jf)|,|Yz(jf)|');grid on;
93 title('Ampl. karakteristike ulaznog i izlaznog signala posle Chebisev 1');
94 legend('ulazi signal','izlazni signal');
95 saveas(fgure(4),'ZAD.3.akf-aid.uui.png');
96
97
98 figure(5)
99 subplot(2,1,1)
100 plot(t,x);
101 xlabel('t[s]');ylabel('x(t)');grid on;title('Ulazni signal');
102 subplot(2,1,2)
103 plot(t,yz,'r');
104 xlabel('t[s]');ylabel('x(t)');grid on;title('Izlazni signalza Chebisev 1');
105 saveas(fgure(5),'ZAD.3.ulaz-izlaz.cheb.png');
106
107
108
109
110
111 %elliptic
112
113 [n2,Wn2]=ellipord(Wp1,Ws1,Rp1,Rs1,'s');
114 [b2,a2]=ellip(n2,Rp1,Rs1,Wp1,'stop','s');
115
116 %diskretizacija bilinearnom metodom
117 fp2=10;
118 [bze,aze]=bilinear(b2,a2,fs,fp2);
119
120 %odziv analognog filtra
121 [h2,w2]=freqs(b2,a2,N/2+1);
122 %odziv digitalnog filtra
123 [hze,fze]=freqz(bze,aze,N/2+1,fs);
124
125
126 %amplitudska frekvencijska karakteristika filtra
127 figure(7)
128 plot(w1/(2*pi),20*log10(abs(h2)),'k-','Linewidth',1.5);hold on;
129 plot(fze,20*log10(abs(hze)),'r','LineWidth',1.5);
130 xlabel('f[Hz]');grid on;
131 title('Amplitudska karakteristika analognog i digitalnog Eliptic filtra');
132 legend('analogni filter','digitalni filter');
133 saveas(fgure(7),'ZAD.3.afk-aid-elliptic.png');
134
135 %filtriranje x
136 yze=filter(bze,aze,x);
137 yae=filter(b2,a2,x);
138 Yze=fft(yze,N)/length(yze);
139 Yz1e=abs(Yze(1:N/2+1));
140 Yz1e(2:N/2+1)=2*Yz1e(2:N/2+1);
141
142 Ye=fft(yae,N)/length(yae);
143 Yale=abs(Ye(1:N/2+1));
144 Yale(2:N/2+1)=2*Yale(2:N/2+1);
145
146 figure(8)
147 plot(f1,X1,'Linewidth',1.5);hold on;
148 plot(f1,Yz1e,'Linewidth',1.5);

```

```
149 xlabel('f[Hz]');ylabel('|X(jf)|,|Yz(jf)|');grid on;
150 title('Ampl. karakteristike ulaznog i izlaznog signala posle eliptic');
151 legend('ulazi signal','izlazni signal');
152 saveas(ffigure(8),'ZAD.3_afk_ulaz_izlaz_eliptic.png');
153
154 figure(9)
155 subplot(2,1,1)
156 plot(t,x);
157 xlabel('t[s]');ylabel('x(t)');grid on;title('Ulazni signal');
158 subplot(2,1,2)
159 plot(t,yze,'r');
160 xlabel('t[s]');ylabel('x(t)');grid on;title('Izlazni signal za eliptic');
161 saveas(ffigure(9),'ZAD.3_ulaz_izlaz_eliptic.png');
```


Zadatak 4 • Parametar $S = 0$

- *Snimak samoglasnika u vremenskom domenu*

Figure 17: Prikaz signala $x(t)$

- *Usrednjena amplitudska frekvencijska karakteristika snimljenog signala*

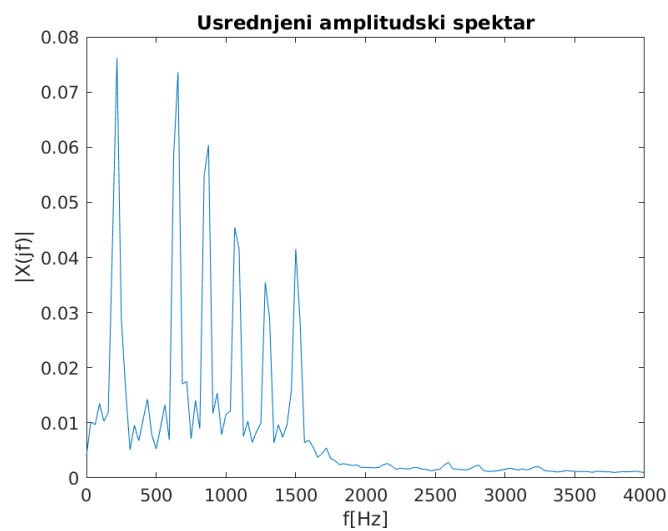


Figure 18: Usrednjena amplitudska frekvencijska karakteristika snimljenog signala

- *Amplitudske frekvencijske karakteristike projektovanih filtara*

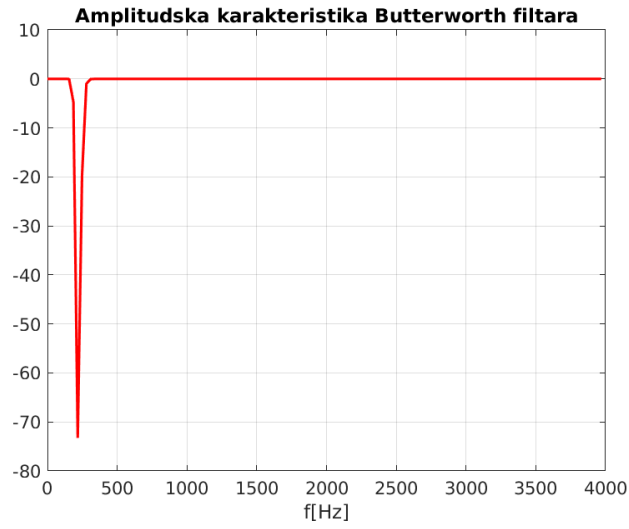


Figure 19: Amplitudska frekvencijska karakteristika projektovanog Butterworth filtra

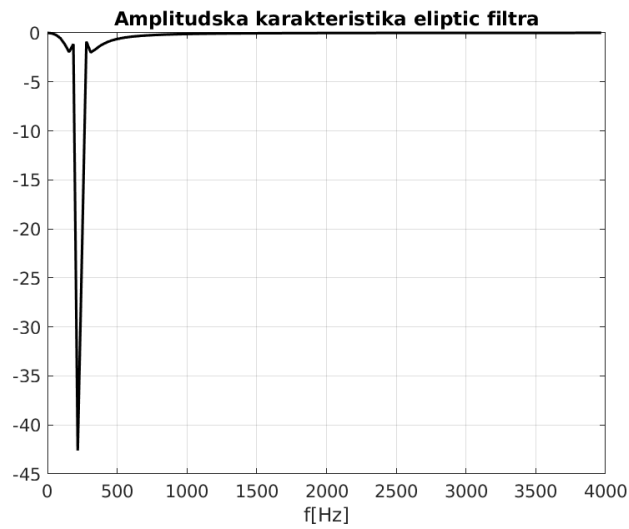


Figure 20: Amplitudska frekvencijska karakteristika projektovanog Eliptic filtra

- *Originalni i filtrirani signali u vremenskom domenu*

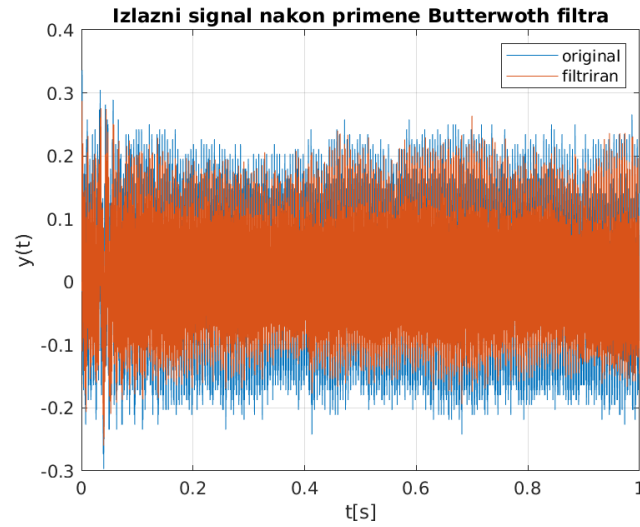


Figure 21: Originalan i filtriran signal pomocu Butterwoth filtra u vremenskom domenu

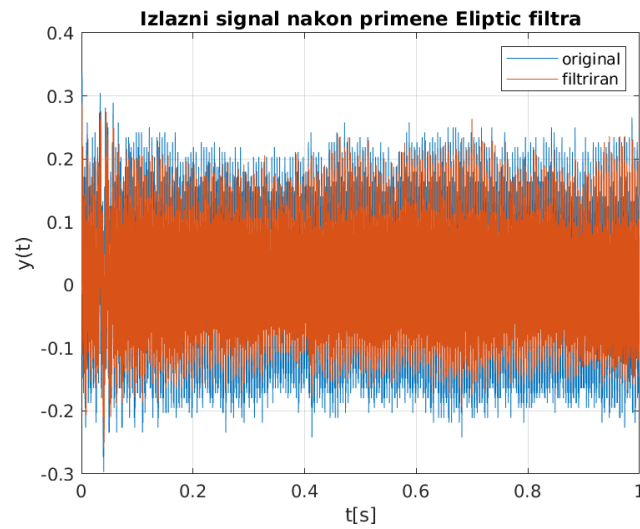


Figure 22: Originalan i filtriran signal pomocu Eliptic filtra u vremenskom domenu

- Amplitudske frekvencijske karakteristike originalnog i filtriranog signala

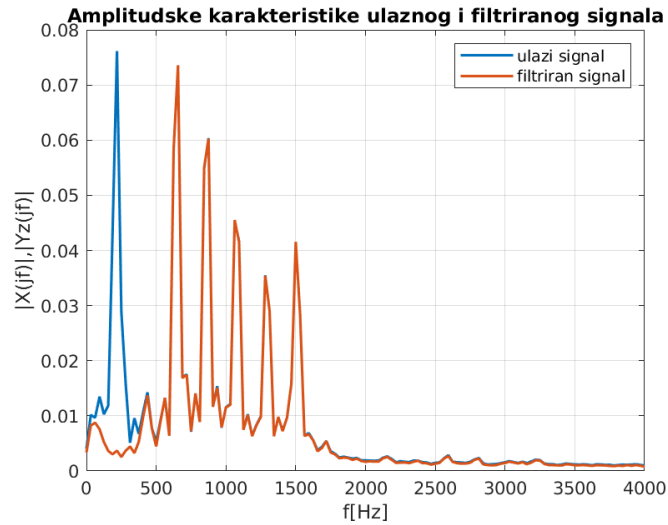


Figure 23: Amplitudske frekvencijske karakteristike originalnog i filtriranog signala sa Butterwoth filtrom

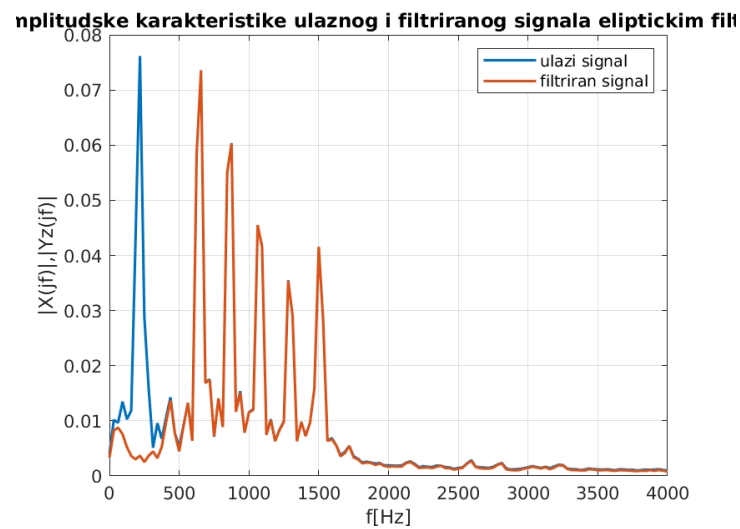


Figure 24: Amplitudske frekvencijske karakteristike originalnog i filtriranog signala sa Eliptic filtrom

- *Diskusija za razliciti red filtra*

Na sledecoj slici nalaze se dva filtra isprojektovana tako da imaju razlicit red filtra. Uticaj reda filtra je sledeci: Ako je red filtra mali, filter je 'losiji' jer ne moze sve da ponisti sto zelimo da ne utice na signal, ali povecavanjem reda filtra rizikujemo stabilnost filtra koji se projektuje.

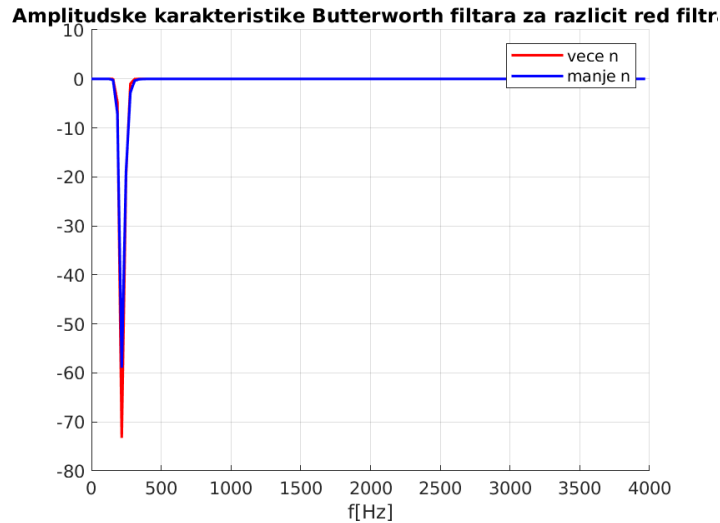


Figure 25: Amplitudske frekvencijske karakteristike projektovanog Butterwoth filtra sa razlicitim redom filtra

- *Programski kôd*

Kod koriscen za pravljenje .wav fajla

```
1 filename='snimak1.wav';
2 Fs=8000;
3 recObj=audiorecorder;
4 disp('Start speaking. ');
5 t=1; %1 sekund
6 recordblocking(recObj,t);
7 disp('End of Recording');
8 x=getaudiodata(recObj);
9 play(recObj);
10 audiowrite(filename,x,Fs);
```

Kod koriscen za obradu trazenu u zadatku

```
1 clear all;close all;clc;
2
3 %definisanje imena fajla koji se ucitava,prethodno snimljen u snimanje.m
4 filename='snimak1.wav'
5 [x,fs]=audioread(filename);
6 %definisanje vremenske ose da bi se nacrtao snimak
7 t=0:1/fs:(length(x)-1)/fs;
8
9 %iscrtavanje originalnog signala koji je ucitan iz snimak.wav
10 figure(1)
11 plot(t,x)
12 xlabel('t[s]'),ylabel('x(t)'),title('Snimak');
13 saveas(figure(1),'ZAD.4.original.png');
14 %test primer za funkciju sa kraja spectar_of_20_ms koja vraca amplitudsku
15 %karakteristiku signala koji je duzine 20ms.Duzina od 160 je 8000/50 koliko
16 %ima odbiraka u 20 ms;
17 x120=x(1:161);
```

```

18 [Sp1,fpl]=spectar.of_20.ms(x120,fs);
19 ssr=zeros(1,length(Sp1));
20
21
22 %usrednjavanje spektra originalnog signala
23 for i=0:49
24     [spkr,fsp]=spectar.of_20.ms(x(160*i+1:160*(i+1)),fs);
25     ssr=ssr+spkr;
26 end
27 ssr=ssr./50;
28 %iscrtavanje usrednjenog spektra signala na ulazu svih 50 puta
29 figure(2)
30 plot(fsp,ssr);
31 xlabel('f[Hz]'),ylabel('|X(jf)|'),title('Usrednjeni amplitudski spektar');
32 saveas(figure(2),'ZAD.4-original.usrednjen-spektar.png');
33 N=2^nextpow2(160);
34
35
36 %Butterworth;
37
38 %frekvencija koju treba da ne propustimo:
39 Frekv=220;%[Hz]
40
41 %opsezi u kojima ce biti zakrivljenje filtra
42 Wp1=[85*Frekv/100 120*Frekv/100]*2/fs;
43 Ws1=[30*Frekv/100 180*Frekv/100]*2/fs;
44 %drugi opseg je da bi se smanjio red filtra radi objasnjenja
45 Ws11=[0.1*Frekv/100 250*Frekv/100]*2/fs;
46
47
48 %maksimalno slabljenje u propusnom opsegu
49 Rp1=2;%[dB]
50 %minimalno slabljenje u nepropusnom opsegu
51 Rs1=40;%[dB]
52
53 %pravljenje digitalnog filtra i izracunavanje reda n1 i opsega Wn1
54 [n1, Wn1]=buttord(Wp1,Ws1,Rp1,Rs1);
55 %izracunavanje parametara filtra a i b koji se prosledjuju funkciji filter
56 [b1, a1] = butter(n1,Wn1,'stop');
57
58 %projektovanje filtra manjeg reda
59 [n11, Wn11]=buttord(Wp1,Ws11,Rp1,Rs1);
60 [b11,a11]=butter(n11,Wn11,'stop');
61 %odziv filtra manjeg reda
62 [h11, f11]=freqz(b11,a11,N/2+1,fs);
63
64 %odziv filtra
65 [h1, f]=freqz(b1,a1,N/2+1,fs);
66
67
68 %amplitudska frekvencijska karakteristika filtra
69 figure(3)
70 hold on;
71 plot(f,20*log10(abs(h1)), 'r', 'LineWidth',1.5);
72 plot(f11,20*log10(abs(h11)), 'b', 'LineWidth',1.5);
73 xlabel('f[Hz]');grid on;
74 title('Amplitudske karakteristike Butterworth filtara za razlicit red filtra');
75 legend('vece n','manje n');
76 saveas(figure(3),'ZAD.4-aks-razlicito.n.png');
77
78 figure(10)
79 plot(f,20*log10(abs(h1)), 'r', 'LineWidth',1.5);
80 xlabel('f[Hz]');grid on;
81 title('Amplitudska karakteristika Butterworth filtara');
82 saveas(figure(10),'ZAD.4-aks-vece.n.png');
83
84 %filtriranje x primenom Butterwoth filtra
85 y1=filter(b1,a1,x);

```

```

86
87 %usrednjavanje spektra filtriranog signala
88 ssr1=zeros(1,length(Sp1))';
89 for i=0:49
90     [spkr, fsp]=spectar_of_20_ms(y1(160*i+1:160*(i+1)), fs);
91     ssr1=ssr1+spkr;
92 end
93
94 ssr1=ssr1./50;
95
96 figure(4)
97 plot(fsp, ssr, 'Linewidth', 1.5); hold on;
98 plot(fsp, ssr1, 'Linewidth', 1.5);
99 xlabel('f[Hz]'); ylabel('|X(jf)|, |Yz(jf)|'); grid on;
100 title('Amplitudske karakteristike ulaznog i filtriranog signala');
101 legend('ulazi signal', 'filtriran signal');
102 saveas(figure(4), 'ZAD.4.aks.nakon.butt.filtriranja.png');
103
104
105 %Pravljenje eliptic filtra
106
107 [n2, Wn2]=ellipord(Wp1, Ws1, Rp1, Rs1);
108 [b2, a2]=ellip(n2, Rp1, Rs1, Wp1, 'stop');
109 [h2, f2]=freqz(b2, a2, N/2+1, fs);
110
111
112 %amplitudska frekvencijska karakteristika filtra
113 figure(5)
114 plot(f2, 20*log10(abs(h2)), 'k-', 'Linewidth', 1.5);
115 xlabel('f[Hz]'); grid on;
116 title('Amplitudska karakteristika eliptic filtra');
117 saveas(figure(5), 'ZAD.4.aks.eliptic.filtriranja.png');
118 %filtriranje x
119 y2=filter(b2, a2, x);
120
121 %usrednjavanje spektra filtriranog signala
122 ssr2=zeros(1,length(Sp1))';
123 for i=0:49
124     [spkr, fsp]=spectar_of_20_ms(y1(160*i+1:160*(i+1)), fs);
125     ssr2=ssr2+spkr;
126 end
127 ssr2=ssr2./50;
128 figure(6)
129 plot(fsp, ssr, 'Linewidth', 1.5); hold on;
130 plot(fsp, ssr2, 'Linewidth', 1.5);
131 xlabel('f[Hz]'); ylabel('|X(jf)|, |Yz(jf)|'); grid on;
132 title('Amplitudske karakteristike ulaznog i filtriranog signala eliptickim filtrom');
133 legend('ulazi signal', 'filtriran signal');
134 saveas(figure(6), 'ZAD.4.aks.original.i.eliptic.png');
135
136 figure(7)
137 plot(fsp, ssr, 'Linewidth', 1.5); hold all;
138 plot(fsp, ssr1, 'Linewidth', 1.5);
139 plot(fsp, ssr2, 'Linewidth', 1.5);
140 xlabel('f[Hz]'); ylabel('|X(jf)|, |Yz(jf)|'); grid on;
141 title('Amplitudske karakteristike originala i filtriranih ');
142 legend('original', 'Butterwoth', 'Eliptic');
143
144 figure(8)
145 plot(t, x); hold on;
146 plot(t, y1);
147 xlabel('t[s]'); ylabel('y(t)'); grid on;
148 title('Izlazni signal nakon primene Butterwoth filtra');
149 legend('original', 'filtriran');
150 saveas(figure(8), 'ZAD.4.izlaz.nakon.filtriranja.png');
151
152 figure(9)
153 plot(t, x); hold on;

```

```

154 plot(t,y1);
155 xlabel('t[s]');ylabel('y(t)');grid on;
156 title('Izlazni signal nakon primene Eliptic filtra');
157 legend('original','filtriran');
158 saveas(figure(9),'ZAD.4_izlaz.nakon.filtra.elptic.png');
159
160 %funkcija koja je trazena da vraca spektar signala od 20,ms;
161 function [X1,fsp] = spectar_of_20_ms(x,fs)
162     N=2^nextpow2(length(x));
163     X=fft(x,N)/length(x);
164     X1=abs(X(1:N/2+1));
165     X1(2:N/2+1)=2*X1(2:N/2+1);
166     fsp=0:fs/N:fs/2
167 end

```

- Bonus - kôd i rezultati Rezultati su :

Za zadati signal iz zadatka 4 na osnovu funkcije pitch koja je ugradjena funckija matlaba dobijen je niz frekvencija. Usrednjavanjem ovog niza dabijena je frekvencija od

$$\text{trazeni} = 213.1044$$

Pomocu koda je dobijena vrednost od

$$\text{dobijen} = 218.75$$

Kod koji je koriscen za izracunavanje ove frekvencije

```

1      clear all;close all;clc;
2
3      %definisanje imena falja koji se ucitava,prethodno snimljen u snimanje.m
4      filename='snimak1.wav';
5      [x,fs]=audioread(filename);
6      %definisanje vremenske ose da bi se nacrtao snimak
7      t=0:1/fs:(length(x)-1)/fs;
8
9      %ocektivana vrednost
10     a=pitch(x,fs);
11     pitch_wanted=sum(a)/length(a);
12     %iznosi 213.1044
13
14
15
16     %furijeova transformacija
17     N=2^nextpow2(length(x));
18     X=fft(x,N)/length(x);
19     X1=abs(X(1:N/2+1));
20     X1(2:N/2+1)=2*X1(2:N/2+1);
21     f1=0:fs/N:fs/2;
22
23     figure(1)
24     plot(f1,X1);
25     xlabel('f[Hz]'),ylabel('|X(jf)|'),title('Amplitudski spektar');
26     x120=x(161:160*2+1);
27     [Sp1,fp1]=spectar_of_20_ms(x120,fs);
28     ssr=zeros(1,length(Sp1));
29     %usrednjavanje spektra originalnog signala
30     for i=0:49
31         [spkr,fsp]=spectar_of_20_ms(x(160*i+1:160*(i+1)),fs);
32         ssr=ssr+spkr;
33     end
34     ssr=ssr./50;
35     %proba na jednom delu signala
36     pitch=pitch_of_20_ms(Sp1,fp1);
37
38     pitch_sum=0;

```



```
39 num=50;
40 for i=0:49
41     [spkr, fsp]=spectar_of_20_ms(x(160*i+1:160*(i+1)), fs);
42     pitch_to_sum=pitch_of_20_ms(spkr, fsp);
43     if(pitch_to_sum>300)
44         num=num-1;
45     else
46         pitch_sum=pitch_sum+pitch_to_sum;
47     end
48 end
49
50
51 %konacna dobijena frekvencija dobijena usrednjavanjem signala je 218.75
52 konacno=pitch_sum/num;
53 %funkcija koja je trazena da vraca spektar signala od 20,ms;
54 function [X1, fsp] = spectar_of_20_ms(x, fs)
55     N=2^nextpow2(length(x));
56     X=fft(x, N)/length(x);
57     X1=abs(X(1:N/2+1));
58     X1(2:N/2+1)=2*X1(2:N/2+1);
59     fsp=0:fs/N:fs/2;
60 end
61 function pitch_freq = pitch_of_20_ms(X1, f1)
62     max_to_find=max(X1);
63     for i=1 :length(X1)
64         if(X1(i)==max_to_find)
65             index=i;
66             pitch_freq=f1(index);
67             break;
68         end
69     end
70
71 end
```