

Modelska analiza I

Naloga 110

22. 12. 2022



10. naloga – Spektralna analiza in filtriranje

1. Signaloma s 512 točkami na datotekah `val2.dat` in `val3.dat`¹ določi frekvenčni spekter. Preskusite različne okenske funkcije. Kako se spremeni spekter, če analiziramo krajše intervale (64, 128, ... točk)?

Signal $u(t)$, ki prihaja v merilno napravo s prenosno funkcijo $r(t)$, se ob dodatku šuma $n(t)$ preoblikuje v

$$c(t) = u(t) * r(t) + n(t) = s(t) + n(t).$$

Iz izmerjenega časovnega poteka $c(t)$ bi radi, ob poznavanju odzivne funkcije $r(t)$ in ob nekaterih predpostavkah o šumu $n(t)$, rekonstruirali vpadni signal $u(t)$. N. Wiener je predlagal naslednjo rešitev, ki sledi iz minimizacije napake po metodi najmanjših kvadratov². Pred dekonvolucijo je treba transformiranko $C(f)$ pomnožiti s filtrom,

$$\Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2}.$$

2. S pomočjo Wienerjevega filtra napravi dekonvolucijo signalov na datotekah `signal{0,1,3,4}.dat`. Število točk v posameznem signalu je 512. Na zadnjih treh datotekah je signalu primešan šum. Prenosna funkcija je

$$r(t) = \frac{1}{2\tau} \exp(-|t|/\tau), \quad \tau = 16.$$

V fotografiji imamo pogosto opravka s popačitvami konvolucijske narave: razmazanje zaradi gibanja (*motion blur*), tresenje objektiva med ekspozicijo, slabo fokusiranje, različne optične aberacije. V primeru, da poznamo prenosno funkcijo (t.i. *point spread function*), lahko posežemo po dekonvolucijskih metodah, s katerimi lahko do neke mere rekonstruiramo originalno fotografijo, odvisno seveda od zrnatosti fotografije.

3. Poskusi očistiti podobe Lene (`lena_k□_n□.pgm`)³ v arhivu `lena_slike.tar.gz`, razmazane s tremi znanimi konvolucijskimi jedri: tresoč objektiv (`kernel1.pgm`), slab fokus (`kernel2.pgm`) ter uklonska mrežica (`kernel3.pgm`). Datotekam je primešana različna količina Gaussovega šuma (RMS = 0, 4, 8, 16).

Pri filtriranju uporabi Wienerjev filter. Po potrebi je smiselno tudi direktno maskirati okolice valovnih vektorjev, kjer je prenosna funkcija singularna. Upoštevaj tudi, da spektralne metode sliko razumejo v periodičnem smislu, kar ustvari ostre skoke na robu. Ugotovi, ali lahko z okenskimi funkcijami ali razširitvijo slike s konstantnimi vrednostmi zmanjšaš vpliv artefaktov na robu.

Poskusi tudi očistiti sliko z dodano periodično motnjo (`*_nx.pgm`).

`pgm` format je P2 različica `ppm` formata, ki shranjuje sivine v tekstovni obliki, na začetku pa ima glavo oblike P2 `x y n`, kjer sta `x` in `y` dimenziji slike, `n` = 255 je pa maksimalna vrednost sivine, ki predstavlja belo barvo. Ta format je priporočljiv tudi za izhodne slike, saj ga je enostavno pretvoriti v kateri koli drug format in nazaj s programom `convert` iz paketa `imagemagick`:

```
> convert a.pgm a.png
> convert a.png -compress none a.pgm4
```

S tem se izognemo klicanju knjižnic za kodiranje slikovnih formatov. Pazi, po filtriranju se lahko pojavijo vrednosti, ki ne padejo v območje [0, 255]. Take presežne vrednosti je treba popraviti pred zapisovanjem izhodne slike.

¹<http://predmeti.fmf.uni-lj.si/modelska/podatki/>

²gl. *Numerical Recipes*, pogl. 12.6

³datoteke so na voljo na predmeti.fmf.uni-lj.si/modelska/podatki/

⁴brez `-compress none` argumenta namesto tekstovnega zapiše binarni format

Naloga 1:

- Vplvr m+duvnh obenstn k funkij:
 - Signal ms. sum
 - Sintna vlova
 - Pso dcuji
- Analiza krajovih / redkrajovih intervalov
- $N = 2^n?$

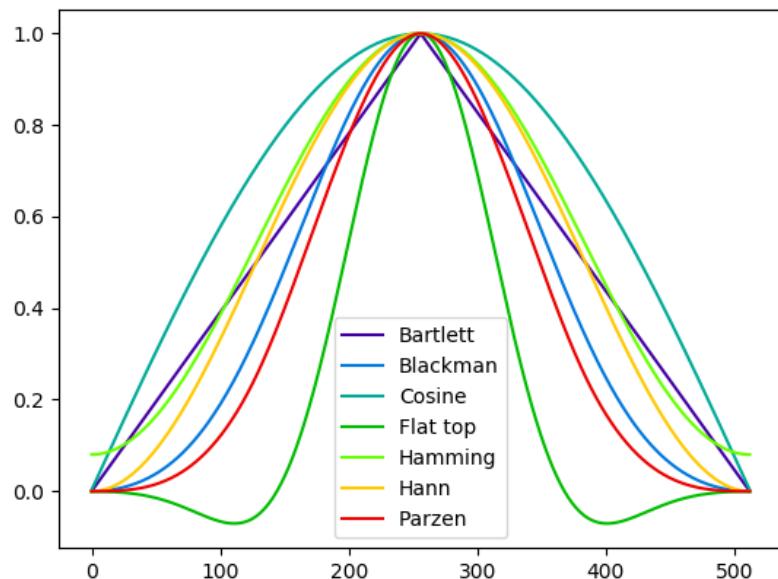
110 - Spektralna analiza in filtriranje

Blaž Levpušček, 28212007

24. 10. 2022

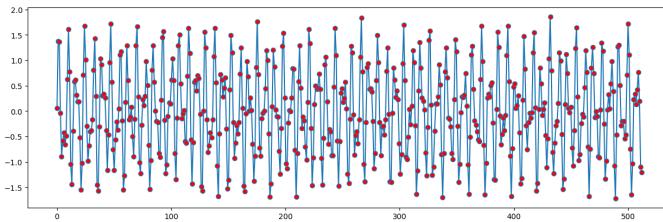
1 Določanje frekvenčnega spektra

Pri tej nalogi sem z diskretno Fourierjevo transformacijo analiziral dva signala. Analizo sem naredil na originalnih signalih ter na signalih, na katerih sem uporabil različne okenske funkcije, ki so prikazane na grafu 1.

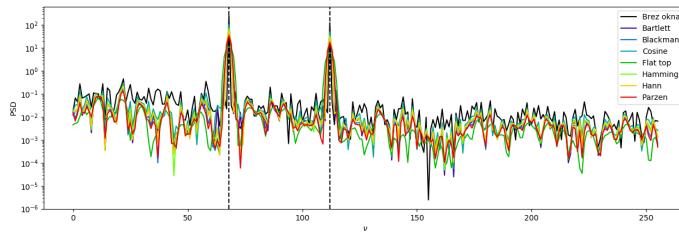


Slika 1: Uporabljene okenske funkcije.

Na sliki 2 je prikazan prvi signal, na sliki 3 pa njegova Fourierjeva transformacija z uporabo različnih okenskih funkcij.

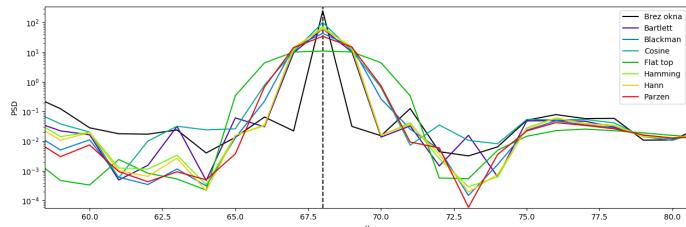


Slika 2: Originalen signal 1.

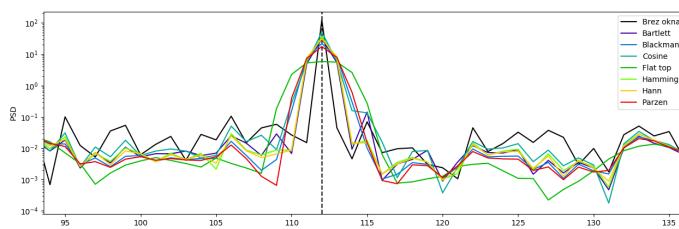


Slika 3: Fourierjeva transformacija prvega signala.

Opazimo da dva vrhova, označena s črtkano črto, zelo izstopasta od ostalih (logaritemska skala). Na grafu 4 sta prikazana od blje.



(a) vrh 1



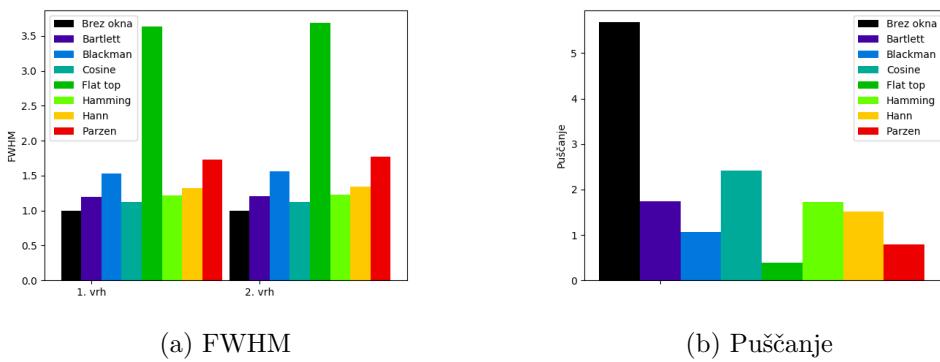
(b) vrh 2

Slika 4: Na grafih sta prikazana najvišja vrhova prvega signala.

Vidimo, da uporaba okenskih funkcij ne spremeni frekvence vrha, ampak samo njegovo višino ter širino ter količino puščanja.

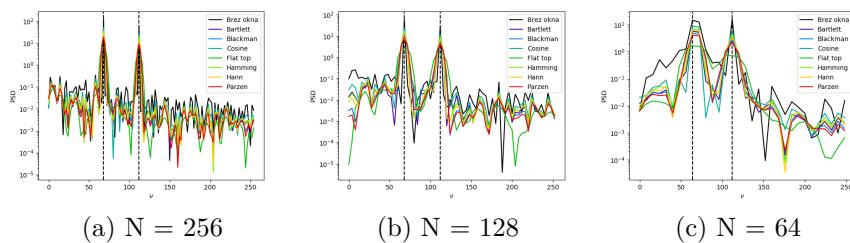
Na sliki 5 je prikazano, kakšni sta širini (FWHM) označenih vrhov ter kakšno je puščanje signala za različne okenske funkcije. Puščanje sem definiral kot kumulativno višino vseh vrhov, ki so manjši od označenih ter večji od neke mejne vrednosti. V primeru drugega signala se vidi, da takšna definicija morda ni najboljša.

Opazimo grobo zvezo, da okenske funkcije, ki zmanjšajo puščanje povečajo širino vrha.

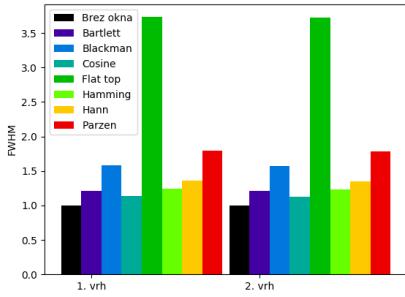
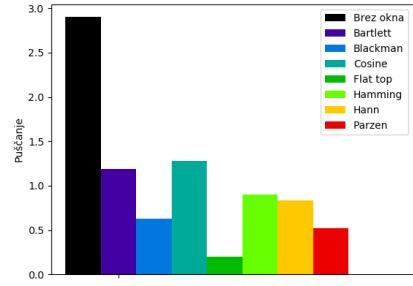
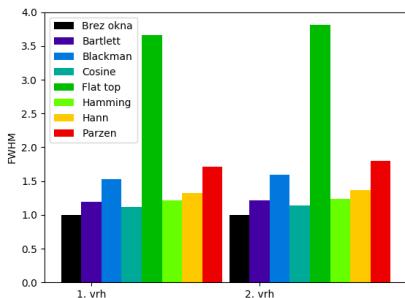
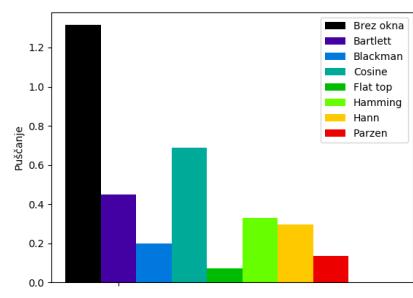
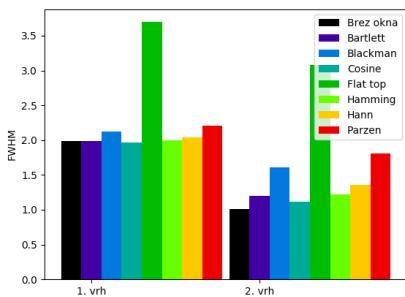
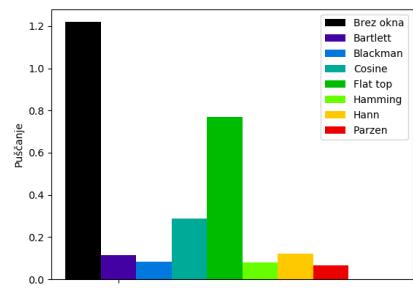


Slika 5: Na grafih sta prikazani FWHM vrhov in puščanje za različne okenske funkcije.

Zgornja analiza je bila izvedena na signalu dolžine $N = 512$. Preveril sem tudi, kako dolžina signala vpliva na dobljene karakteristike vrhov. Na spodnjih slikah je prikazana analiza za prvih N točk signala.

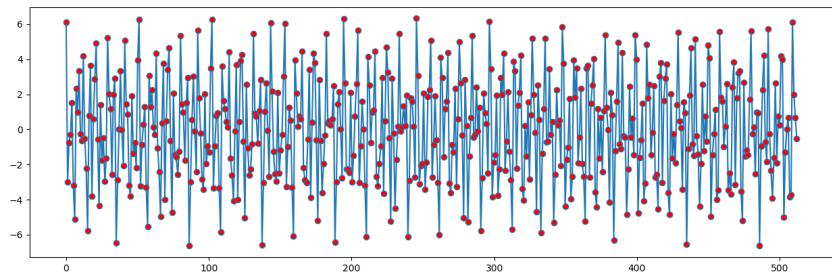


Slika 6: Na grafih so prikazane Fourierjeve transformacije prvega signala različnih dolžin N .

(a) FWHM, $N = 256$ (b) Puščanje, $N = 256$ (c) FWHM, $N = 128$ (d) Puščanje, $N = 128$ (e) FWHM, $N = 64$ (f) Puščanje, $N = 64$

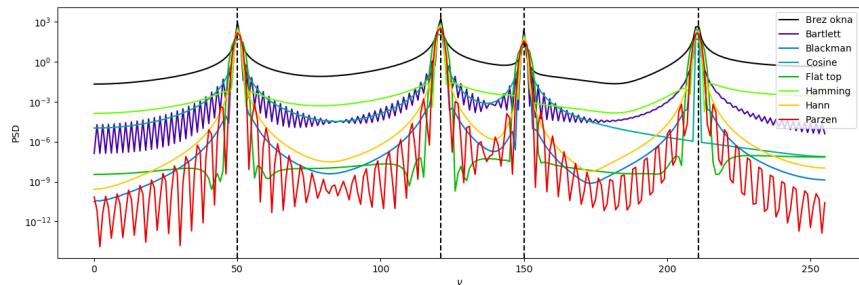
Slika 7: Na grafih sta prikazani FWHM vrhov in puščanje za različne okenske funkcije za različne dolžine N prvega signala.

Enako analizo sem naredil za drugi signal, ki je prikazan na sliki 8 in njegove Fourierjeve transformiranke na sliki 9.

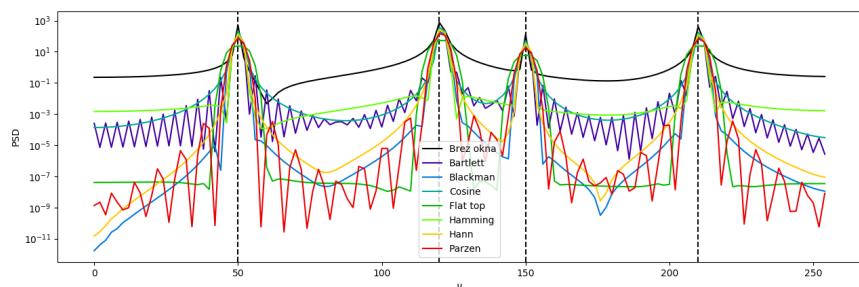


(a) Signal 2

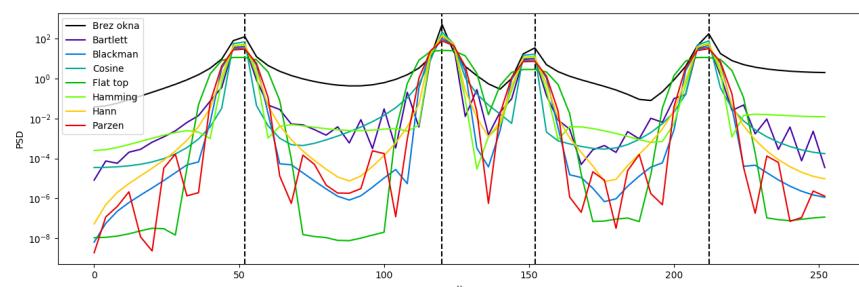
Slika 8: Na grafu je prikazan signal 2.



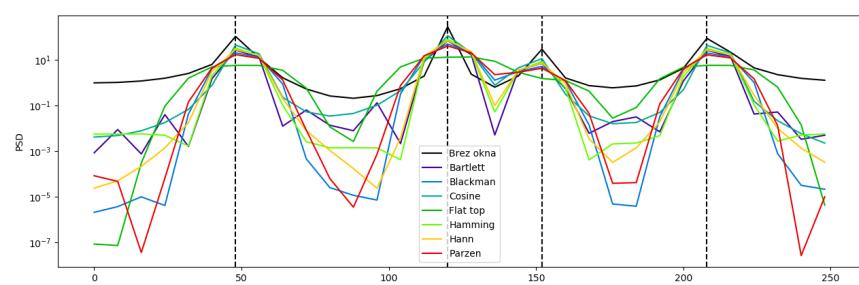
(a) Fourierjeva transformiranka, $N = 512$



(b) Fourierjeva transformiranka, $N = 256$

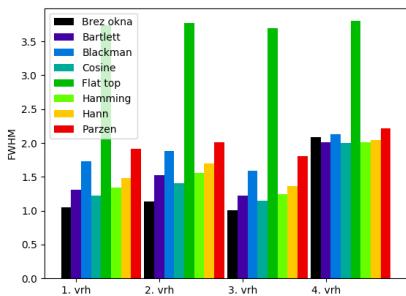


(c) Fourierjeva transformiranka, $N = 128$

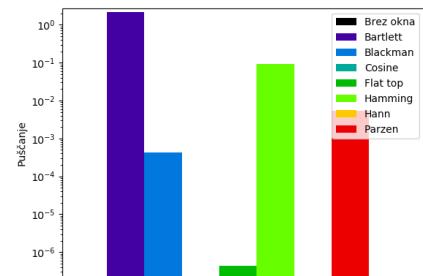


(d) Fourierjeva transformiranka, $N = 64$

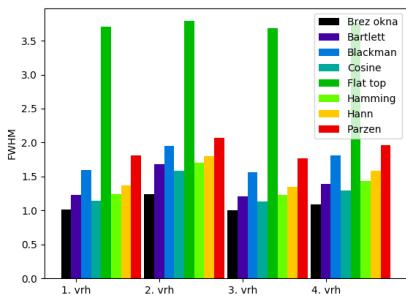
Slika 9: Na grafih so prikazane Fourierjeve transformiranke za različne dolžine signala 2.



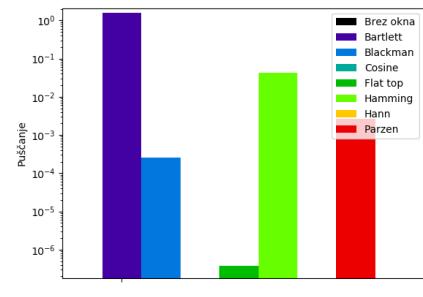
(a) FWHM, N = 512



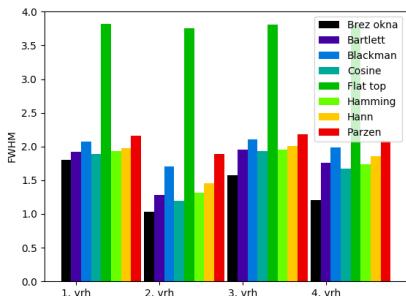
(b) Puščanje, N = 512



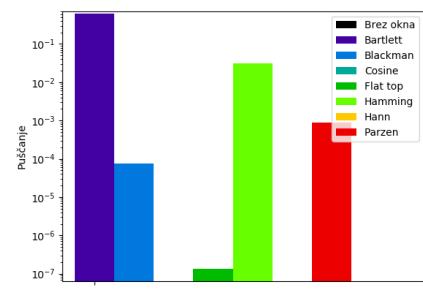
(c) FWHM, N = 256



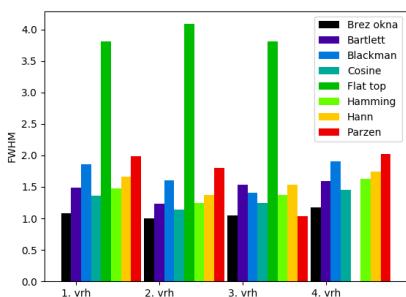
(d) Puščanje, N = 256



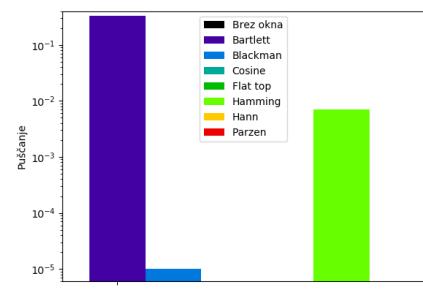
(e) FWHM, N = 128



(f) Puščanje, N = 128



(g) FWHM, N = 64



(h) Puščanje, N = 64

Slika 10: Na grafih sta prikazani FWHM vrhov in puščanje za različne oken-ske funkcije za različne dolžine N drugega signala.

MODELSKA ANALIZA I

10. naloga – Spektralna analiza in filtriranje

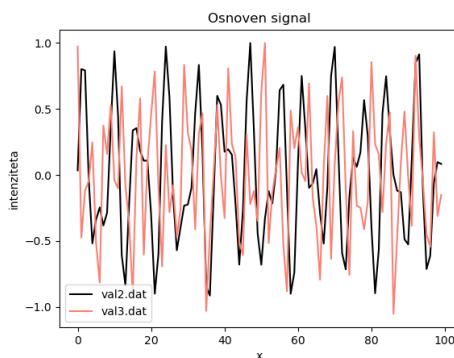
Ema Mlinar, vpisna št.: 28222071

20. december 2022

1 Okenske funkcije

Pri tej nalogi bomo obravnavali spektralno analizo. Podatke bomo poskusili čim bolje obdelati in popraviti nepravilnosti.

Pri prvem primeru dobimo dva signala, ki sta prikazana na sliki 1.



Slika 1: Dva signala *val2.dat* in *val3.dat*, ki ju bomo obdelali.

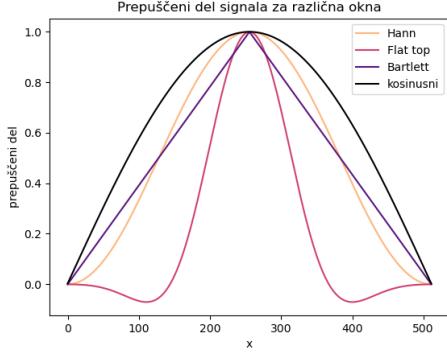
Signala izgledata zelo podobno, a s podrobnejšo analizo bomo videli, da se v njih skrivajo drugačne frekvence.

Glavno vlogo pri spektralni analizi igra Fourierjeva transformacija. Preden signal pretvorimo v frekvenčni prostor pa ga lahko še pomnožimo z različnimi okenskimi funkcijami. Na sliki 2 je prikazanih nekaj različnih okenskih funkcij, ki jih lahko uporabimo pri filtriranju.

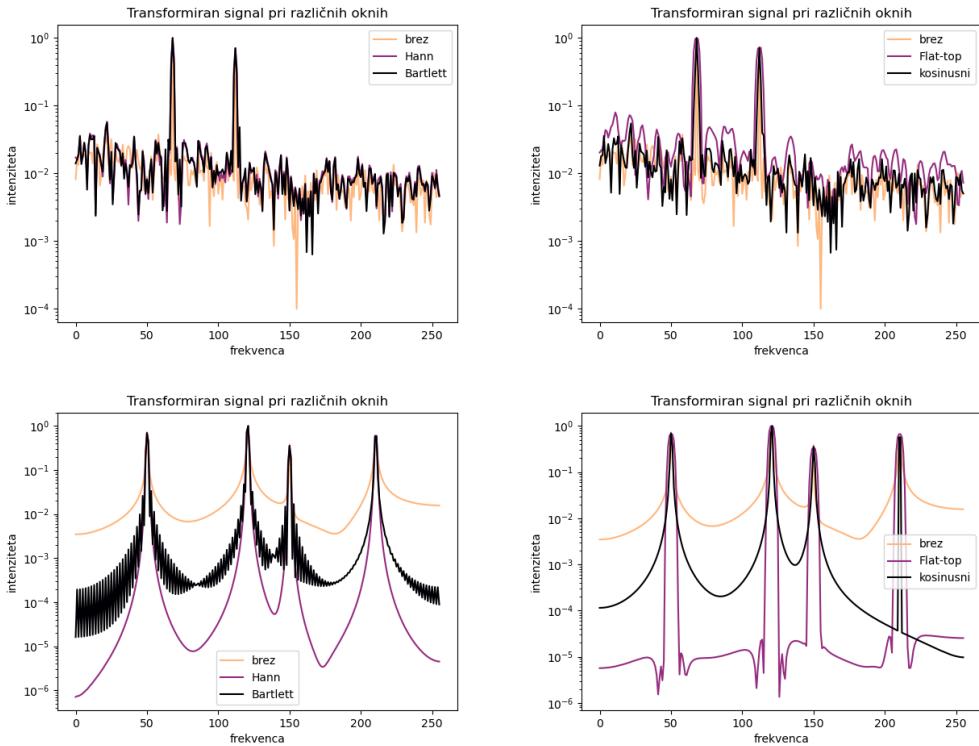
Sedaj pa uporabimo ta okna najprej na setih podatkov, glej sliko 3.

Vidimo lahko, da se v primeru prvega signala transformacije z različnimi okni skoraj ne razlikujejo (niti v logaritemski skali). Razlika pa je zelo očitna pri drugem signalu. Vidimo, da okna bolj ali manj uspešno porežejo frekvence, ki niso del vrhov, hkrati pa vrh pustijo praktično nespremenjen. Najbolj učinkovito ozadje porežeta okni Hann in Flat-top. Prav tako pa opazimo tudi zanimivo obliko pri oknu Bartlett, pojavi se namreč fina nihanja. Sklepam, da so ta nihanja posledica Fourierjevega razvoja Bartlettovega okna. Trikotnik je namreč seštevek mnogo sinusnih valov.

Poglejmo si sedaj še boj natančno, kaj se dogaja v samih vrhovih intenzitete frekvence. Bolj približan vrh vidimo na sliki 4. Podatki tu so sicer iz datoteke *val2.dat*, a zelo podobno obliko dobimo tudi pri *val3.dat*.



Slika 2: Prikaz oblike različnih oken, ki jih lahko uporabimo pri analizi podatkov.



Slika 3: Transformacija signalov ob uporabi štirih različnih oken ter primerjava brez okna. V zgornji vrstici so transformirane za datoteko *val2.dat*, v spodnji pa za *val3.dat*.

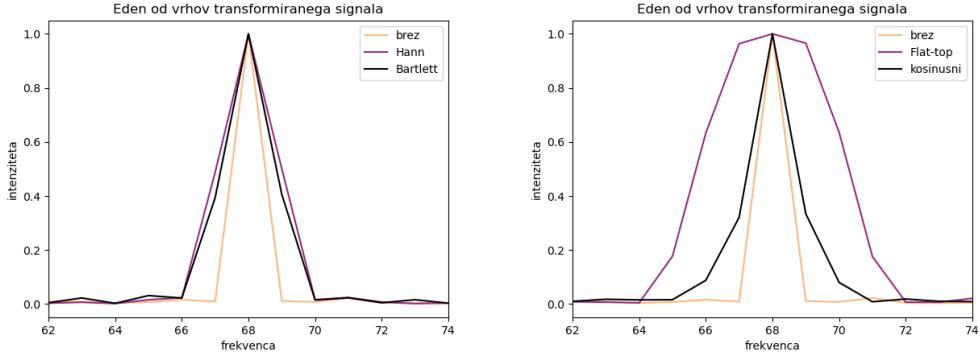
Prva stvar, ki jo opazimo je, da so vsi vrhovi špičasti, razen tisti s Flat-top oknom. Oblike vrhov so približno enake, precej pa se razlikuje širina. Poglejmo si to bolj natančno. Na grafu 5 je prikazana odvisnost FWHM širine okna za različna okna od FWHM širine vrha, ki nastane pri transformaciji. Opazimo lahko, da kljub temu, da so okna zelo različnih oblik, zelo lepo sledijo neki krivulji. Kot nam pravi teorija je ta krivulja sorazmerna funkciji $\frac{1}{x}$. Če to funkcijo prilagajamo na naše podatke, vidimo, da dobimo zelo lepo ujemanje.

Stvari pa se lahko lotimo tudi bolj sistematično. Uporabimo lahko Gaussov filter in spremojemo vrednost širine okna preko vrednosti σ , glej sliko 6.

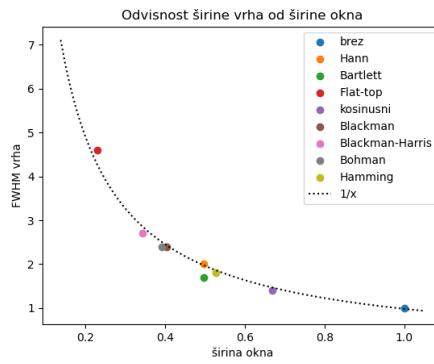
Vidimo, da spet dobimo zelo dobro ujemanje z prilagajano teoretično krivuljo.

Poglejmo si sedaj še, če ima kaj vpliva število točk (delež podatkov), ki jih uporabimo. Na sliki 7 je prikazana transformacija, ki jo dobimo.

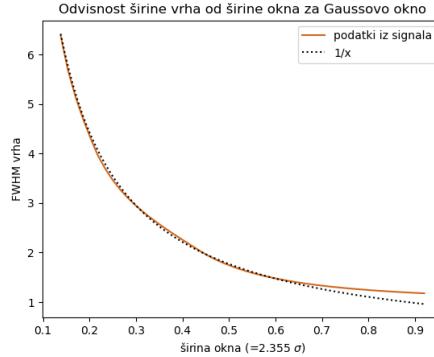
Vidimo, da so krivulje pri večjem številu točk precej bolj gladke. Poleg tega pa je markantna značilnost



Slika 4: Transformacija signala ob uporabi štirih različnih oken ter primerjava brez okna. Prikazan je bližji posnetek vrhov.



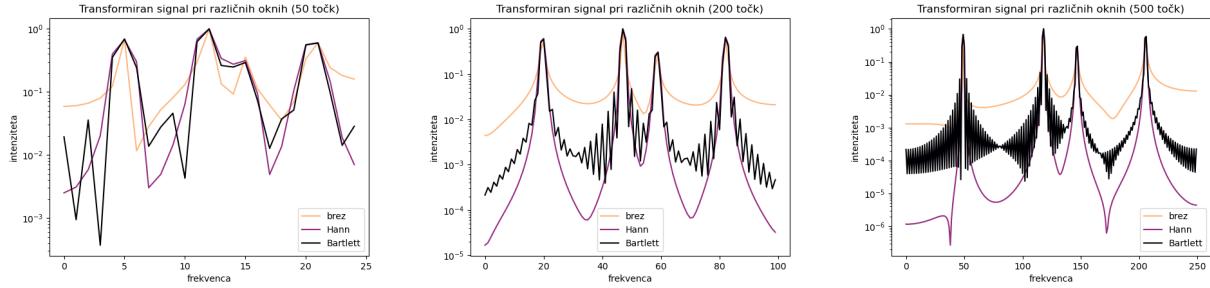
Slika 5: FWHM širina okna ter pripadajoča FWHM širina vrha, ki nastane pri transformaciji. Prikazana je vrednost za 8 različnih oken ter za transformacijo brez okna. Zraven pa je narisana še prilagajana krivulja $f(x) = \frac{a}{x}$, kjer je a prosti parameter.



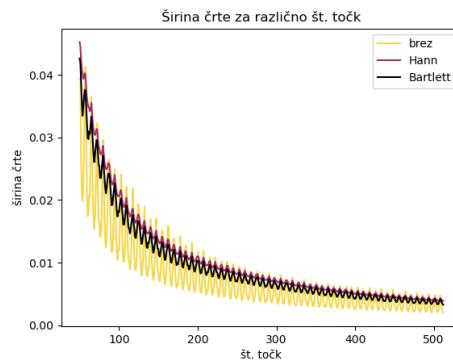
Slika 6: FWHM širina okna ter pripadajoča FWHM širina vrha, ki nastane pri transformaciji z Gaussovim oknom za različne širine okna. Zraven pa je narisana še prilagajana krivulja $f(x) = \frac{a}{x}$, kjer je a prosti parameter.

tudi širina vrhov, ki je večja pri manj uporabljenih točkah. Poglejmo si to bolj natančno, glej sliko 8. Vse krivulje (se pravi za različna okna) so na zelo podoben način odvisne od števila točk, ki jih vzamemo. Nihanja v krivulji ugibam, da so posledica tega, da vrh sestavlja le nekaj točk in je tako zelo odvisno od diskretne narave. Nihanje je večje, če ne uporabimo nobenega okna.

Kot vidimo na sliki 9 se krivuljam zelo lepo prilega funkcija $f(N) \propto \frac{1}{N}$. Če to preučimo bolj natančno, ugotovimo, da je ta zveza precej logična saj je razetros na sosednjih nekaj frekvenc vedno enak ne glede

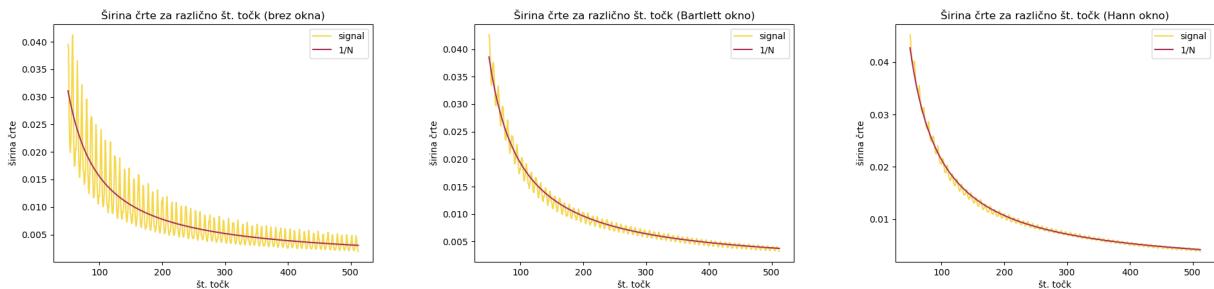


Slika 7: Frekvenčni spekter za različne deleže uporabljenih podatkov (50, 200 in 500 točk). Za primerjavo: na prejšnjih grafih je bilo uporabljenih 512 točk.



Slika 8: Frekvenčni spekter za različne deleže uporabljenih podatkov (točk) za dva različna filtra ter brez filtra.

na število točk. Hkrati pa če obe x osi na sliki 7 skaliramo na 1, pride pri manjšem številu točk interval med posameznima sosednjima frekvencama večji. Če napišemo nekaj razmerij ugotovimo, da je odvisnost od N prav N^{-1} kot smo videli na grafu 9.



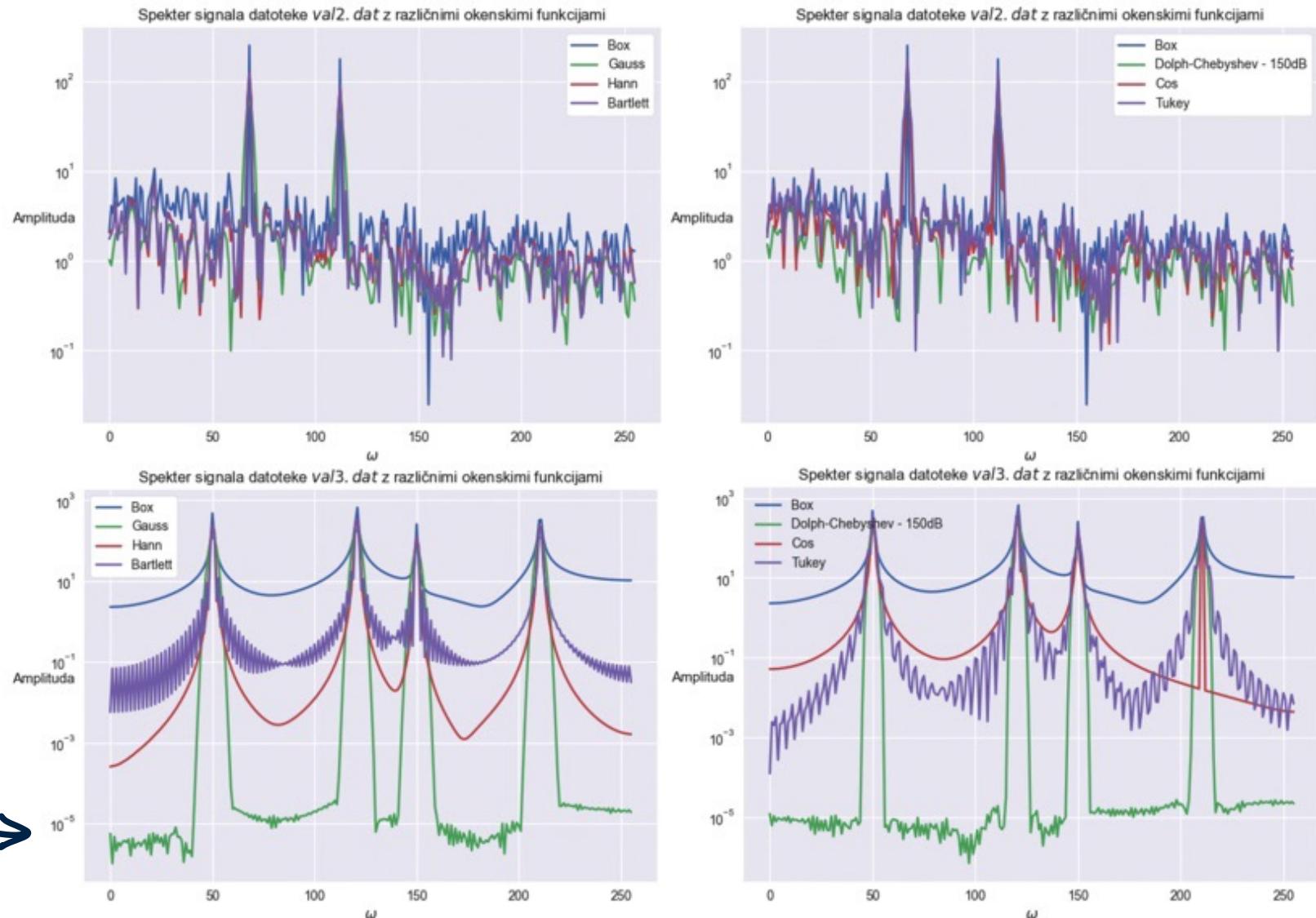
Slika 9: Frekvenčni spekter za različne deleže uporabljenih podatkov (točk) za dva različna filtra ter brez filtra. Zraven je prilagajana še krivulja $f(N) = \frac{a}{N}$, kjer je a prost parameter, ki ga prilagajamo.

2 Dekonvolucija signala

2.1 Brez šuma

Pri drugi nalogi moramo iz signala odstaniti prenosno funkcijo detektorja s pomočjo dekonvolucije. Uporabili bomo Fourierjevo transformacijo, saj moramo v frekvenčnem prostoru signal le deliti s transformiranko prenosne funkcije. Na sliki 10 je prikazana dekonvolucija signala v realnem in frekvenčnem

Filip Kržo



Slika 4: Prikaz spektrov obeh signalov po predhodnem množenju z okenskimi funkcijami.

Nádoba 2:

- Využití Wienerova filtru na rekonstrukci signálu
- Upravení místních filtrov.
- Příklad: Realizace implementačního filtru.
(analýza dnu prostoru, upoštění specifické DFT)

Blež Levenskih

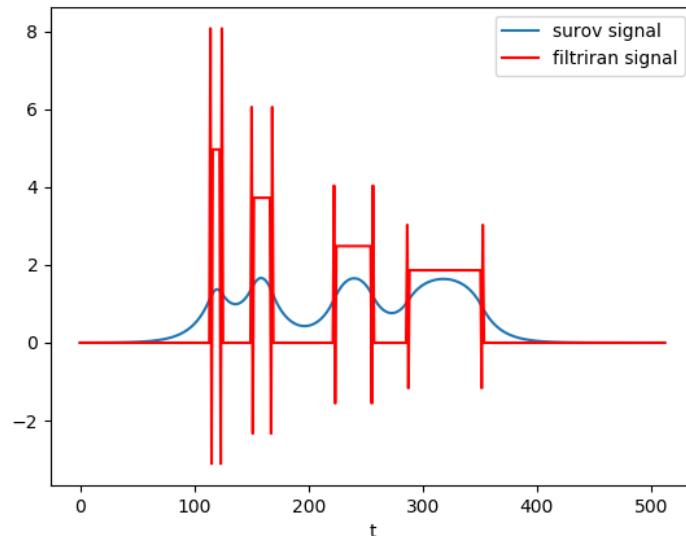
Na grafu 10, za primer, ko je $N = 64$ vidimo, da sta se dva sosednja vrhova prelila v enega ob uporabi okna, ki je preveč razširilo vrhove (Flat top). Na istem grafu tudi vidimo pomankljivosti mojega računanja puščanja, saj ker moja definicija šteje samo neizrazite vrhove ne upoštevam signala, ki je zvezen. Zato ima toliko signalov nedefinirano puščanje.

2 Wienerjev filter

Pri tej nalogi sem s pomočjo Wienerjevega filtra dekonvuliral štiri podobne signale z vedno več šuma. Prenosna funkcija je

$$r(t) = \frac{1}{32}e^{-\frac{|t|}{16}}.$$

Na sliki 11 je prikazan surov signal ter dekonvuliran signal s prenosno funkcijo $r(t)$. Lahko si mislimo, da je originalen signal škatlast.



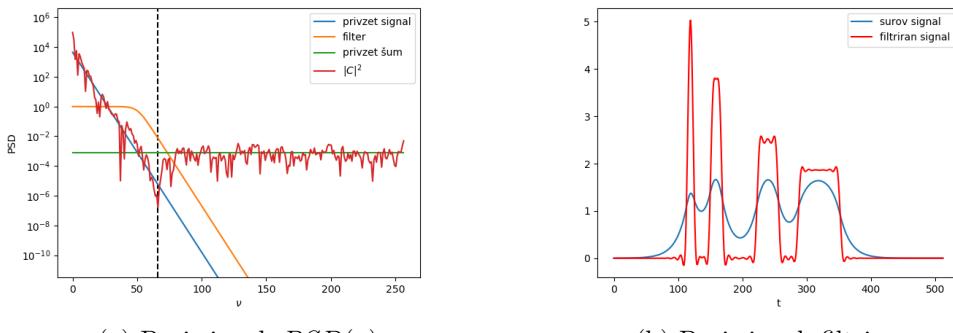
Slika 11: Surov in dokonvuliran signal.

Kljub temu, da ta signal ni (veliko) zašumljen, si poglejmo, kaj dobimo, če na njemu uporabimo Wienerjev filter. Na sliki 12 vidimo kako izgleda surov signal ($|C|^2$) v frekvenčnem prostoru. Na grafu a) lahko vidimo, da sem za šum privzel konstantno vrednost, za dejanski signal pa eksponentno odvisnost od frekvence. Črna črta prikazuje mejo, do kje sem uporabil podatke za

prilagajanje parametrov eksponentne funkcije in od kje naprej konstante. Za filter sem potem uporabil zvezo

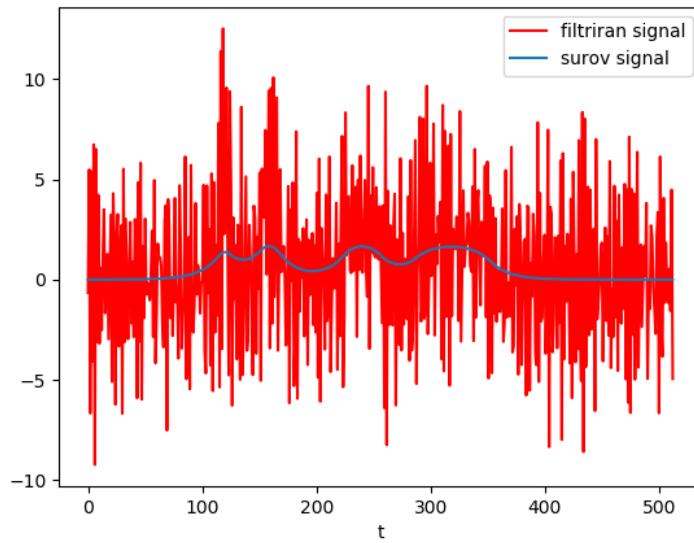
$$\Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2}.$$

Na grafu b) lahko vidimo, da s takim filtrom filtriran signal izgubi nekaj karakteristik škatlastega signala, odstrani pa tudi neželene artefakte na roku škatel, ki so tokrat nadomeščene z manjšimi prenihavanji.



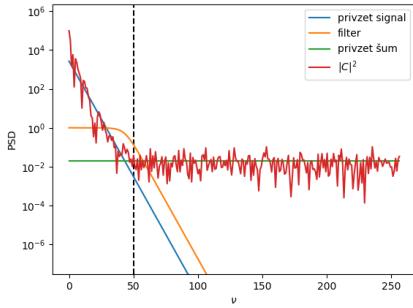
Slika 12: Na grafih je prikazan prvi signal v časovnem in frekvenčnem prostoru z uporabo Wienerjevega filtra.

Poglejmo si sedaj kaj se zgodi, če imamo bolj zašumljen signal. Če ne uporabimo filtra dobimo signal, ki je prikazan na sliki 13. Vidimo, da je dokonvoluiran signal popolnoma neuporaben.

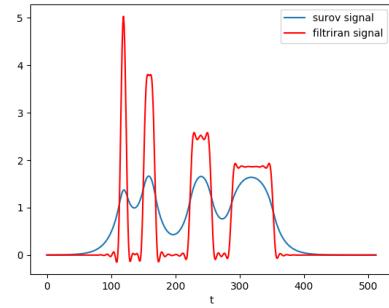


Slika 13: Surov in dokonvuliran drugi signal.

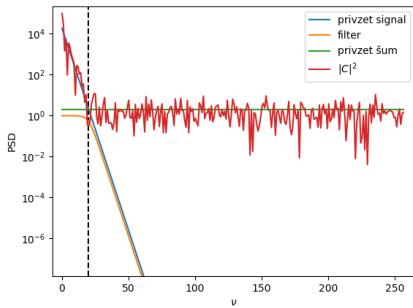
Na sliki 14 so prikazani drugi, tretji in četrti signal v frekvenčnem in časovnem prostoru skupaj z uporabljenimi filtri. Opazimo, da filtriranje odstrani večino šuma iz našega signala. Pri veliki zašumljenosti se škatlasta oblika originalnega signala skoraj povsem izgubi.



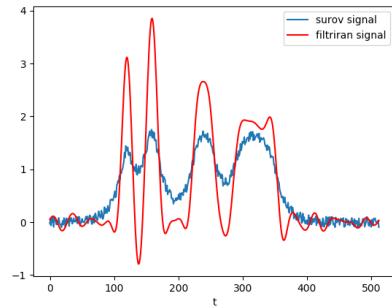
(a) Drugi signal, $PSD(\nu)$



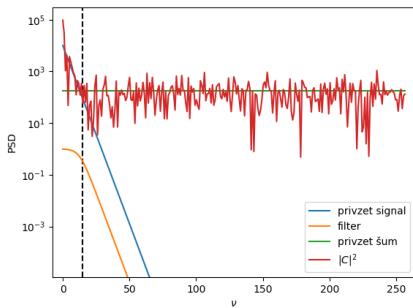
(b) Drugi signal, filtriran



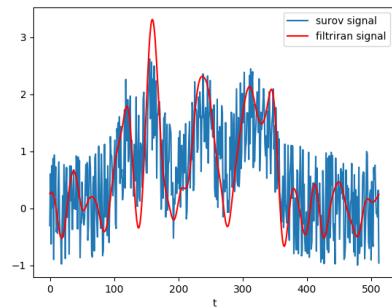
(c) Trerji signal, $PSD(\nu)$



(d) Trerji signal, filtriran



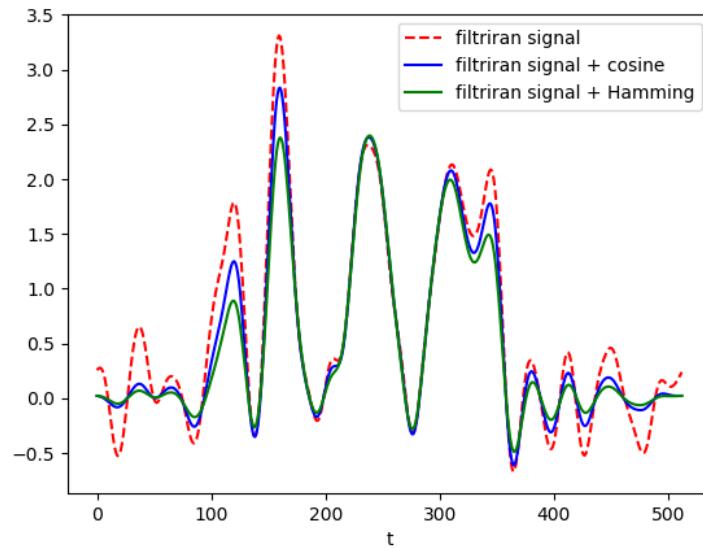
(e) Četrti signal, $PSD(\nu)$



(f) Četrti signal, filtriran

Slika 14: Na grafih so prikazani 2., 3., 4. signal v časovnem in frekvenčnem prostoru z uporabo Wienerjevega filtra.

Ker iz zadnjega signala dobimo dokaj veliko stranskih vrhov, ki spominjajo na puščanje sem na njemu poizkusil uporabiti še nekaj okenskih funkcij. Na sliki 15 vidimo, da okenske funkcije res zmanjšajo intenziteto signala, kjer ga v originalnem ni, vendar se zmanjša tudi intenziteta vrhov, ki bi morali ostati, sploh na robu signala.

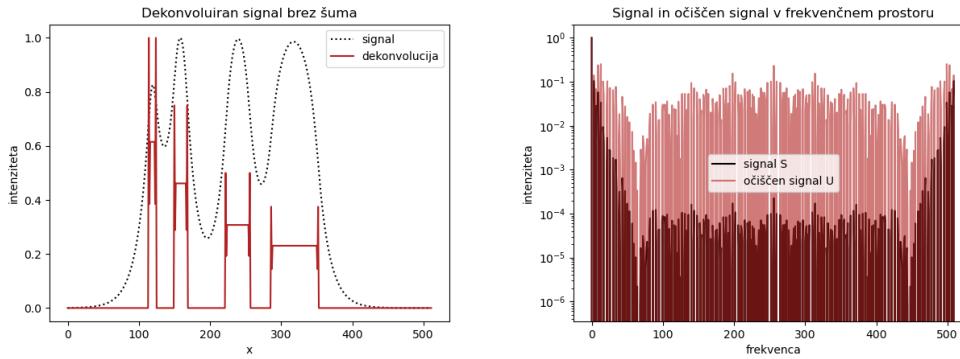


Slika 15: Dekonvuliran signal z različnimi okenskimi funkcijami.

3 Filtriranje slike

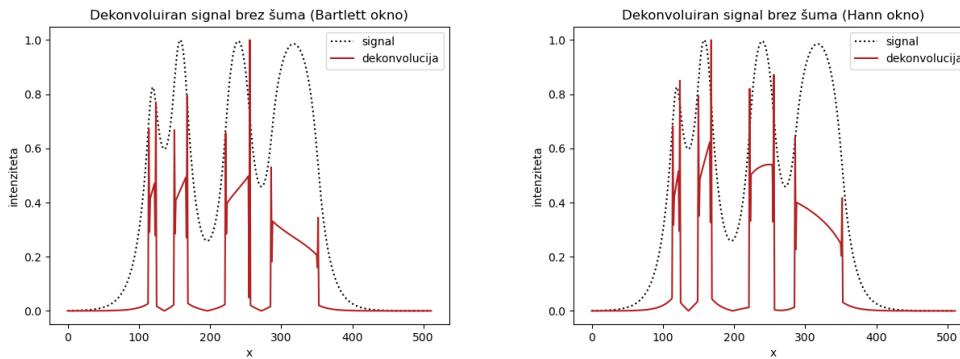
Pri tej nalogi sem poizkusil očistiti podobo Lene, ki je bila razmazana s tremi konvolucijskimi jedri, ki so prikazane na sliki 16.

prostoru.



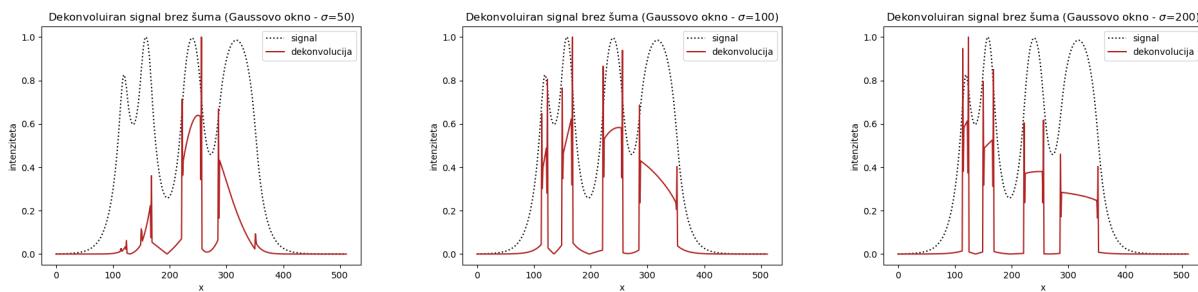
Slika 10: Dekonvolucija signalov v realnem in frekvenčnem prostoru.

Poglejmo si še, kakšen vpliv imajo okna pri rekonstrukciji tega signala, glej sliko 11.



Slika 11: Dekonvolucija signala z uporabo dveh različnih oken.

Vidimo, da uporaba oken spremeni škatlast vrh v vrh, ki ima zgornjo stranico rahlo nagnjeno. Kot si lahko ogledamo na sliki 12 je ta nagib odvisen od širine okna.

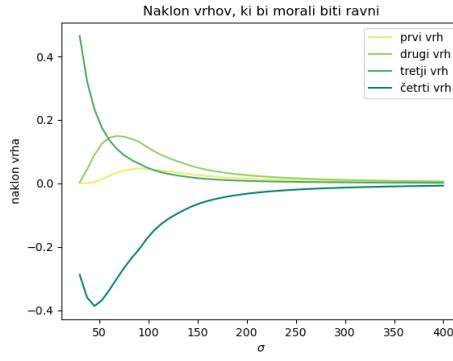


Slika 12: Dekonvolucija signala z uporabo Gaussovih oken z različno širino ($\sigma = 50, 100, 200$).

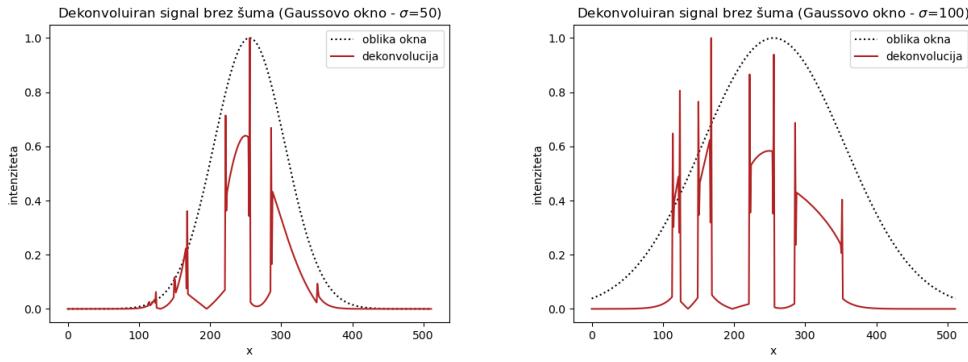
Vidimo tudi, da imajo vrhovi na levi večji nagib, na desni pa je nagib celo v drugo smer. Poglejmo si to bolj podrobno, glej sliko 13.

Vidimo, da imajo pri dovolj veliki širini okna vsi vrhovi približno raven vrh. Naklon se nato razlikuje glede na to, kateri vrh opazujemo (tako višina krivulje kot tudi faza), so pa vse enake oblike.

Vendar moram z naslednjim odstavkom iznitičiti pomen vseh teh zanimivih grafov in krivulj (tudi mene je razočaralo). Zares to, kar vidimo na grafu 13 predstavlja le različne odvode Gaussove funkcije. Oz. z drugimi besedami, neravnost vrha je posledica oblike okna, glej sliko 14.



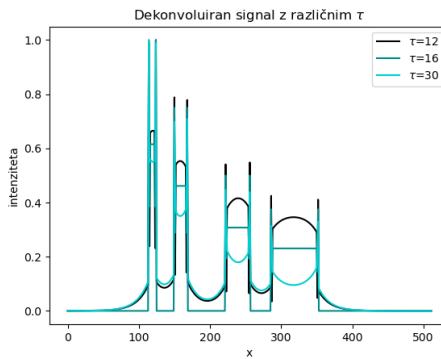
Slika 13: Naklon vrha za različne vrhove (vrh bi moral biti škatlast, torej imeti naklon 0).



Slika 14: Dekonvoluiran signal ter primerjava oblike z obliko okna.

In res, oblika v vrhovih se zelo lepo ujema z obliko okna v tisti točki (majhna razlika v naklonu je le posledica tega, da v štartu niso vsi vrhovi enako visoki - in bi zato graf bil bolj raztegnjen v y osi, če bi skalirali).

Poglejmo, kaj se zgodi, če uporabimo napačen τ za prenosno funkcijo, glej sliko 15, kjer je prikazana rekonstrukcija vrhov pri prenizkem τ ter pri previsokem τ .

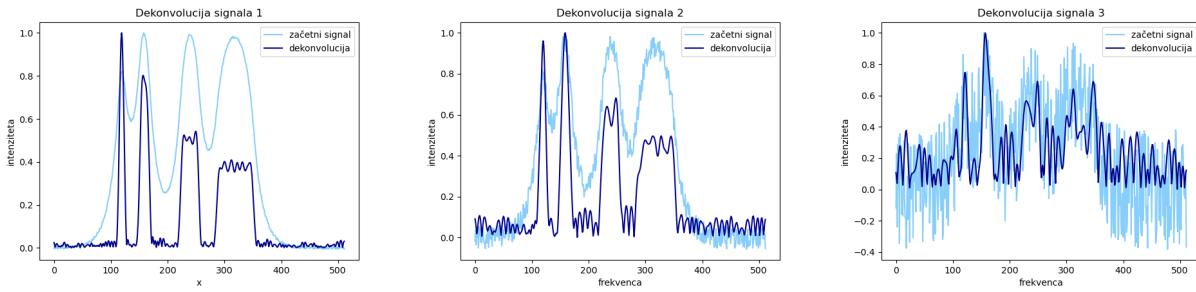


Slika 15: Rezultat dekonvolucije pri previsokem in prenizkem τ .

Vidimo, da previsoka vrednost τ povzroči konkavno obliko vrhov, prenizka pa konveksno.

2.2 S šumom

Signal pa lahko delno rekonstruiramo, tudi če so podatki (malo) zašumljeni. Na sliki 16 je prikazana dekonvolucija različno zašumljenih signalov, ki smo jih dobili pri nalogi.

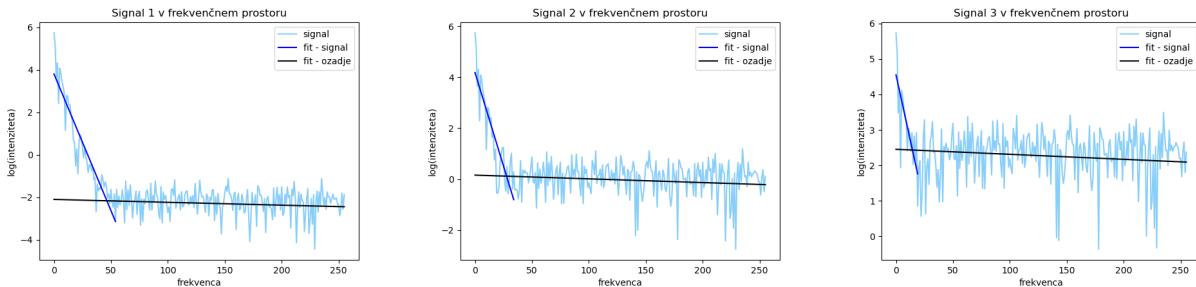


Slika 16: Dekonvolucija signalov pri različnih stopnjah šuma.

Vidimo, da malo zašumljen signal 1 uspemo zelo dobro rekonstruirati kljub šumu. Precej dobro nam uspe tudi za signal 2, nekaj podatkov pa lahko potegnemo iz zadnjega, najbolj zašumljenega signala. Tako dobro rekonstrukcijo smo lahko dosegli s pomočjo Wienerjevega filtra, s katerim pomnožimo signal pred dekonvolucijo. Wienerjev filter izvedemo v frekvenčnem prostoru in ga lahko zapišemo kot:

$$\Phi(f) = \frac{|S(f)|^2}{|S(F)|^2 + |N(f)|^2} \quad (1)$$

kjer sta absolutni vrednosti S in N oceni za velikost signala in šuma. Ti vrednosti dobimo s pomočjo prilagajanja krivulj na dveh območjih (tam, kjer vemo, da prevladuje šum in tam, kjer vemo, da prevladuje signal). Prilaganje krivulj na podatke v frekvenčni sliki je prikazano na sliki 17.



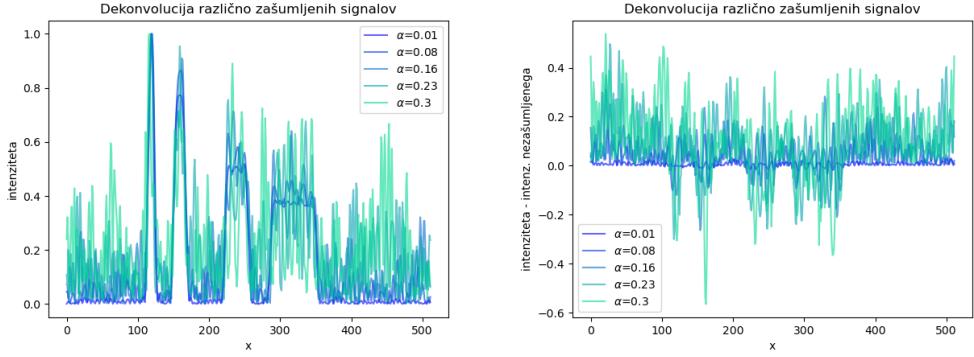
Slika 17: Določanje vrednosti S in N v Wienerjevem filtru s pomočjo prilagajanja dveh premic v logaritemski skali.

Sedaj pa poskusimo simulirati še nekaj vmesnih vrednosti za šum, poimenujmo količino šuma α (večja α pomeni več šuma, manjša pa manj). Na sliki 18 je prikazanih nekaj dodatnih dekonvolucij prvotnega signala z različno velikostjo šuma.

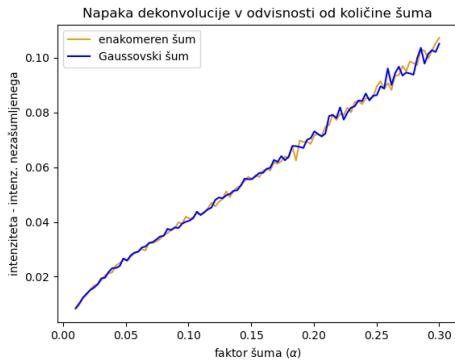
Napaka pri večjih α postane večja, vrednost ozadja pa se poviša. Opazimo pa tudi, da se sama širina vrhov ne spremeni.

Poskusimo sedaj dobiti zvezno sliko, kako se spreminja napaka (razlika od nezašumljenega modela) v odvisnosti od vrednosti parametra α . Odvisnost kaže slika 19.

Vidimo lahko, da dobimo zelo lepo premico, majhna ukrivljenost na posameznih delih premice je verjetno posledica tega, da za prilaganje podatkov za vrednosti $|N|$ in $|S|$ (zaradi prevelike količine podatkov) nismo natančne vrednosti za mejo med intervalom, ko prevladuje signal in intervalom, kjer prevladuje šum, ampak le oceno za to vrednost (ki je bila sicer linearna interpolacija med optimalnim intervalom za najmanjši α in za največji α).



Slika 18: Nekaj dodatnih primerov z umetno dodanim šumom. Na desni sliki je prikazana razlika med vrhovi brez šuma in tistimi s šumom.



Slika 19: Napaka v odvisnosti od vrednosti parametra α . Za vrednost napake je vzeta standardna deviacija razlik med dekonvolucijo ter dekonvolucijo nezašumljenega modela. Posamezna točka na grafu je povprečje 100 zagonov programa. Prikazana je krivulja za model z umetno dodanim gaussovskim šumom ter umetno dodanim šumom enakomerne porazdelitve med dvema vrednostima (razmagnjenima za standardno deviacijo ekvivalentne gaussove porazdelitve).

Na sliki 19 je prikazana tudi primerjava med gaussovskim in enakomernim šumom. V prvem primeru smo dodali gaussovsko porazdeljen naključni šum s standardno deviacijo enako α , v drugem primeru pa enakomerno porazdelitev naključnih vrednosti med $-\alpha$ in α . Vidimo, da dobimo zelo podobno za obe vrsti šuma.

3 Slike Lene

Sedaj pa bomo preizkusili dekonvolucijo signala še v 2D, poskušali bomo popraviti napake na slikah Lene. Pri tem bomo za jedra uporabili 3 napake, ki so prikazane na sliki 20.

Ta jedra najprej uporabimo za dekonvolucijo nezašumljene slike, dobimo rekonstrukcije prikazane na sliki 21

Vidimo, da Wienerjev filter bistveno pripomore k kvaliteti slike. V primeru jeder 1 in 3 sta dekonvoluciji precej uspešni, malo manj pa je uspešno pri jedru 2. Poglejmo, kaj dobimo, če dodamo še šum, glej sliko 22