

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

MODELSKA ANALIZA I

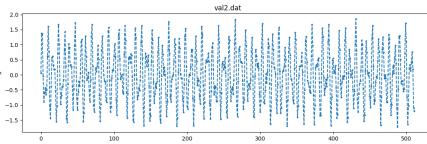
10. naloga: Spektralna analiza in filtriranje

Žiga Šinigoj, 28222025

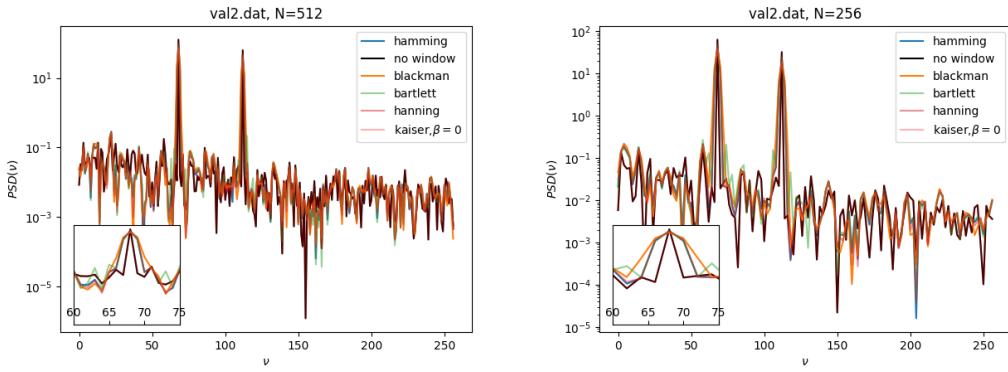
Ljubljana, december 2022

1 Frekvenčni spekter in vpliv okenskih funkcij

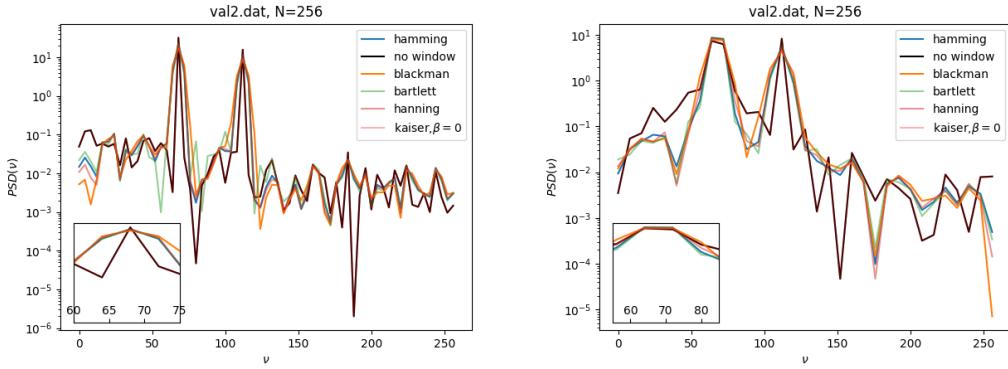
Signalu iz datotek *val2.dat* (slika 1) in *val3.dat* (slika 3(a)) je bilo potrebno določiti frekvenčni spekter in vpliv okenskih funkcij. Frekvenčni spekter signala val2 prikazuje slika 2. Če vzamem vse točke v časovni domeni (512) dobim najbolje določene vrhove oziroma frekvence signala. Skoraj vse okenske funkcije predvsem razširijo območje okrog vrhov, da ni tako ostro definirano. Pri vzorčenju signala se pojavijo naslednji efekti: Potujevanje signala, če vzorčim s frekvenco ki je manjša od dvakratnika Nyquistove frekvence. Temu se izognem tako, da vzorčim z večjo ali enako frekvenco Nyquistovi. Efekt, katerega se načeloma ni mogoče znebiti, je puščanje signala, ki nastane če ne vzorčim s periodo signala, kar pa je vedno res, saj ne poznam lastnosti signala, ki ga analiziram. Na kvaliteto spektra vpliva tudi število točk vzorčenja N. Signal sem vzorčil z manj točkami $N = 256, 128, 64$ (slika 2, 3). Pričakovano imam tudi v frekvenčnem spektru enako število točk. Če vzorčim manj točk dobim slabše definiran frekvenčni spekter, kar se na slikah lepo vidi. Različna okna različno močno zadušijo robne frekvence, odvisno od njihove oblike. To se mogoče na logaritemski skali malo manj vidi.



Slika 1: Signal iz datoteke val2.



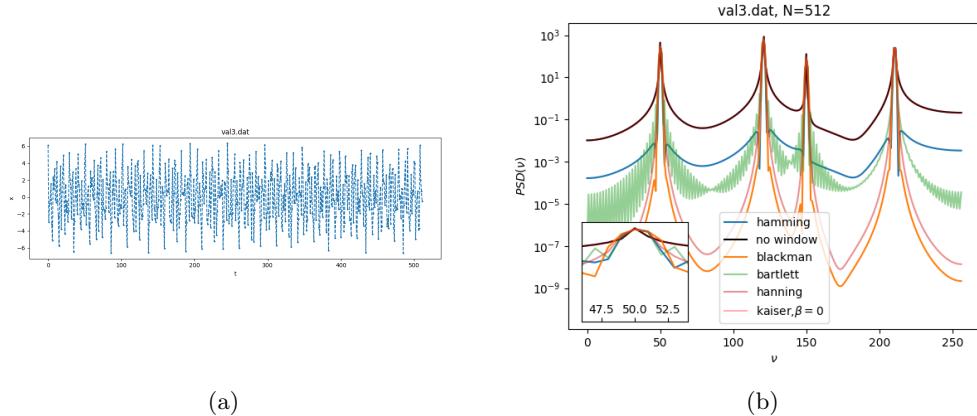
Slika 2: Frekvenčni spekter signala pri dveh različnih vzorčenjih N in pri uporabi različnih okenskih funkcij. Manjša slika predstavlja približan prvi vrh iz leve.



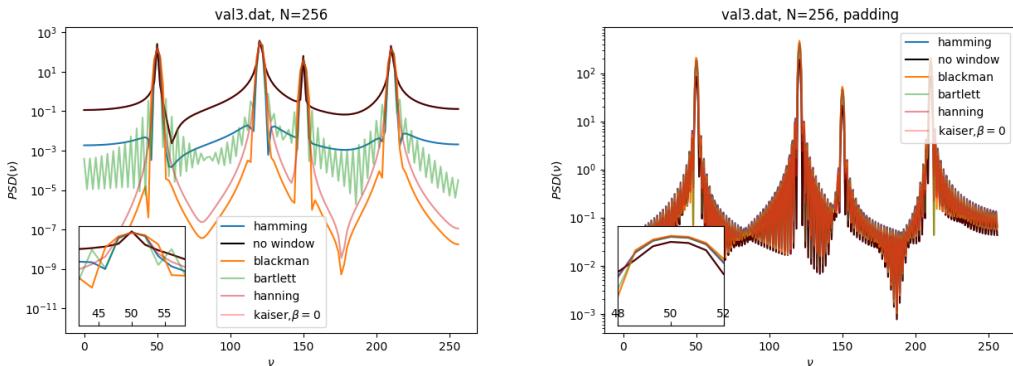
Slika 3: Frekvenčni spekter signala pri dveh različnih vzorčenjih N=128 (levo) in N=64 (desno) (napaka na grafih) in pri uporabi različnih okenskih funkcij. Manjša slika predstavlja približan prvi vrh iz leve.

Podobno analizo sem naredil tudi za signal val3. Sestavljen je iz štirih frekvenc. V tem primeru se vplivi posameznih okenskih funkcij bolj razlikujejo (slika 4(b)). Tako kot prej, okenske funkcije razširijo spekter

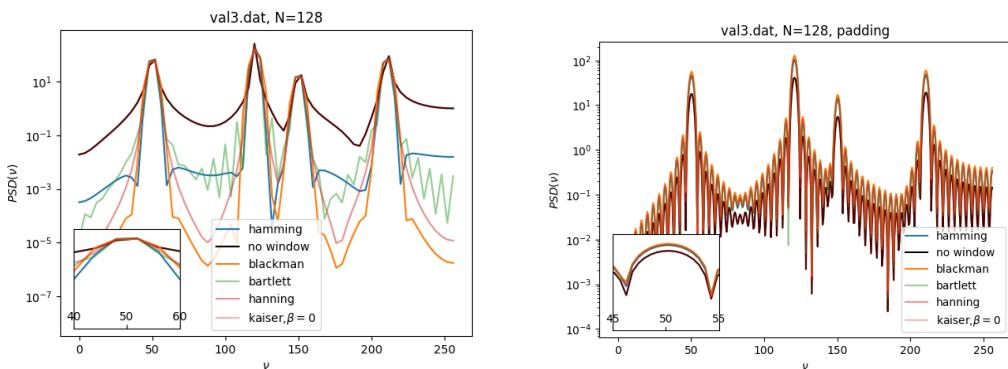
okrog vrha. V tem primeru signale, ki niso vrhovi, najbolj zadušita okni Blackman in Hanning. Okno Kaiser je za parameter $\beta = 0$ kar identiteta in vidim lahko, da ne spremeni originalnega signala. Okno Bartlett močno oscilira. Tako kot prej sem pogledal spektre v primeru ko vzorčim manj točk. V primeru da vzorčim z manj točk, vendar bi si želel spekter, ki ima veliko točk lahko uporabim metodo *zero padding*, s katero interpoliram točke v frekvenčnem spektru. Oboje je prikazano na slikah 5, 6, 7. Pri metodi *zero padding* sem dodal $512 - N$ ničel pri različnem vzorčenju N . Metoda ne izboljša kvalitete spektra, ampak samo doda oziroma interpolira vmesne točke v spektru. Z manjšanjem števila vzorčenih točk se vrhovi bolj nedoločeni saj je manj točk v frekvenčnem spektru. *Zero padding* doda točke med originalnimi točkami in mi omogoča, da bolj natančno odčitam frekvence signala.



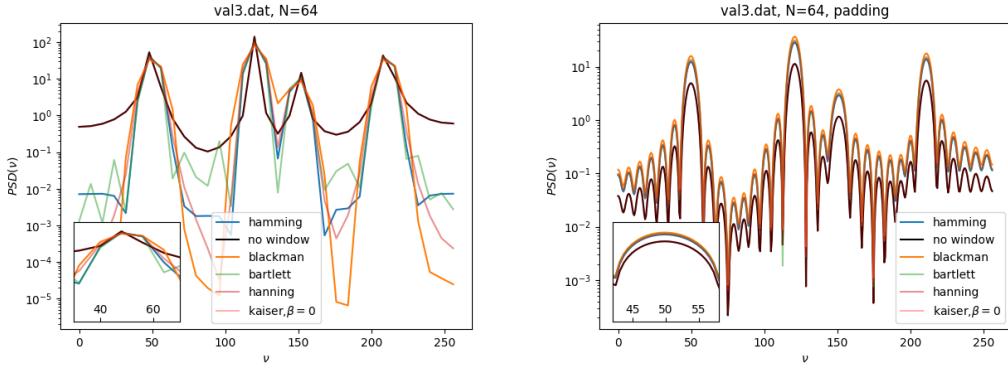
Slika 4: a) Signal iz datoteke val2. b) Frekvenčni spekter pri vzorčenju N in pri uporabi različnih okenskih funkcij.



Slika 5: Frekvenčni spekter signala pri vzorčenju N in pri uporabi različnih okenskih funkcij. Desna slika uporablja dodajanje ničel (padding) na konec časovnega intervala tako, da je celoten vektor dolg 512.



Slika 6: Frekvenčni spekter signala pri vzorčenju N in pri uporabi različnih okenskih funkcij. Desna slika uporablja dodajanje ničel (padding) na konec časovnega intervala tako, da je celoten vektor dolg 512.



Slika 7: Frekvenčni spekter signalov pri vzorčenju N in pri uporabi različnih okenskih funkcij. Desna slika uporablja dodajanje ničel (padding) na konec časovnega intervala tako, da je celoten vektor dolg 512.

2 Dekonvolucija signalov z Weinerjevim filtrom

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

$$c(t) = u(t) * r(t) + n(t) = s(t) + n(t) . \quad (1)$$

Če poznam prenosno funkcijo merilne naprave in naravo šuma, lahko rekonstruiram vpadni signal $u(t)$. Uporabim lahko konvolucijski teorem

$$s(t) = (u * r)(t) = \mathcal{F}^{-1}(U \cdot R) , \quad (2)$$

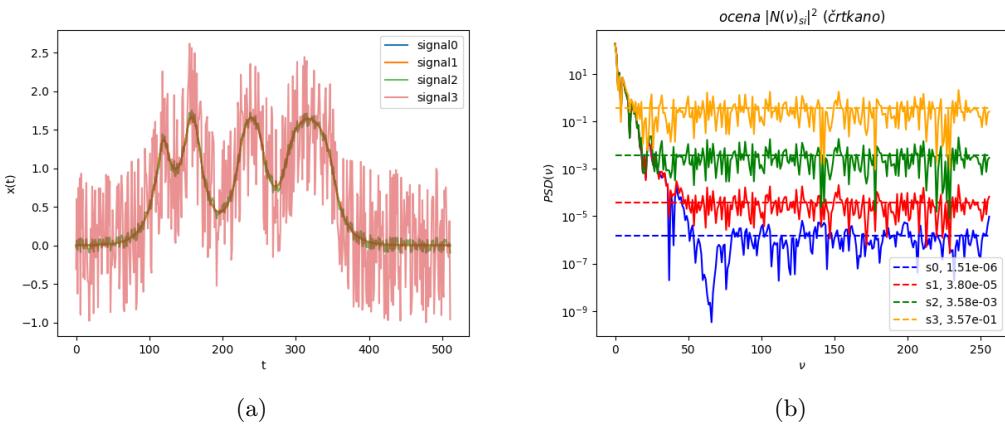
kjer sta U in R Fourierovi transformiranki u in r. Če signal ne vsebuje šuma, lahko rekonstruiram $u(t)$ kot

$$\mathcal{F}^{-1}(S/R) . \quad (3)$$

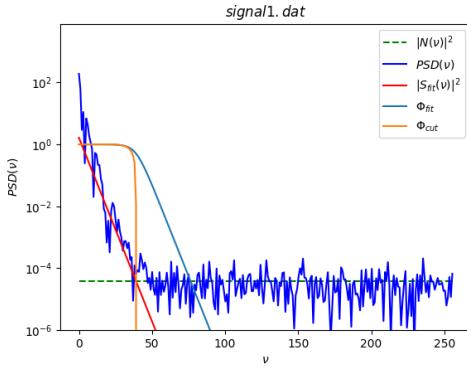
To prikazuje slika 10(a). Vhodni signal je škatlaste oblike. V primeru, ko je prisoten tudi šum je potrebno pred inverzno Furierovo transformacijo, transformiranko C pomnožiti s Weinerjevim filtrom

$$\Phi(\nu) = \frac{|S(\nu)|^2}{|S(\nu)|^2 + |N(\nu)|^2} . \quad (4)$$

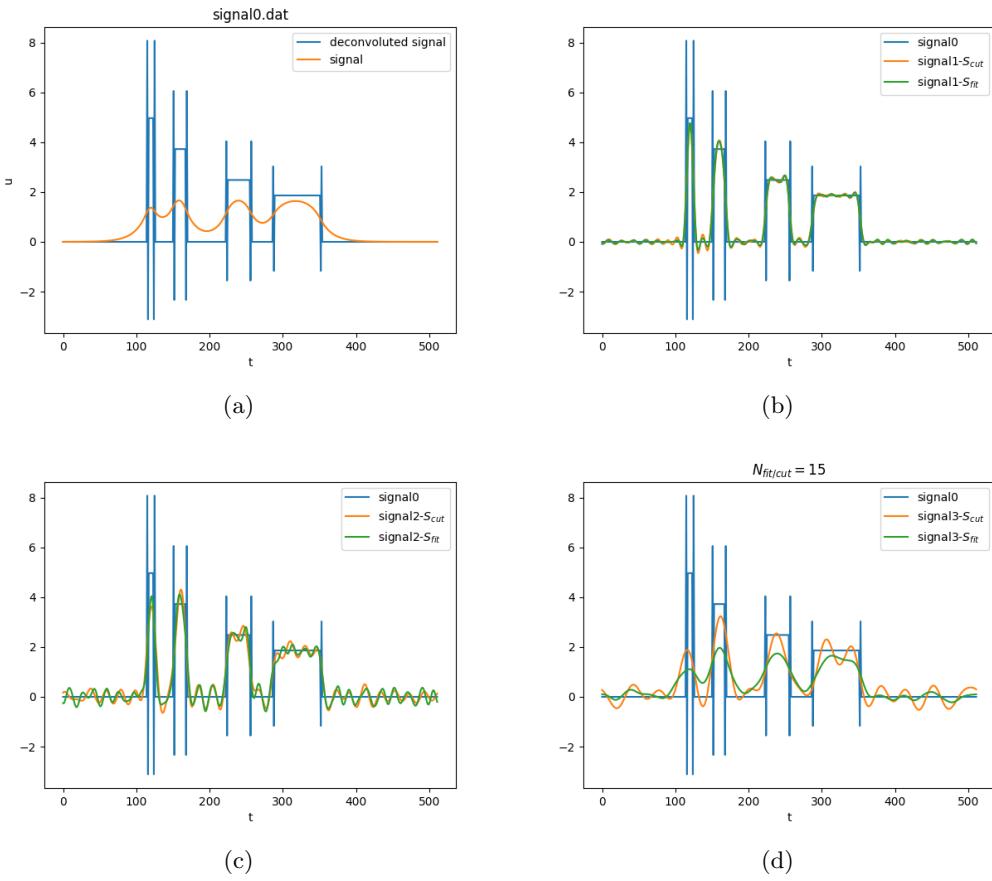
V ta namen je potrebno določiti velikost šuma. To sem naredil tako, da sem grafično ocenil mejo med šumom in signalom in potem povprečil točke (šum sem vzel za konstanto) (slika 8(b)). Naredil sem 2 filtra (slika 9). V prvem primeru sem na PSD signala prilagodil eksponentno krivuljo in potem tvoril filter Φ_{fit} . V drugem primeru sem vzel signal, do frekvence, kjer sem ocenil mejo med signalom in šumom in nato odrezal frekvence večje od mejne. Ta filter sem označil z Φ_{cut} . Dekonvolucijo signalov s šumom prikazujejo slike 10 (b), (c), (d).



Slika 8: a) Signali v časovni domeni. b) PSD signalov in ocena šuma (črtkano).

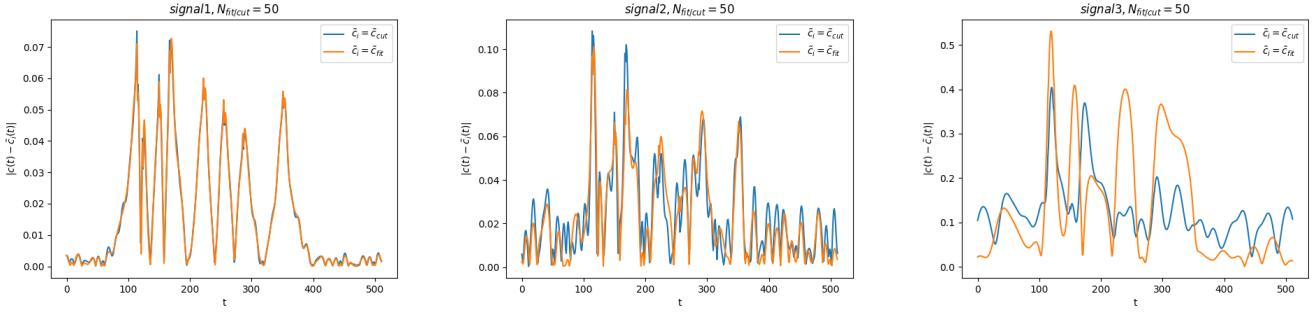


Slika 9: Frekvenčni spekter signala in konstrukcija 2 Weinerjevih filtrov Φ_{fit} in Φ_{cut} za primer signala *signal1.dat*, kjer sem odrezal signal na ocenjeni meji šuma (Φ_{cut}) in kjer sem konstruiral filter iz fita eksponentne funkcije na signal (Φ_{fit}).



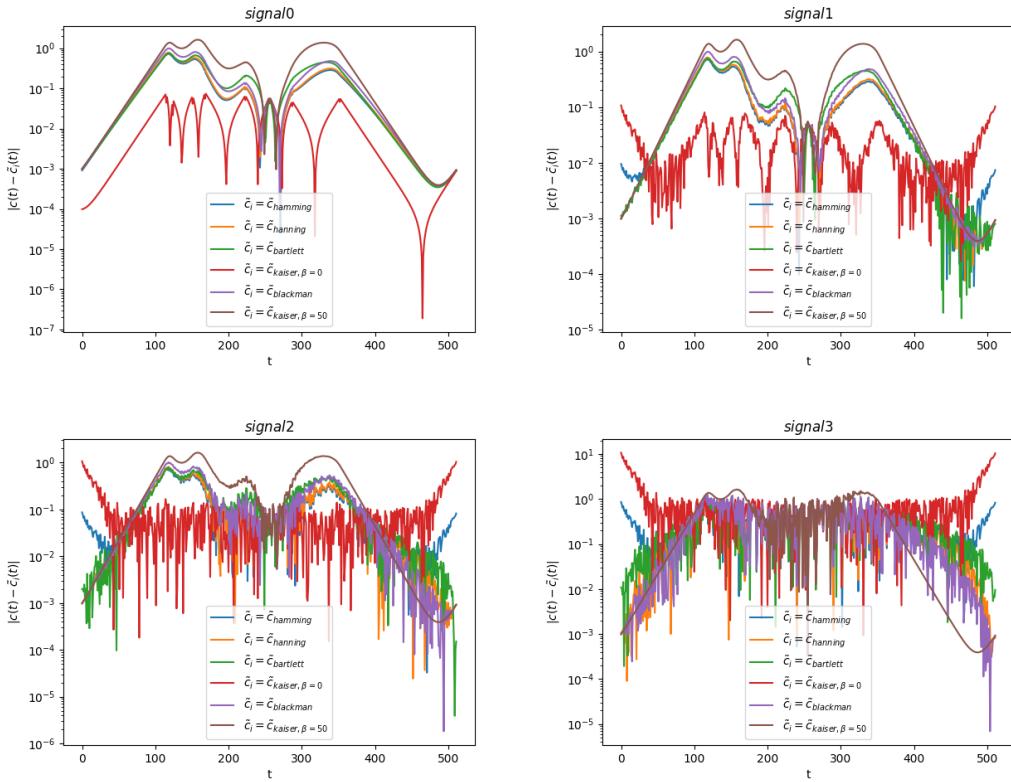
Slika 10: Dekonvolucija različno zašumljenih signalov z dvema Weinerjevima filtrovoma. $N_{fit/cut}$ označuje mejo med signalom in šumom v frekvenčnem spektru. Če ni posebej poudarjeno velja $N_{fit/cut} = 50$

Filtra približno enako dobro opiseta dekonvoluiran signal. Z večanjem šuma dobim slabšo rekonstrukcijo signala. V primeru ko imam najbolj zašumljen signal je po mojem mnenju bolje uporabiti filter Φ_{cut} . Zanimalo je pogledati tudi odstopanja od začetnega zašumljenega signala. V primeru, ko imam opravka s signalom, ki ni zašumljen, bi s konvolucijo škatlastega signala dobil začetni signal (oranžen na sliki 10(a)). V tem primeru ni bilo odstopanj. Pogledal sem odstopanja začetnega signala od ponovne konvolucije dekonvoluiranega signala s prenosno funkcijo $r(t)$, ko imam prisoten šum. Ta so tudi na nek način pokazatelj, kako uspešna je dekonvolucija. Odstopanja so prikazana na sliki 11. Pogledal sem tudi odstopanja v primeru, ko ne uporabim Weinerjevega filtra, ampak samo okenske funkcije (slika 12). Filter Kaiser, ki je identiteta, pove samo odstopanja, če ne upoštevam Weinerjevega filtra. Odstopanja se pričakovano večja z večanjem šuma.



Slika 11: Absolutna razlika med originalnim signalom in konvolucijo dekonvoluiranega signala \tilde{c} s prenosno funkcijo $r(t)$ pri uporabi dveh filtrov.

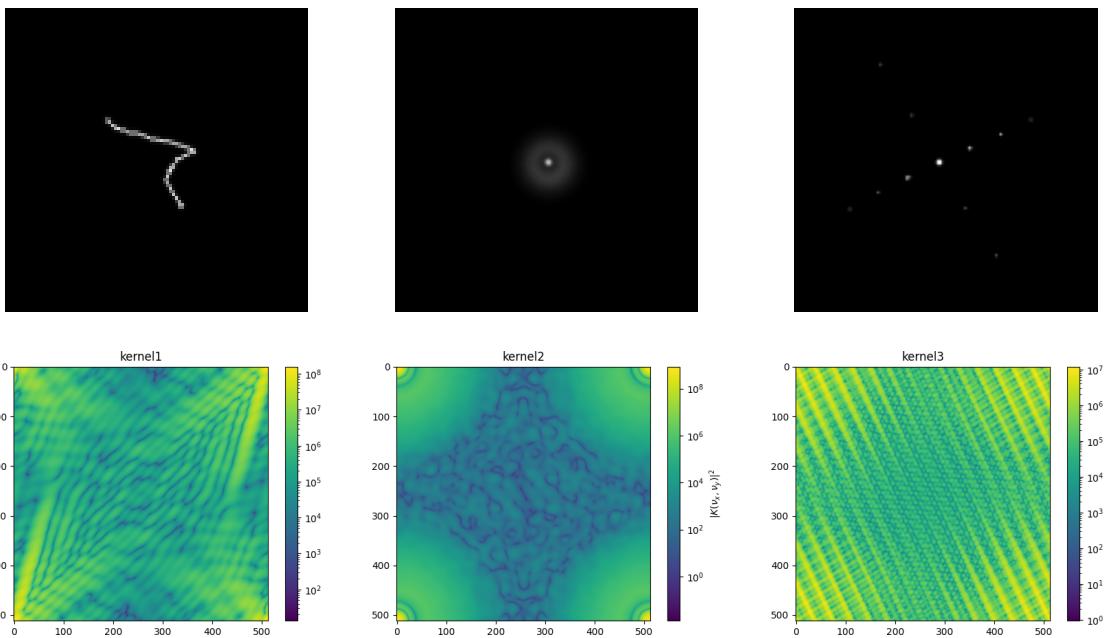
Če ne uporabimo Weinerjevega filtra, potem najboljši opis signala $u(t)$ dobim če uporabim Hamming in Hanning okno.



Slika 12: Absolutna razlika med originalnim signalom in konvolucijo dekonvoluiranega signala \tilde{c} s prenosno funkcijo $r(t)$ pri uporabi različnih okenskih funkcij brez Weinerjevega filtra.

3 Dekonvolucija slik z Weinerjevim filtrom

Filtriranje lahko razširim na 2D primer in ga uporabim za čiščenje slik, ki so lahko popačene zaradi gibanja, tresenja objektiva in ostalih motenj. V primeru, da poznam prenosno funkcijo lahko z dekonvolucijo rekonstruiram originalno fotografijo. Najprej sem pogledal Fourierove transformiranke konvolucijskih jeder (slika 13). Tako kot v 1D primeru, je potrebno določiti PSD šuma, ki naj bo tudi tokrat konstanta. To sem naredil grafično in sicer tako, da sem vsakič pogledal PSD slike in potem izbral 2 območji (2 štirikotnika) iz katerih sem izračunal povprečno vrednost. To sem vzel za ozadje oziroma šum in nato naredil Weinerjev filter. Najprej sem poskusil očistiti slike, ki so posledica tresočega objektiva (slika 14). Z večanjem šuma dobim manj ostro sliko. Ker je potrebno vsakič oceniti šum, so slike različno temne. Če vzamem manjšo



Slika 13: Zgoraj: Konvolucijska jedra v časovni domeni (približano). Spodaj: Konvolucijska jedra v frekvenčnem prostoru. Pozitivnim frekvencam ustreza en kvadrant spektra. Kernel1 predstavlja tresoč objektiv, kernel2 slab fokus, kernel3 uklonsko mrežico.



Slika 14: Čiščenje slik iz tresočega objektiva (kernel1). 1.vrstica: Popačene slike. 2. vrstica: Očiščene slike. 3.vrstica: Očiščene slike z razširjitvijo slike (padding), kjer sem dodal na vsak rob 100 povprečnih vrednosti slike okrog popačene slike. Količina Gaussovega šuma ($RMS = 0, 4, 8, 16$) narašča od leve proti desni.

vrednost šuma bom obdržal poleg šuma tudi več detajlov na sliki in ta bo ostrejša vendar bolj temna. V primeru ko sem dodal konstantne vrednosti okrog slike, sem dobil sliko, ki je manj valovita, ne zdi se mi pa bolj ostra od slike nad njo. Sivin ob robu mi ni uspelo odpraviti s to metodo. Na sliki zelo vpliva ocena šuma. Najslabše rezultate čiščenja sem dobil iz popačenih slik zaradi fokusa (slika 15). V tem primeru je



Slika 15: Čiščenje slik iz slabo fokusiranega objektiva (kernel2). 1.vrstica: Popačene slike. 2. vrstica: Očiščene slike. 3.vrstica: Očiščene slike z razširitvijo slike (padding), kjer sem dodal na vsak rob 50 povprečnih vrednosti slike okrog popačene slike. Količina Gaussovega šuma ($RMS = 0, 4, 8, 16$) narašča od leve proti desni.

bilo potrebno izbrati veliko vrednost šuma in posledično ozek ozek filter, ki je odrezal večino podrobnosti slik. Če se naredil malce širši filter je celotna slika postala siva. V tem primeru nisem izboljšal začetnih slik. Tudi dodajanje konstantnih vrednosti okrog slike ne bistveno izboljša slike.

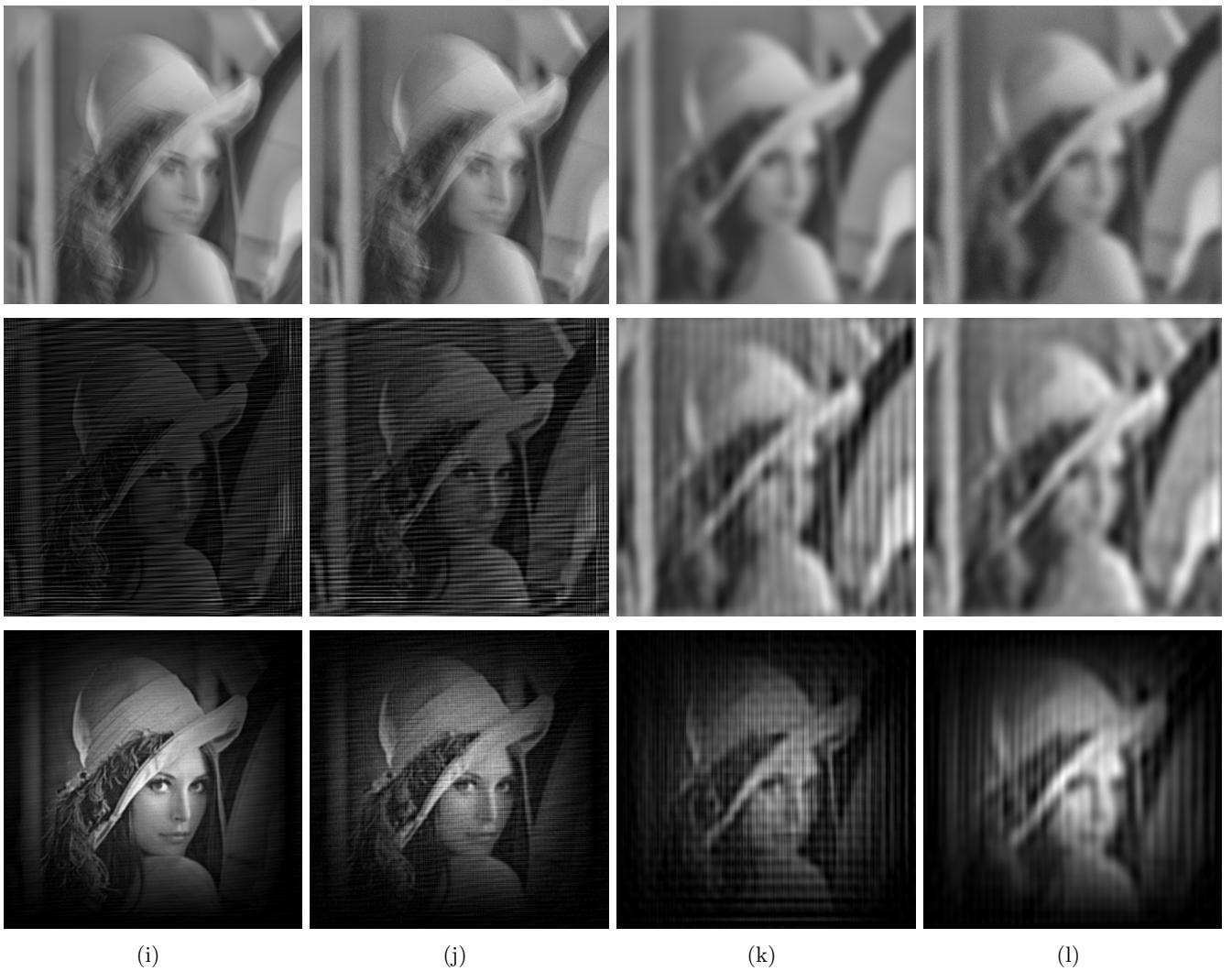
Čiščenje slike zaradi efekta uklonske mrežice prikazuje slika 17. V tem primeru se najbolj vidi izboljšavo slike, če dodam konstantne vrednosti okrog slike (razširim sliko). Čiščenje slike, ki ima dodano periodično motnjo prikazuje slika 17. Poleg razširitve slike lahko uporabim okenske funkcije. Za najboljšo, v smislu, da najbolj izostri sliko, se je izkazala okenska funkcija Blackman. Ta precej poreže robne frekvence (slika 18) ter precej izostri center slike. Največjo izboljšava se pokaže pri slikah, ki so razmazane zaradi slabega fokusa. Z uporabo Blackman okna dobim bolj ostro sliko v primeru Kernel2, RMS=0,4.



Slika 16: Čiščenje slik iz aparata z uklonskim pojavom (kernel3). 1.vrstica: Popačene slike. 2. vrstica: Očiščene slike. 3.vrstica: Očiščene slike z razširitvijo slike (padding), kjer sem dodal na vsak rob 100 povprečnih vrednosti slike okrog popačene slike. Količina Gaussovega šuma ($RMS = 0, 4, 8, 16$) narašča od leve proti desni.



Slika 17: Čiščenje slik iz aparata s periodično motnjo. 1.vrstica: Popačene slike. 2. vrstica: Očiščene slike. 3.vrstica: Očiščene slike z razširitvijo slike (padding), kjer sem dodal na vsak rob 100 povprečnih vrednosti slike okrog popačene slike.



(i)

(j)

(k)

(l)

Slika 18: Čiščenje slik z uporabo okenske funkcije Blackman. 1.vrstica: Popačene slike, 2.vrstica: Očiščene slike (nekatere iz zgornjih slik). 3.vrstica: Očiščene slike z uporabo okenske funkcije Blackman. Stoplci: i) Kernel1, RMS=0 j) Kernel1, RMS=4, k) Kernel2, RMS=0, l) Kernel2, RMS=4.

3.1 Optimalno določanje šuma

Določanje šuma grafično ni idealna metoda, zato sem poskusil implementirati algoritem, ki bi sam našel optimalno vrednost. V ta namen sem našel originalno sliko Lene (lahko bi vzel tudi najboljši približek, RMS=0). Za oceno kvalitete slike potrebujem matematično mero. Za slike se očitno uporablja mera SSIM, katere definicije ne bom podajal. Uporabil sem vgrajeno metodo, ki izračuna vrednost SSIM. Podati je treba dve sliki, najboljšo (original) sliko in popačeno sliko in mera SSIM pove kako "dobra" oziroma podobna je popačena slika originalni slikam. Za dano popačeno sliko sem izračunal PSD in potem vzel območje 25x25, ki sem ga premikal po diagonali, ob vsaki iteraciji za en korak. Za dano iteracijo sem izračunal povprečno vrednost območja in jo progglasil za vrednost šuma. Nato sem naredil dekonvolucijo in izračunal SSIM med originalni in filtrirano sliko. Če je vrednost $SSIM = 1$ potem sta sliki enaki. Če je $SSIM = 0$, potem sta sliki povsem različni (ni nobenih podobnosti). Po koncu iteracije sem vzel vrednost šuma, ki je povzročila, da je bil SSIM najbližje vrednosti ena. Rezultate med oceno šuma "na roke" in avtomatsko oceno prikazuje slika 20. Dobljene slike niso bolj ostre kot tiste, katerim sem šum ocenil na roke. Čeprav niso slabi približki, so ti primeri bolj izjeme, saj sem v veliko primerih dobil samo sivo sliko. Mislim, da je največja težava matematično določiti kaj je prava mera za ostrino slike, kot jo dojema človeško oko.



Slika 19: Originalna slika Lene.



Slika 20: 1. vrstica: Očišćene slike (izbrane iz zgornje množice slik). 2.vrstica: Očišćene slike z avtomatskim določanjem šuma, mera SSIM = 0.654, 0.637, 0.611, 0.563 (od leve proti desni). Stolpci: e) kernel3, RMS=0 f) kernel3, RMS=4 g) kernel3, RMS=8 h) kernel3, RMS=16