

Univerzitet u Beogradu
Elektrotehnički fakultet

Viktor Ćurčić, 2022/0056

Domaći zadatak

iz predmeta

Digitalna obrada signala (13E053DOS)

Beograd, Jul 2025.

Tabela parametara	
Parametar	Vrednost
P	4
Q	3
R	2
S	2

Contents

1	Zadatak br. 1	3
1.1	Uvod	3
1.2	Analiza	3
2	Zadatak br. 2	4
2.1	Uvod	4
2.2	Analiza	4
3	Zadatak br. 3	5
3.1	Uvod	5
3.2	Analiza	6
4	Zadatak br.4	10
4.1	Uvod	10
4.2	Analiza	10
5	MATLAB kod	12

1 Zadatak br. 1

1.1 Uvod

Napisati program koji izračunava cikličnu konvoluciju signala x i y u $(P + 2)$ tačaka. Potrebno je jedan ispod drugog prikazati grafike signala x i y , a na zasebnom grafiku rezultat konvolucije. Zatim, treba izračunati linearnu konvoluciju ova dva signala, kao i cikličnu konvoluciju u takvom broju tačaka da ona bude jednaka linearnoj. Prikazati ih na grafiku jednu ispod druge i pokazati da su jednake.

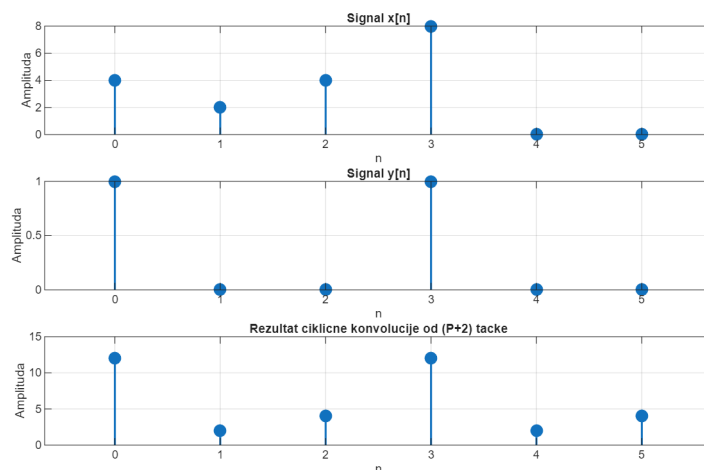
Signali x i y dati su u sledećem obliku:

$$x[n] = \begin{cases} \sin n + 2 \cos 2n + \frac{P}{2} & 0 \leq n < \text{floor}(\frac{P}{2}) \\ 2^n & \text{floor}(\frac{P}{2}) \leq n < P \end{cases}$$

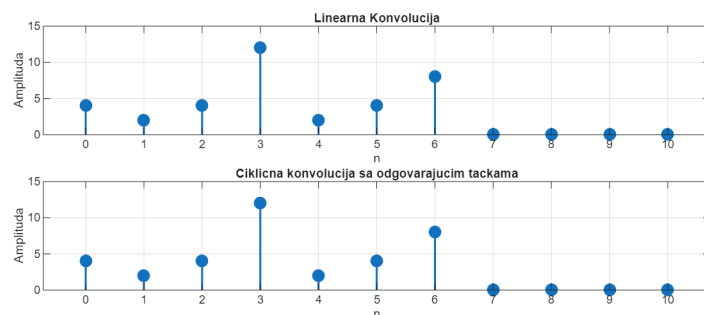
$$y[n] = \begin{cases} (-1)^n + n \bmod 2 & 0 \leq n < \text{floor}(\frac{P}{2}) \\ n - \frac{P}{2} & \text{floor}(\frac{P}{2}) \leq n < P \end{cases}$$

1.2 Analiza

Za zadatu vrednost parametra $P = 4$, dobijeni su sledeći rezultati:



Slika 1.1



Slika 1.2

Proverom korišćenjem ugradjenih funkcija *conv* i *cconv* u MATLAB-u (videti u priloženom kodu), jasno se vidi da su rezultati dobri. **Slika 1.1** prikazuje grafike funkcija $x[n]$ i $y[n]$, i ciklične konvolucije ta dva signala $x[n] * y[n]$ u 6 tačaka. S druge strane, **slika 1.2** prikazuje linearnu i njoj odgovarajuću cikličnu konvoluciju u $2N - 1$ odbiraka.

2 Zadatak br. 2

2.1 Uvod

Za svaki parametar data su 4 signala. Svaki signal je jedna nota odsvirana na jednom instrumentu. Potrebno je iscrtati svaki signal u vremenskom domenu. Zatim, odrediti i iscrtati amplitudsku i faznu frekvencijsku karakteristiku svakog signala (na jednom prozoru sve fazne karakteristike jedne ispod drugih, na drugom sve amplitudske na isti način). Prvi pik u amplitudskoj karakteristici nosi informaciju o noti koja je odsvirana. Treba odrediti za svaki signal na kojoj frekvenciji je taj pik i kojoj noti odgovara (pročitati iz priložene tabele). Za signal po želji pronaći srednji razmak između susjednih pikova.

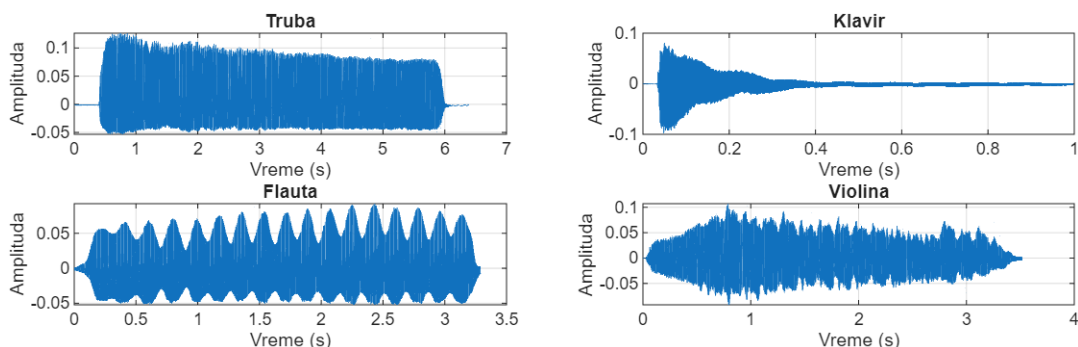
2.2 Analiza

Za vrednost parametra $Q = 3$ analiziraćemo zvukove truba_4.wav, klavir_4.wav, flauta_4.wav i violina_4.wav, respektivno.

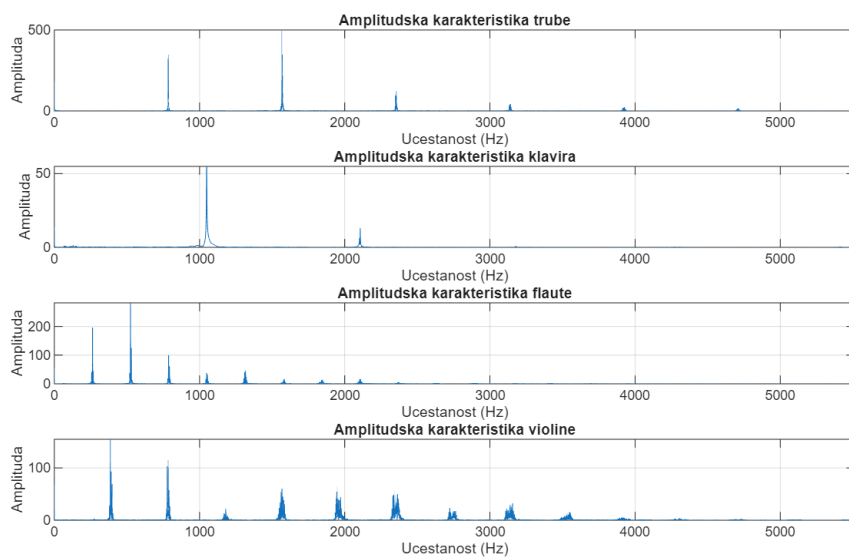
U tabeli ispod date su frekvencije nota (Hz) C i G u različitim oktavama (prva kolona su note, a prva vrsta redni broj oktave). Primer: frekvencija od $65Hz$ odgovara noti C2.

	1	2	3	4	5	6
C	32	65	130	261	523	1047
G	49	98	196	392	784	1661

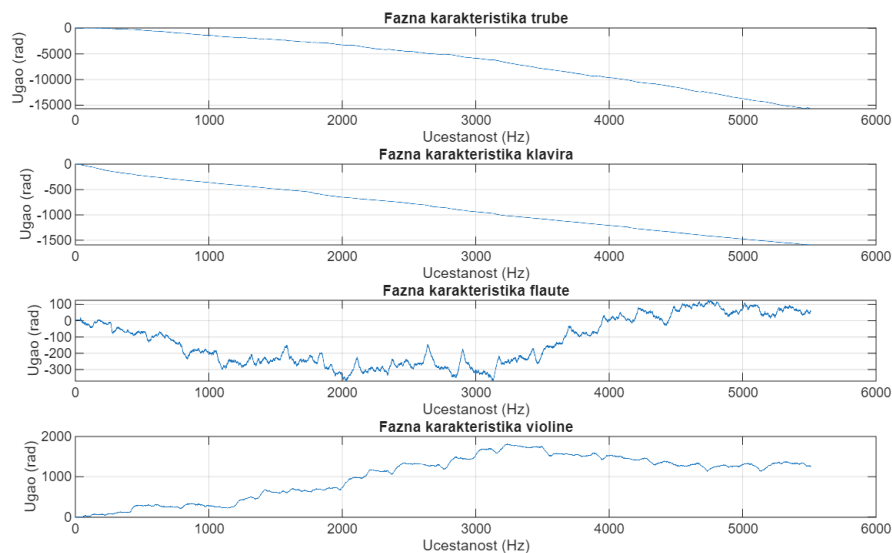
Tabela 2.1



Slika 2.1



Slika 2.2



Slika 2.3

Na **slici 2.1** možemo videti vremenske grafike datih instrumenata, dok se na **slici 2.2** i **slici 2.3** mogu videti amplitudske i fazne karakteristike. **Tabela 2.2** pokazuje vrednosti nota datih instrumenata.

Truba	Klavir	Flauta	Violina
G5	C6	C5	G4

Tabela 2.2

Srednji razmak izmedju susednih pikova iznosi 781.7142 Hz .

3 Zadatak br. 3

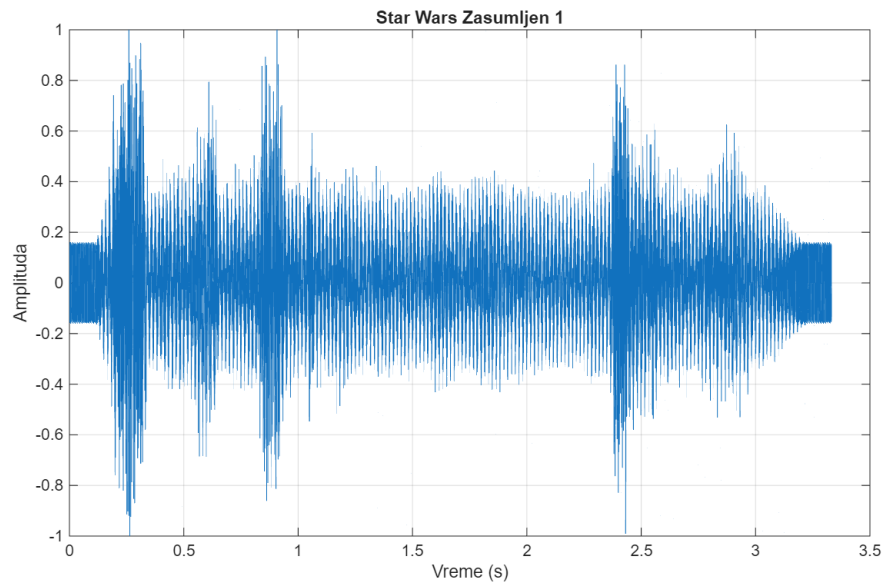
3.1 Uvod

Za svaki parametar dat je jedan audio signal koji je jedna replika iz poznatog filma zašumljena na dva načina. Za prvu zašumljenu verziju treba prikazati vremenski oblik signala, kao i amplitudsku frekvencijsku karakteristiku signala i uočiti na kojim učestanostima u spektru se nalaze komponente od kojih potiče sinusoidalni šum. Za svaku komponentu u spektru koju treba ukloniti projektovati digitalni IIR eliptični filter. Izabrati pogodan tip filtra (lowpass, highpass, bandpass ili bandstop) tako da jedan filter uklanja samo jednu sinusoidalnu komponentu. Signal nakon filtriranja sačuvati koristeći naredbu `audiowrite('isfiltriran1.wav', x, Fs)`. Potom, treba preslušati isfiltrirani signal i potvrditi da je uspešno uklonjen šum. Na kraju, treba iscrtati amplitudske frekvencijske karakteristike filtera, kao i amplitudsku frekvencijsku karakteristiku signala pre i posle filtriranja (na jednom grafiku preklapljenе karakteristike pre i posle filtriranja prvim filterom, a na drugom preklapljenе karakteristike pre i posle filtriranja drugim filterom). Iscrtati vremenski oblik filtriranog signala (na istom grafiku kao i vremenski oblik pre filtriranja).

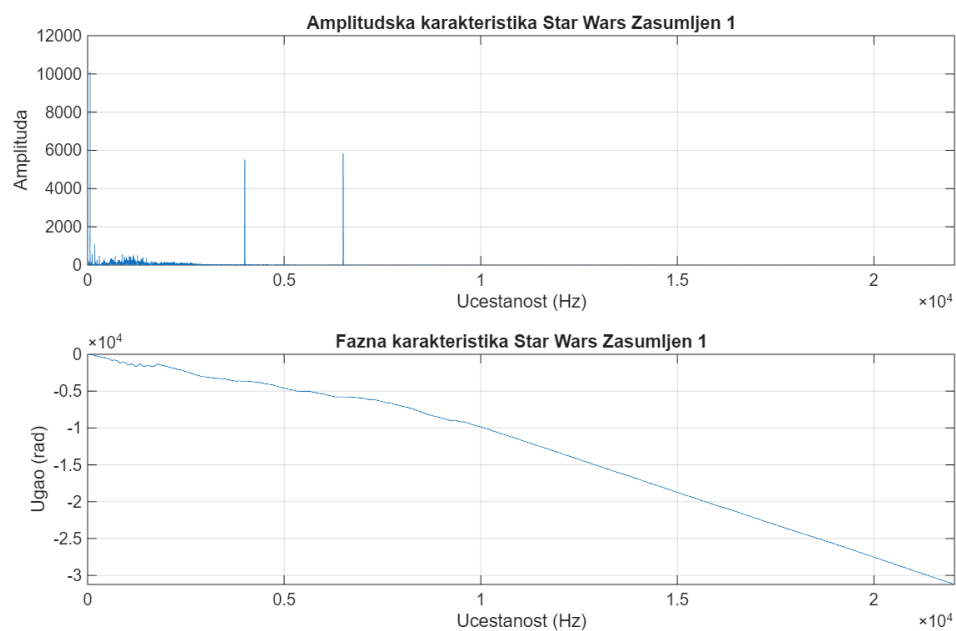
Druga verzija zašumljenog signala dobijena je dodavanjem belog Gausovog šuma na originalni signal. Potrebno je projektovati analogni IIR eliptični filter odgovarajućeg tipa (lowpass, highpass, bandpass ili bandstop), diskretizovati ga bilinearnom metodom i isfiltrirati signal tako da se što manje ugrozi koristan deo signala, a da se ukloni što više šuma. Signal nakon filtriranja sačuvati koristeći naredbu `audiowrite('isfiltriran2.wav', x,`

Fs). Preslušati filtrirani signal i proceniti da li je pobošljan kvalitet. Na kraju, treba iscertati amplitudsku karakteristiku filtra (pre i posle diskretizacije na jednom grafiku), amplitudske karakteristike signala pre i posle filtriranja (na istom grafiku), kao i vremenske oblike signala pre i posle filtriranja (na istom grafiku).

3.2 Analiza

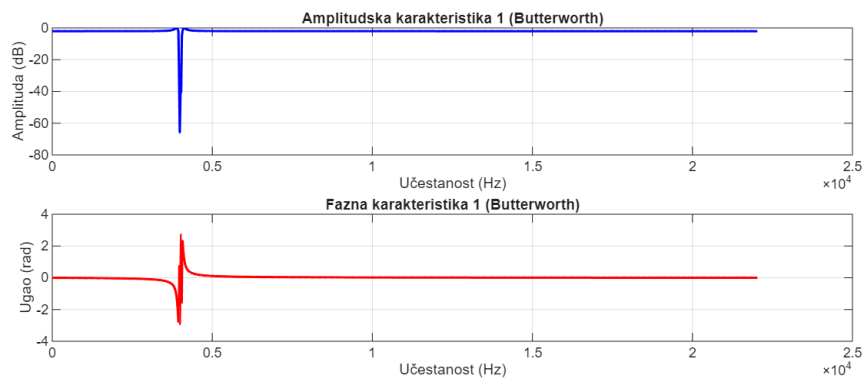


Slika 3.1

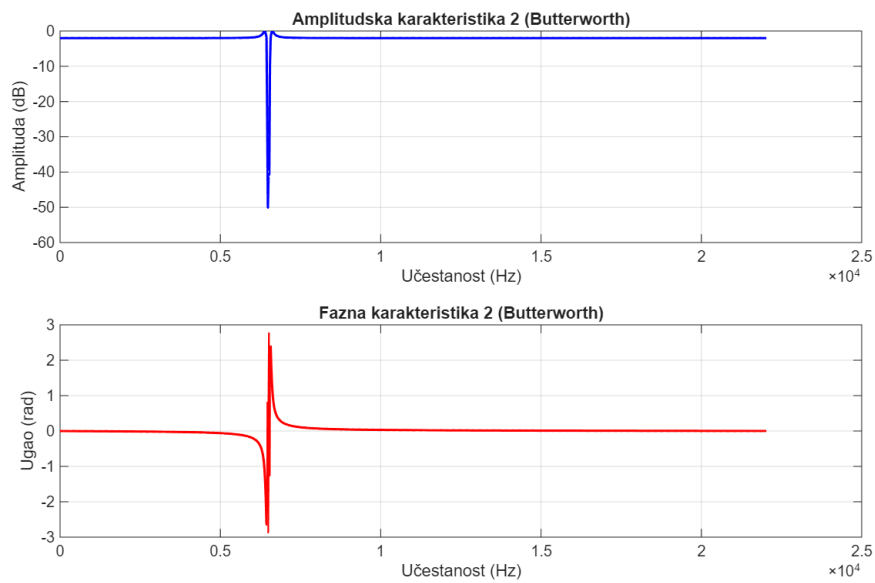


Slika 3.2

Na **slici 3.1** je prikazan vremenski prikaz datog zašumljenog signala, dok su na **slici 3.2** prikazani amplitudski i fazni spektar signala. Na učestanostima 4000 Hz i 6000 Hz se jasno uočavaju sinusoidalne komponente, dok je na učestanosti 0 Hz DC komponenta.

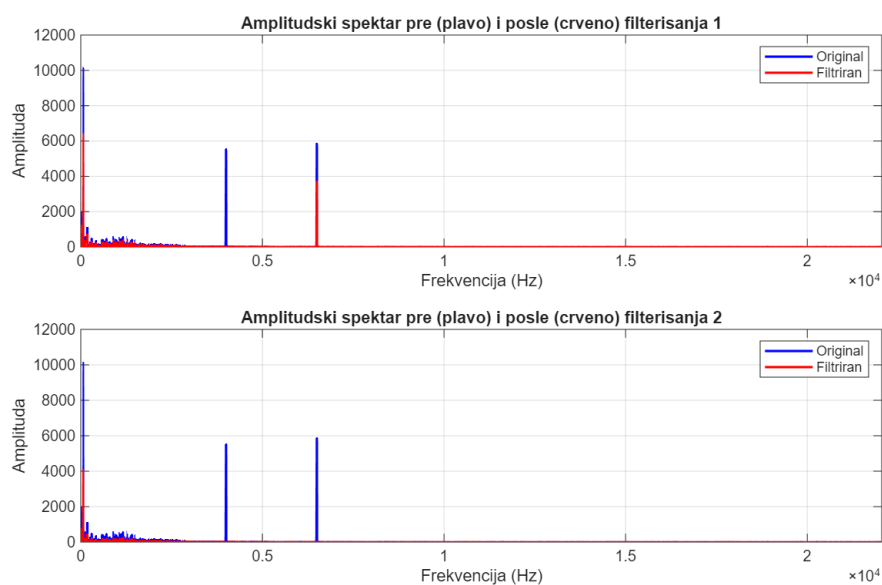


Slika 3.3

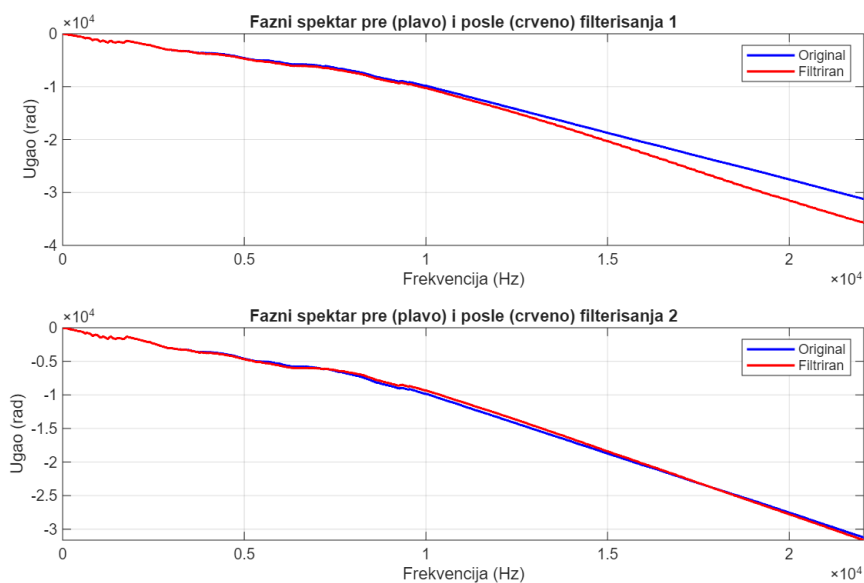


Slika 3.4

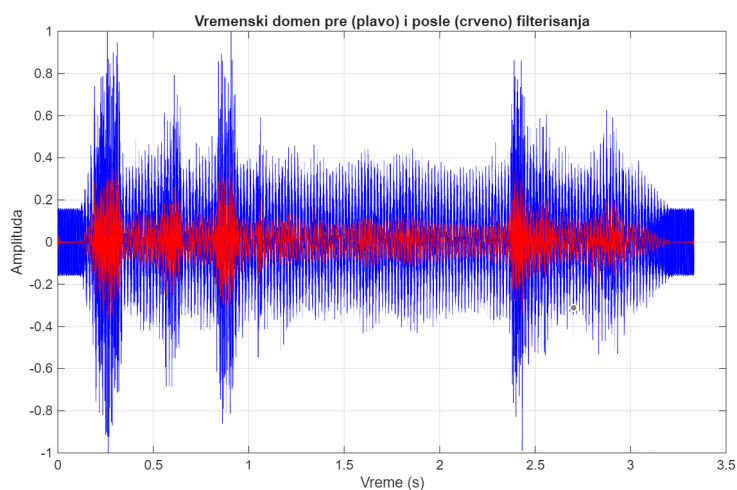
Grafici na **slici 3.3** i **slici 3.4** prikazuju amplitudske i fazne karakteristike Butterworth, odn. eliptičnog filtra za frekvencije na kojima su uočene sinusoidalne komponente.



Slika 3.5

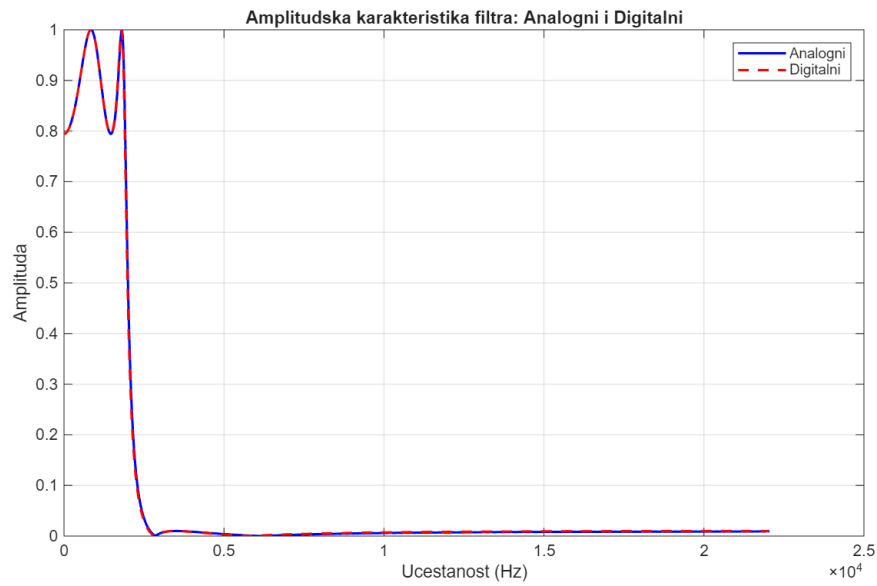


Slika 3.6

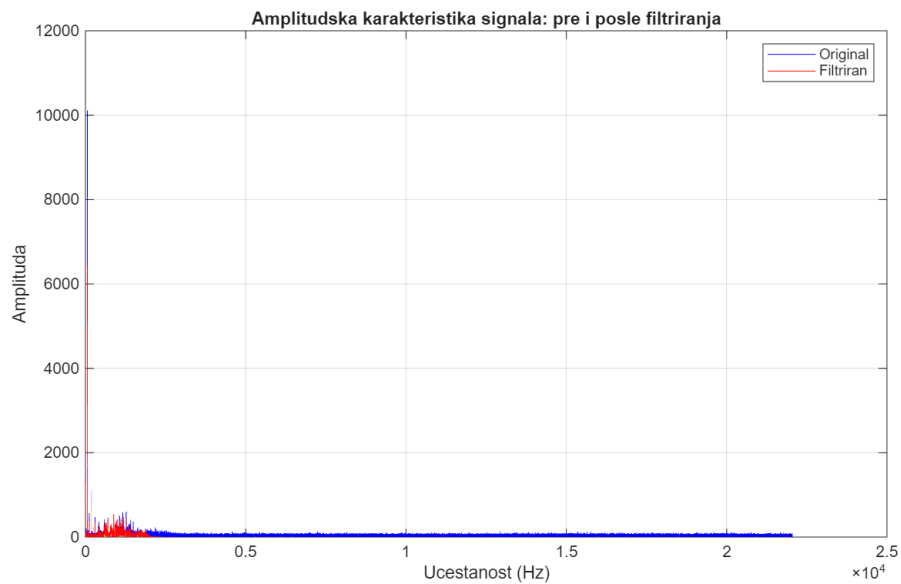


Slika 3.7

Slika 3.5 i **slika 3.6** prikazuju efekat filtriranja na amplitudsku i faznu karakteristiku signala. Vidi se da su nakon filtriranja obe sinusoidalne komponente isfiltrirane. Takodje, zbog uslova zadatka da za sve filtre treba uzeti da je maksimalno slabljenje u propusnom opsegu 2 dB , a minimalno slabljenje u nepropusnom opsegu 40 dB , dolazi do određenog odsecanje DC komponente. To se takodje može videti i na **slici 3.7**.

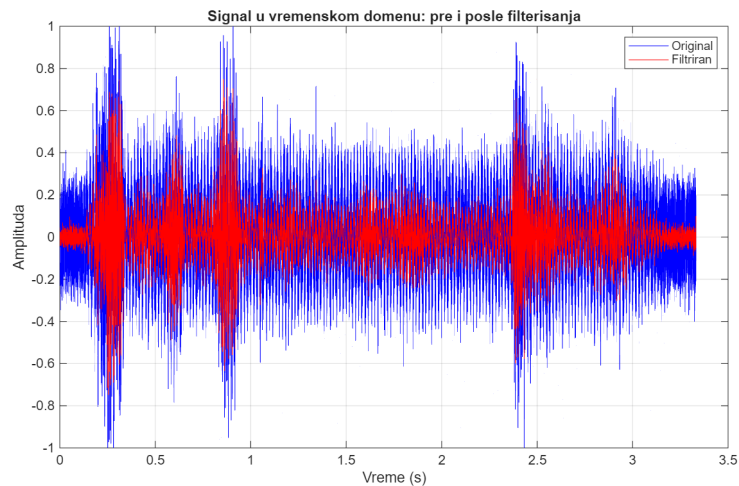


Slika 3.8



Slika 3.9

Na **slici 3.8** i **slici 3.9** vidimo efekat filtriranja IIR eliptičnim filtrom i diskretizacije bilinearnom metodom na drugi zašumljen signal. Zbog uslova da maksimalno slabljenje u propusnom opsegu 2 dB , a minimalno slabljenje u nepropusnom opsegu 40 dB , videćemo da će doći do odsecanja DC komponente, odnosno jačina zvuka će biti slabija. S druge strane, koristan deo signala će većinski biti očuvan.



Slika 3.10

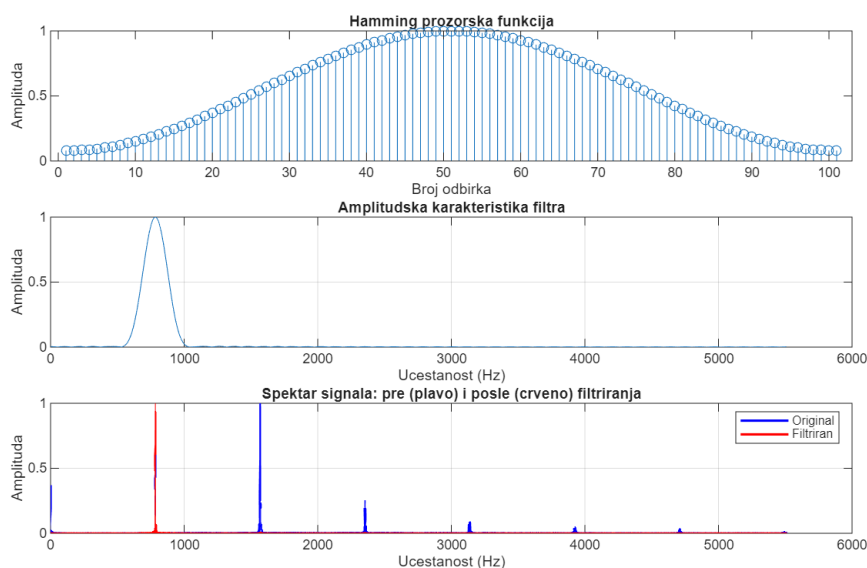
4 Zadatak br.4

4.1 Uvod

Signal koji se koristi u ovom zadatku je prvi signal iz drugog zadatka (truba_4.wav). Projektovati FIR filter odgovarajućeg reda sa Hamming prozorskom funkcijom, tako da propusti samo prvi pik u amplitudskoj karakteristici signala. Potrebno je prikazati odbirke prozorske funkcije i amplitudsku karakteristiku filtra. Prikazati na jednom grafiku amplitudsku karakteristiku signala pre i posle filtriranja.

Nakon toga, odabrati neku neadekvatnu periodu odabiranja signala takvu da dodje do zamene učestanosti (da se “pomeri” preostali pik signala). Iscertati amplitudsku frekvencijsku karakteristiku novodobijenog signala i slušanjem potvrditi da je zamenjena učestanost.

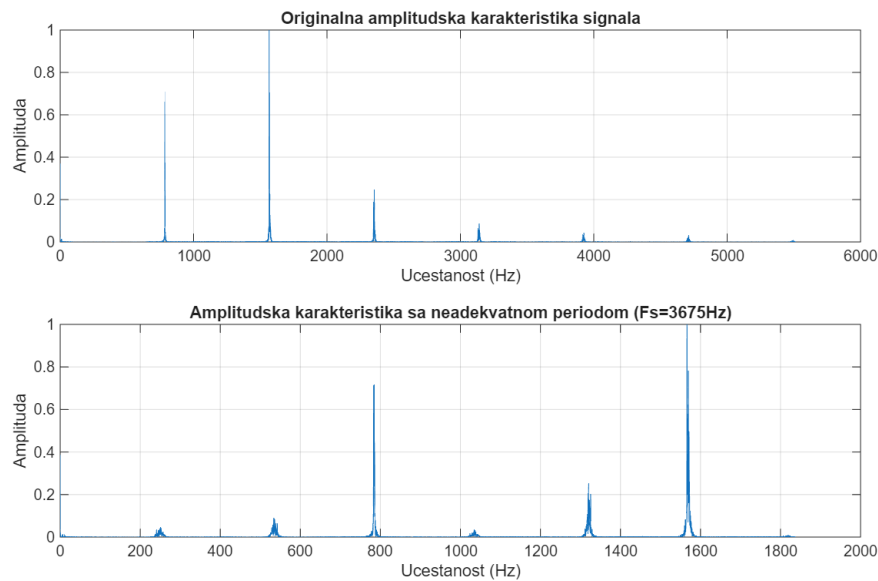
4.2 Analiza



Slika 4.1

Na **slici 4.1** možemo videti prikaz Hamming prozorske funkcije, kao i amplitudsku karakteristiku projektovanog FIR filtra. Zbog korišćenja ugrađene funkcije `fir1()`, filter

će sigurno biti stabilan. Na spektru signala, možemo videti da je uslov zadatka ispunjen - propušten je samo prvi pik.



Slika 4.2

Na **slici 4.2** je prikazan originalna, kao i undersample-ovana amplitudska karakteristika, sa faktorom nedovoljnog smplovanja od 3. Kako bi nedovoljno smplovanje bilo ispunjeno, potrebno je da u originalnoj amplitudskoj karakteristici postoje odbirci na učestanostima $\geq \frac{F_{snovi}}{2} = 1837.5Hz$, što možemo videti da je ispunjeno. Takodje, možemo nedovoljno smplovanje čuti i iz zvuka.

5 MATLAB kod

```
p = mod(56, 4) + 4
Q = mod(11, 4)
R = mod(2022+56, 3)
S = mod(56, 3)

n = 0:5;

x = zeros(1,6);
for k = 1:length(n)
    if n(k) < 2
        x(k) = sin(n(k)) + 2*cos(2*n(k)) + 2;
    elseif n(k) < 4
        x(k) = 2^n(k);
    end
end

y = zeros(1,6);
for k = 1:length(n)
    if n(k) < 2
        y(k) = (-1)^n(k) + mod(n(k),2);
    elseif n(k) < 4
        y(k) = n(k) - 2;
    end
end
```

ZADATAK 1, a)

```
conv_result = zeros(1,6);
for m = 0:5
    sum_val = 0;
    for k = 0:5
        index = mod(m - k, 6);
        sum_val = sum_val + x(k+1) * y(index+1);
    end
    conv_result(m+1) = sum_val;
end

figure;
subplot(3,1,1);
stem(n, x, 'filled', 'LineWidth', 1.5, 'MarkerSize', 8);
title('Signal x[n]');
xlabel('n');
ylabel('Amplituda');
grid on;

subplot(3,1,2);
stem(n, y, 'filled', 'LineWidth', 1.5, 'MarkerSize', 8);
title('Signal y[n]');
xlabel('n');
ylabel('Amplituda');
grid on;

subplot(3,1,3);
stem(n, conv_result, 'filled', 'LineWidth', 1.5, 'MarkerSize', 8);
title('Rezultat ciklicne konvolucije od (P=2) tacke');
xlabel('n');
ylabel('Amplituda');
grid on;
```

ZADATAK 2

```
105 [audio_data1, Fs1] = audioread('truba_4.wav');
106 [audio_data2, Fs2] = audioread('klavir_4.wav');
107 [audio_data3, Fs3] = audioread('flauta_4.wav');
108 [audio_data4, Fs4] = audioread('violina_4.wav');
109
110 duration1 = length(audio_data1)/Fs1;
111 time1 = linspace(0, duration1, length(audio_data1));
112
113 duration2 = length(audio_data2)/Fs2;
114 time2 = linspace(0, duration2, length(audio_data2));
115
116 duration3 = length(audio_data3)/Fs3;
117 time3 = linspace(0, duration3, length(audio_data3));
118
119 duration4 = length(audio_data4)/Fs4;
120 time4 = linspace(0, duration4, length(audio_data4));
121
122
```

ZADATAK 2 a)

```
123 figure;
124
125 subplot(4,2,1);
126 plot(time1, audio_data1);
127 title('Truba');
128 xlabel('Vreme (s)');
129 ylabel('Amplituda');
130 grid on;
131
132 subplot(4,2,2);
133 plot(time2, audio_data2);
134 title('Klavir');
135 xlabel('Vreme (s)');
136 ylabel('Amplituda');
137 grid on;
138
139 subplot(4,2,3);
140 plot(time3, audio_data3);
141 title('Flauta');
142 xlabel('Vreme (s)');
143 ylabel('Amplituda');
144 grid on;
145
146 subplot(4,2,4);
147 plot(time4, audio_data4);
148 title('Violina');
149 xlabel('Vreme (s)');
150 ylabel('Amplituda');
151 grid on;
152
153
```

ZADATAK 1, b)

```
61 N = length(x);
62 M = length(y);
63
64 linear_conv = zeros(1, N+M-1);
65
66 for m = 0:N+M-2
67     sum_val = 0;
68     for k = max(0, m-(M-1)):min(m, N-1)
69         sum_val = sum_val + x(k+1) * y(m-k+1);
70     end
71     linear_conv(m+1) = sum_val;
72 end
73
74 x_sredjeno = [x zeros(1, M-1)];
75 y_sredjeno = [y zeros(1, N-1)];
76
77 ciklicna_conv = zeros(1, N+M-1);
78 for m = 0:N+M-2
79     sum_val = 0;
80     for k = 0:N+M-2
81         index = mod(m - k, N + M - 1);
82         sum_val = sum_val + x_sredjeno(k+1) * y_sredjeno(index+1);
83     end
84     ciklicna_conv(m+1) = sum_val;
85 end
86
87 figure;
88
89 subplot(3,1,1);
90 stem(0:length(linear_conv)-1, linear_conv, 'filled', 'LineWidth', 1.5, 'MarkerSize', 8);
91 title('Linearna konvolucija');
92 xlabel('n');
93 ylabel('Amplituda');
94 grid on;
95
96 subplot(3,1,2);
97 stem(0:length(ciklicna_conv)-1, ciklicna_conv, 'filled', 'LineWidth', 1.5, 'MarkerSize', 8);
98 title('Ciklicna konvolucija sa odgovarajucim tackama');
99 xlabel('n');
100 ylabel('Amplituda');
101 grid on;
102
103 provera1 = conv(x, y)
104 provera2 = cconv(x, y)
```

ZADATAK 2 b)

```
154 N1 = length(audio_data1);
155 N2 = length(audio_data2);
156 N3 = length(audio_data3);
157 N4 = length(audio_data4);
158
159 fft_res1 = fft(audio_data1);
160 fft_res2 = fft(audio_data2);
161 fft_res3 = fft(audio_data3);
162 fft_res4 = fft(audio_data4);
163
164 freq1 = (0 : N1/2-1) * (Fs1 / N1);
165 freq2 = (0 : N2/2-1) * (Fs2 / N2);
166 freq3 = (0 : N3/2-1) * (Fs3 / N3);
167 freq4 = (0 : N4/2-1) * (Fs4 / N4);
168
169 figure;
170
171 subplot(4,1,1);
172 plot(freq1, abs(fft_res1(1:N1/2)));
173 xlim([0 Fs1/2]);
174 title('Amplitudska karakteristika trube');
175 xlabel('Ucestanost (Hz)');
176 ylabel('Amplituda');
177 grid on;
178
179 subplot(4,1,2);
180 plot(freq2, abs(fft_res2(1:N2/2)));
181 xlim([0 Fs2/2]);
182 title('Amplitudska karakteristika klavira');
183 xlabel('Ucestanost (Hz)');
184 ylabel('Amplituda');
185 grid on;
186
187 subplot(4,1,3);
188 plot(freq3, abs(fft_res3(1:N3/2)));
189 xlim([0 Fs3/2]);
190 title('Amplitudska karakteristika flaute');
191 xlabel('Ucestanost (Hz)');
192 ylabel('Amplituda');
193 grid on;
194
195 subplot(4,1,4);
196 plot(freq4, abs(fft_res4(1:N4/2)));
197 xlim([0 Fs4/2]);
198 title('Amplitudska karakteristika violine');
199 xlabel('Ucestanost (Hz)');
200 ylabel('Amplituda');
201 grid on; N truba = 784, klavir = 1049, flauta = 262, violina = 385
202
203 figure;
```

206	
207	subplot(4,1,1);
208	plot(freq1, unwrap(angle(fft_res1(1:N1/2))));
209	title('Fazna karakteristika trube');
210	xlabel('Ucestanost (Hz)');
211	ylabel('Ugao (rad)');
212	grid on;
213	
214	subplot(4,1,2);
215	plot(freq2, unwrap(angle(fft_res2(1:N2/2))));
216	title('Fazna karakteristika klavira');
217	xlabel('Ucestanost (Hz)');
218	ylabel('Ugao (rad)');
219	grid on;
220	
221	subplot(4,1,3);
222	plot(freq3, unwrap(angle(fft_res3(1:N3/2))));
223	title('Fazna karakteristika flaute');
224	xlabel('Ucestanost (Hz)');
225	ylabel('Ugao (rad)');
226	grid on;
227	
228	subplot(4,1,4);
229	plot(freq4, unwrap(angle(fft_res4(1:N4/2))));
230	title('Fazna karakteristika violine');
231	xlabel('Ucestanost (Hz)');
232	ylabel('Ugao (rad)');
233	grid on;
234	
ZADATAK 2 c)	
235	disp('Truba: G5, Klavir: C6, Flauta: C4, Violina: G4');
236	
ZADATAK 2 d) TRUBA	
237	[pks, locs] = findpeaks(abs(fft_res1(1:N1/2+1)), 'MinPeakHeight', 0.7 * max(abs(fft_res1(1:N1/2+1))), 'MinPeakDistance', 650);
238	
239	peak_freqs = locs * (Fs1 / N1);
240	
241	peak_diffs = diff(peak_freqs);
242	
243	sr_razdaljina = mean(peak_diffs);
244	disp(['Srednji razmak između susjednih pikova: ', num2str(sr_razdaljina), ' Hz']);
245	
246	
ZADATAK 3 Prvi signal	
247	[audiodata1, Fs1] = audioread('star_wars_zasumljen1.wav');
248	trajanje1 = length(audiodata1)/Fs1;
249	time1 = linspace(0, trajanje1, length(audiodata1));
250	
251	figure;
252	plot(time1, audiodata1);
253	title('Star wars zasumljen 1');
254	xlabel('Vreme (s)');
255	ylabel('Amplituda');
256	grid on;
257	
258	N1 = length(audiodata1);
259	
260	fftres1 = fft(audiodata1);
261	freq1 = (0:N1/2-1)*(Fs1/N1);
262	
263	figure;
264	
265	subplot(2,1,1);
266	plot(freq1,abs(fftres1(1:N1/2)));
267	xlim([0 Fs1/2]);
268	title('Amplitudska karakteristika Star Wars Zasumljen 1');
269	xlabel('Ucestanost (Hz)');
270	ylabel('Amplituda');
271	grid on;
272	
273	subplot(2,1,2);
274	plot(freq1, unwrap(angle(fftres1(1:N1/2))));
275	xlim([0 Fs1/2]);
276	title('Fazna karakteristika Star Wars Zasumljen 1');
277	xlabel('Ucestanost (Hz)');
278	ylabel('Ugao (rad)');
279	grid on;
280	
281	Rp = 2;
282	Rs = 40;
283	
284	f_noise1 = 4000;
285	f_noise2 = 6500;
286	
287	[b1, a1] = ellip(4, Rp, Rs, [f_noise1-50 f_noise1+50]/(Fs1/2), 'stop');
288	[b2, a2] = ellip(4, Rp, Rs, [f_noise2-50 f_noise2+50]/(Fs1/2), 'stop');
289	
290	filtered_signal1 = filtfilt(b1, a1, audiodata1);
291	filtered_signal2 = filtfilt(b2, a2, filtered_signal1);
292	
293	audiowrite('isfiltrirani1.wav', filtered_signal2, Fs1);
294	
295	sound(filtered_signal2, Fs1);
296	
297	figure;
298	[h1, w1] = freqz(b1, a1, 1024, Fs1);
299	
300	subplot(2,1,1);
301	plot(w1, 20*log10(abs(h1)), 'b', 'LineWidth', 1.5);
302	title('Amplitudska karakteristika 1 (Butterworth)');
303	xlabel('Ucestanost (Hz)');

300	subplot(2,1,2);	366	[x, fs] = audioread('star_wars_zasumijen2.wav');
301	plot(wd, angle(h1), 'r', 'linewidth', 1.5);	367	clear freqs
302	title('Fazna karakteristika 1 (Butterworth)');	368	
303	xlabel('Ucestanost (Hz)');	369	X_f = abs(fft(x));
304	ylabel('Ugao (rad)');	370	
305	grid on;	371	fc = 1000;
306		372	wc = 2 * Fs * tan(pi * fc / Fs);
307		373	n = 4;
308		374	[bs, as] = ellip(n, Rp, Rs, wc, 's');
309	figure;	375	
310	[h2, w2] = freqz(b2, a2, 1024, Fs);	376	poles_analog = roots(as);
311		377	if any(real(poles_analog) >= 0)
312		378	error('Nestabilan');
313		379	else
314		380	disp('Stabilan');
315		381	end
316		382	
317	subplot(2,1,1);	383	[bd, ad] = bilinear(bs, as, Fs);
318	plot(w2, 20*log10(abs(h2)), 'b', 'linewidth', 1.5);	384	poles_digital = roots(ad);
319	title('Amplitudska karakteristika 2 (Butterworth)');	385	
320	xlabel('Ucestanost (Hz)');	386	if any(abs(poles_digital) >= 1)
321	ylabel('Amplituda (dB)');	387	error('Nestabilan');
322	grid on;	388	disp('Stabilan');
323		389	end
324	subplot(2,1,2);	390	
325	plot(w2, angle(h2), 'r', 'linewidth', 1.5);	391	x_filtered = filtfilt(bd, ad, x);
326	title('Fazna karakteristika 2 (Butterworth)');	392	
327	xlabel('Ucestanost (Hz)');	393	audiowrite('isfiltriran2.wav', x_filtered, Fs);
328	ylabel('Ugao (rad)');	394	
329	grid on;	395	w_analog = linspace(0, 2*pi*Fs/2, 1024);
330		396	H_analog = freqz(bs, as, w_analog);
331	fft_filtered1 = fft(filtered_signal1);	397	[H_digital, w_digital] = freqz(bd, ad, 1024, Fs);
332	fft_filtered2 = fft(filtered_signal2);	398	
333		399	figure;
334	figure;	400	plot(w_analog/(2*pi), abs(H_analog), 'b', 'linewidth', 1.5); hold on;
335	subplot(2,1,1);	401	plot(w_digital, abs(H_digital), 'r--', 'linewidth', 1.5);
336	plot(freq1, abs(fft(fres1(1:Nl/2))), 'b', 'linewidth', 1.5); hold on;	402	legend('Analogni', 'Digitalni');
337	plot(freq1, abs(fft(fres1(1:Nl/2))), 'r', 'linewidth', 1.5);	403	xlabel('Ucestanost (Hz)');
338	xlim(0 Fs1/2); title('Amplitudski spektar pre (plavo) i posle (crveno) filterisanja 1');	404	ylabel('Amplituda');
339	xlabel('Frekvencija (Hz)');	405	title('Amplitudska karakteristika filtra: Analogni i Digitalni');
340		406	grid on;
341	subplot(2,1,2);	407	
342	plot(freq1, abs(fft(fres1(1:Nl/2))), 'b', 'linewidth', 1.5); hold on;	408	X_orig = abs(fft(x));
343	plot(freq1, abs(fft(fres1(1:Nl/2))), 'r', 'linewidth', 1.5);	409	X_filt = abs(fft(x_filtered));
344	xlim(0 Fs1/2); title('Fazni spektar pre (plavo) i posle (crveno) filterisanja 2');	410	f = linspace(0, Fs, length(x));
345	xlabel('Frekvencija (Hz)');	411	
346		412	figure;
347		413	plot(f(1:end/2), X_orig(1:end/2), 'b'); hold on;
348	figure;	414	plot(f(1:end/2), X_filt(1:end/2), 'r');
349	subplot(2,1,1);	415	legend('Original', 'Filtriran');
350	plot(freq1, unwrap(angle(fft(fres1(1:Nl/2)))), 'b', 'linewidth', 1.5); hold on;	416	xlabel('Ucestanost (Hz)');
351	plot(freq1, unwrap(angle(fft(fres1(1:Nl/2)))), 'r', 'linewidth', 1.5);	417	ylabel('Amplituda');
352	xlim(0 Fs1/2); title('Fazni spektar pre (plavo) i posle (crveno) filterisanja 1');	418	title('Amplitudska karakteristika signala: pre i posle filtriranja');
353	xlabel('Frekvencija (Hz)');	419	grid on;
354		420	
355	subplot(2,1,2);	421	t = (0:length(x)-1)/Fs;
356	plot(freq1, unwrap(angle(fft(fres1(1:Nl/2)))), 'b', 'linewidth', 1.5); hold on;	422	
357	plot(freq1, unwrap(angle(fft(fres1(1:Nl/2)))), 'r', 'linewidth', 1.5);	423	figure;
358	xlim(0 Fs1/2); title('Fazni spektar pre (plavo) i posle (crveno) filterisanja 2');	424	odsecanje = f_vrh-50, f_vrh+50;
359	xlabel('Frekvencija (Hz)');	425	red = 100;
360		426	
361	figure;	427	b = fir1(red, [odsecanje(1) odsecanje(2)]/(Fs/2), 'bandpass', hamming(red+1));
362	plot(time1, audiodata1, 'b', time1, filtered_signal2, 'r');	428	
363	title('Vremenski domen pre (plavo) i posle (crveno) filterisanja');	429	x_filterovan = filter(b, 1, x);
364	xlabel('Vreme (s)');	430	
365	ylabel('Amplituda');	431	figure;
	grid on;	432	
		433	subplot(3,1,1);
			stem(hamming(red+1));
			title('Hamming prozorska funkcija');
			xlabel('Broj odbirka');
			ylabel('Amplituda');
			grid on;
			subplot(3,1,2);
			[H,freq] = freqz(b,1,1024,Fs);
			plot(freq, abs(H));
			title('Amplitudska karakteristika filtra');
			xlabel('Ucestanost (Hz)');
			ylabel('Amplituda');
			grid on;
			subplot(3,1,3);
			X_filt = abs(fft(x_filterovan));
			X_filt = X_filt(1:Nl/2+1);
			plot(f, X/max(X), 'b', 'linewidth', 1.5); hold on;
			plot(f, X_filt/max(X_filt), 'r', 'linewidth', 1.5);
			title('Spektar signala: pre (plavo) i posle (crveno) filtriranja');
			xlabel('Ucestanost (Hz)');
			ylabel('Amplituda');
			legend('Original', 'Filtriran');
			grid on;
			sound(x_filterovan, Fs);
			audiowrite('truba_filtered.wav', x_filterovan, Fs);
			neadekv_faktor = 3;
			x_neadekv = x(1:neadekv_faktor:end);
			Fs_novi = Fs/neadekv_faktor;
			N_us = length(x_neadekv);
			f_us = (0:N_us/2)*(Fs_novi/N_us);
			X_us = abs(fft(x_neadekv));
			X_us = X_us(1:N_us/2+1);
			figure;
			subplot(2,1,1);
			plot(f, X/max(X));
			title('Originalna amplitudska karakteristika signala');
			xlabel('Ucestanost (Hz)');
			ylabel('Amplituda');
			grid on;
			subplot(2,1,2);
			plot(f_us, X_us/max(X_us));
			title('Amplitudska karakteristika sa neadekvatnom periodom (Fs=numistr(Fs_novi),Hz)');
			xlabel('Ucestanost (Hz)');
			ylabel('Amplituda');
			grid on;
			sound(x_neadekv, Fs_novi);

```

420
421
422
423
424
425
426
427
428
429
430
431
432
433
t = (0:length(x)-1)/Fs;
figure;
plot(t, x, 'b'); hold on;
plot(t, x_filtered, 'r');
legend('Original', 'Filtriran');
xlabel('Vreme (s)');
ylabel('Amplituda');
title('Signal u vremenskom domenu: pre i posle filterisanja');
grid on;
sound(x_filtered, Fs);

```

ZADATAK 4

```

434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
[x, Fs] = audioread('truba_4.wav');
N = length(x);
f = (0:N/2)*(Fs/N);
X = abs(fft(x));
X = X(1:N/2+1);
f_vrh = 784;
odsecanje = [f_vrh-50, f_vrh+50];
red = 100;
b = fir1(red, [odsecanje(1) odsecanje(2)]/(Fs/2), 'bandpass', hamming(red+1));
x_filterovan = filter(b, 1, x);
figure;
subplot(3,1,1);
stem(hamming(red+1));
title('Hamming prozorska funkcija');
xlabel('Broj odbirka');
ylabel('Amplituda');
grid on;
subplot(3,1,2);
[H,freq] = freqz(b,1,1024,Fs);
plot(freq, abs(H));
title('Amplitudska karakteristika filtra');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
grid on;
subplot(3,1,3);
X_filt = abs(fft(x_filterovan));
X_filt = X_filt(1:Nl/2+1);
plot(f, X/max(X), 'b', 'linewidth', 1.5); hold on;
plot(f, X_filt/max(X_filt), 'r', 'linewidth', 1.5);
title('Spektar signala: pre (plavo) i posle (crveno) filtriranja');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
legend('Original', 'Filtriran');
grid on;

```

ZADATAK 3 Drugi signal

```

366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
[x, fs] = audioread('star_wars_zasumijen2.wav');
clear freqs
X_f = abs(fft(x));
fc = 1000;
wc = 2 * Fs * tan(pi * fc / Fs);
n = 4;
[bs, as] = ellip(n, Rp, Rs, wc, 's');
poles_analog = roots(as);
if any(real(poles_analog) >= 0)
error('Nestabilan');
else
disp('Stabilan');
end
[bd, ad] = bilinear(bs, as, Fs);
poles_digital = roots(ad);
if any(abs(poles_digital) >= 1)
error('Nestabilan');
else
disp('Stabilan');
end
x_filtered = filtfilt(bd, ad, x);
audiowrite('isfiltriran2.wav', x_filtered, Fs);
w_analog = linspace(0, 2*pi*Fs/2, 1024);
H_analog = freqz(bs, as, w_analog);
[H_digital, w_digital] = freqz(bd, ad, 1024, Fs);
figure;
plot(w_analog/(2*pi), abs(H_analog), 'b', 'linewidth', 1.5); hold on;
plot(w_digital, abs(H_digital), 'r--', 'linewidth', 1.5);
legend('Analogni', 'Digitalni');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
title('Amplitudska karakteristika filtra: Analogni i Digitalni');
grid on;
X_orig = abs(fft(x));
X_filt = abs(fft(x_filtered));
f = linspace(0, Fs, length(x));
figure;
plot(f(1:end/2), X_orig(1:end/2), 'b'); hold on;
plot(f(1:end/2), X_filt(1:end/2), 'r');
legend('Original', 'Filtriran');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
title('Amplitudska karakteristika signala: pre i posle filtriranja');
grid on;
t = (0:length(x)-1)/Fs;
figure;
odsecanje = [f_vrh-50, f_vrh+50];
red = 100;
b = fir1(red, [odsecanje(1) odsecanje(2)]/(Fs/2), 'bandpass', hamming(red+1));
x_filterovan = filter(b, 1, x);
figure;
subplot(3,1,1);
stem(hamming(red+1));
title('Hamming prozorska funkcija');
xlabel('Broj odbirka');
ylabel('Amplituda');
grid on;
subplot(3,1,2);
[H,freq] = freqz(b,1,1024,Fs);
plot(freq, abs(H));
title('Amplitudska karakteristika filtra');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
grid on;
subplot(3,1,3);
X_filt = abs(fft(x_filterovan));
X_filt = X_filt(1:Nl/2+1);
plot(f, X/max(X), 'b', 'linewidth', 1.5); hold on;
plot(f, X_filt/max(X_filt), 'r', 'linewidth', 1.5);
title('Spektar signala: pre (plavo) i posle (crveno) filtriranja');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
legend('Original', 'Filtriran');
grid on;
sound(x_filterovan, Fs);
audiowrite('truba_filtered.wav', x_filterovan, Fs);
neadekv_faktor = 3;
x_neadekv = x(1:neadekv_faktor:end);
Fs_novi = Fs/neadekv_faktor;
N_us = length(x_neadekv);
f_us = (0:N_us/2)*(Fs_novi/N_us);
X_us = abs(fft(x_neadekv));
X_us = X_us(1:N_us/2+1);
figure;
subplot(2,1,1);
plot(f, X/max(X));
title('Originalna amplitudska karakteristika signala');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
grid on;
subplot(2,1,2);
plot(f_us, X_us/max(X_us));
title('Amplitudska karakteristika sa neadekvatnom periodom (Fs=numistr(Fs_novi),Hz)');
xlabel('Ucestanost (Hz)');
ylabel('Amplituda');
grid on;
sound(x_neadekv, Fs_novi);

```