10. naloga - Spektralna analiza in filtriranje

Tina Klobas

18. avgust 2020

1 Opis problema

Pri tej nalogi si bomo pogledali tri različne primere spektralne analize in filtriranja podatkov. Z diskretno Fourierovo transformacijo:

$$F_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \mathrm{e}^{-2\pi \mathrm{i} jk/N}, \tag{1}$$

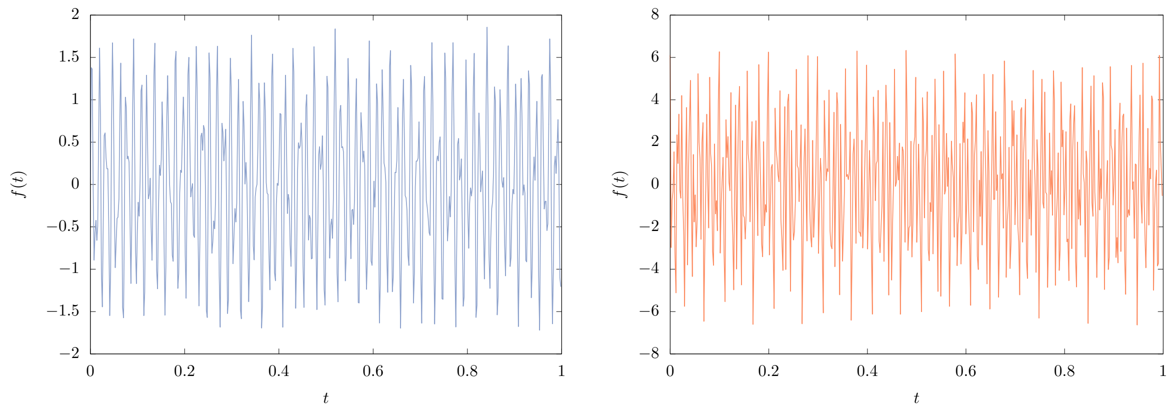
vhodnega signala dobimo frekvenčni spekter, s čimer dobimo informacijo o fazah in amplitudah signala. Moč spektra izračunamo kot:

$$P_k = \frac{1}{2}(|F_k|^2 + |F_{N-k}|^2). \tag{2}$$

V prvem delu si bomo pogledali, kako razpon podatkov in izbira okenskih funkcij vpliva na dobljen spekter. V naslednjem odseku rekonstruirali vpadne signale iz dobljenih signalov, iz katerih bomo z Wienerjevim filtrom odstranili šum. Na koncu si bomo pogledali še rekonstrukcijo zašumljenih slik.

2 Določanje frekvenčnega spektra

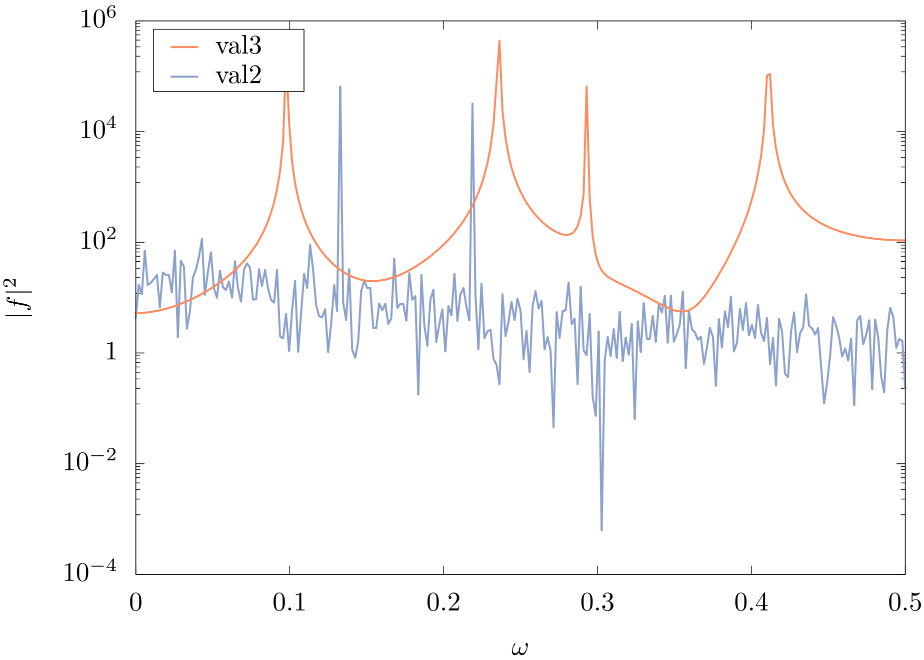
V datotekah `val2.dat` in `val3.dat` imamo shranjena signala s 512 točkami in jima želimo določiti frekvenčni spekter. Izhodni signal, je prikazan na grafu 2.1.



Slika 2.1: Vhodna signala – na levi `val2.dat` in na desni `val3.dat`.

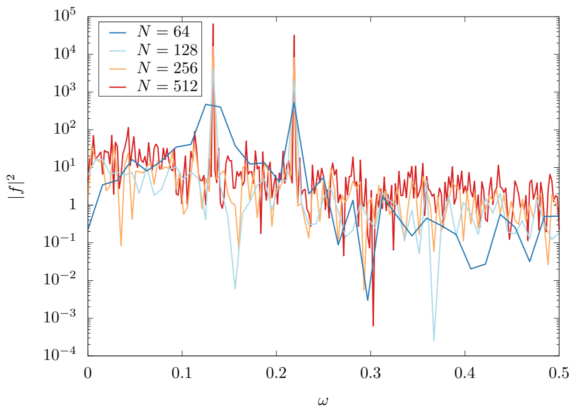
2.1 Reševanje in rezultati

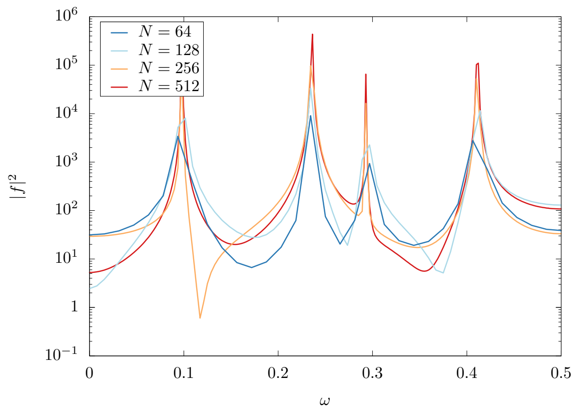
Na sliki 2.2 sta prikazana spektra moči signalov v logaritemski skali.



Slika 2.2: Spektra signalov v logaritemski skali.

Fourierova transformacija deluje dobro za periodične funkcije – v našem primeru vidimo, da ima dobljen graf za prvi spekter – *val2* – dva izrazita vrha, kar kaže na periodičnost spektra, medtem ko so vsi štirje vrhovi drugega spektra – *val3* – precej razširjeni, kar opozarja na nasprotno.
Najprej lahko preverimo, kaj se zgodi, če vzamemo krajše intervale točk; $62, 128, 256$. Naslednji graf 2.3 prikazuje spektre izračunane za štiri različne intervale točk. Za prvi spekter vidimo, da ima bolj ali manj enako obliko in širino vrhov – razen za interval z 64 točkami, ki je bil očitno prekratek in smo izgubili informacije o spektru. Za drugi spekter, pa vidimo, da se vrhovi zelo razširijo in spremenijo obliko – sumimo, da že v začetku nismo ujeli periodičnosti signala.





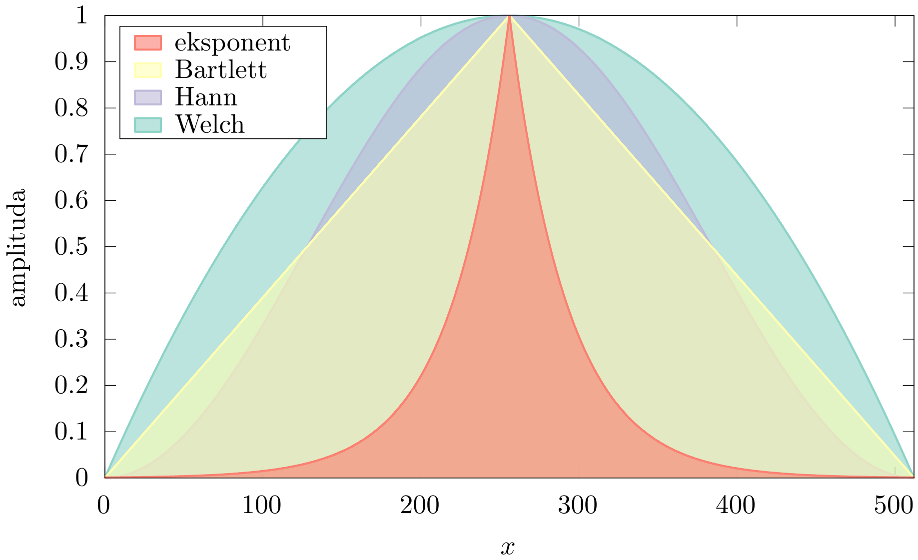
Slika 2.3: Spektri moči (na levi `val2` in na desni `val3` za različne intervale.

Poskusimo obdelati signal še z različnimi okenskimi funkcijami, ki signal na robu postavijo na nič – da dobimo periodičnost signala:

$$\text{Bartlett} = 1 - \left| \frac{j - \frac{N}{2}}{\frac{N}{2}} \right|, \qquad \text{Welch} = 1 - \left(\frac{j - \frac{N}{2}}{\frac{N}{2}} \right)^2, \tag{3}$$

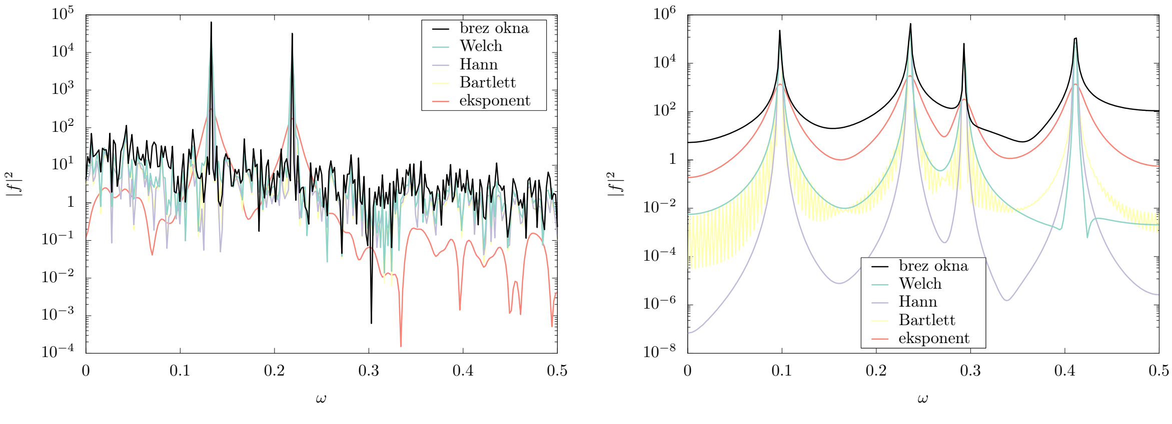
$$\text{Hann} = \sin^2 \frac{\pi j}{N}, \qquad \text{eksponent} = \mathrm{e}^{-|j - \frac{N}{2}| \frac{1}{\tau}}, \tag{4}$$

kjer je N število točk, za parameter τ pa bomo vzeli vrednost $\tau = \frac{N}{2}\frac{8.69}{60}$.



Slika 2.4: Različne okenske funkcije.

Okenske funkcije pomnožimo s signalom v časovnem prostoru in naredimo Fourierovo transformacijo. Na naslednjem grafu 2.5 so podatki obdelani z okenskimi funkcijami. Vidimo, da z okensko funkcijo na prvih podatkih (na sliki levo) spekter le poslabšamo – predvsem z eksponentnim oknom vidimo, da amplitudo precej zmanjšamo in razširimo vrhove. Na drugem spektru (na desni) za eksponentno okno opazimo podobno – znižali smo vrhove in jih razširili. Od ostalih oken se je najbolje izkazala *Hannova* funkcija, ki je vrhove zožala, vmesnemu območju pa se je precej zmanjšala amplituda.



Slika 2.5: Uporaba okenskih funkcij na obeh spektrih.

3 Wienerjev filter

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

$$r(t) = \frac{1}{2\tau} \mathrm{e}^{-|t|/\tau} \quad \tau = 16. \tag{5}$$

3.1 Pristop

Signal $u(t)$, ki prihaja v merilno napravo s prenosno funkcijo $r(t)$, se z dodanim šumom preoblikuje v:

$$c(t) = (u * r)(t) + n(t) = s(t) + n(t), \tag{6}$$

kjer je $s(t)$ izmerjen signal. Če šuma ni, lahko rekonstruiramo vhodni signal:

$$c(t) = (u * r)(t) \quad \xrightarrow{\mathcal{F}} \quad C(\omega) = U(\omega) \cdot R(\omega) \tag{7}$$

$$U(\omega) = \frac{C(\omega)}{R(\omega)} \quad \xrightarrow{\mathcal{F}^{-1}} \quad u(t) = \mathcal{F}^{-1}\left(\frac{C(\omega)}{R(\omega)}\right). \tag{8}$$

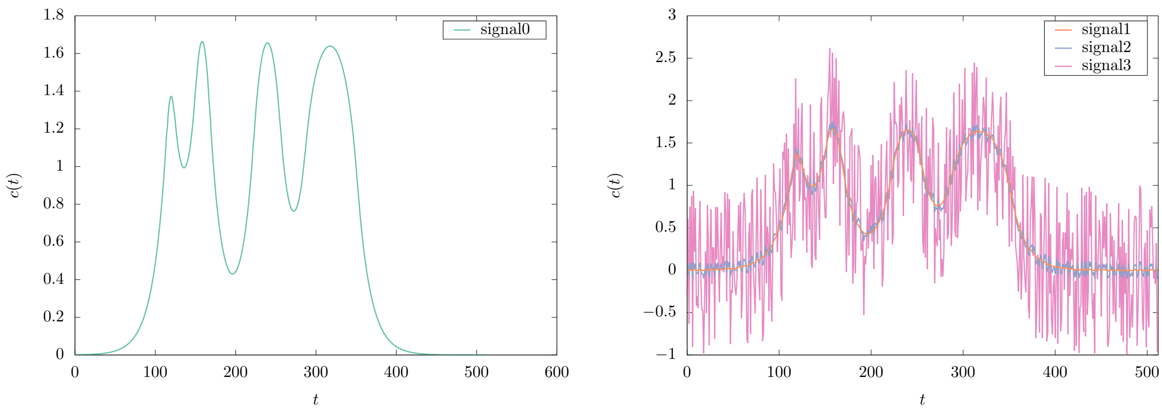
Ko dodamo šum moramo pred dekovolucijo transformiranko $C(\omega)$ pomnožiti z Wienerjevim filtrom:

$$\Phi = \frac{|S(\omega)|^2}{|S(\omega)|^2 + |N(\omega)|^2}, \tag{9}$$

kjer sta $S(\omega)$ in $N(\omega)$ Fourierovi transformiranki $s(t)$ in $n(t)$.

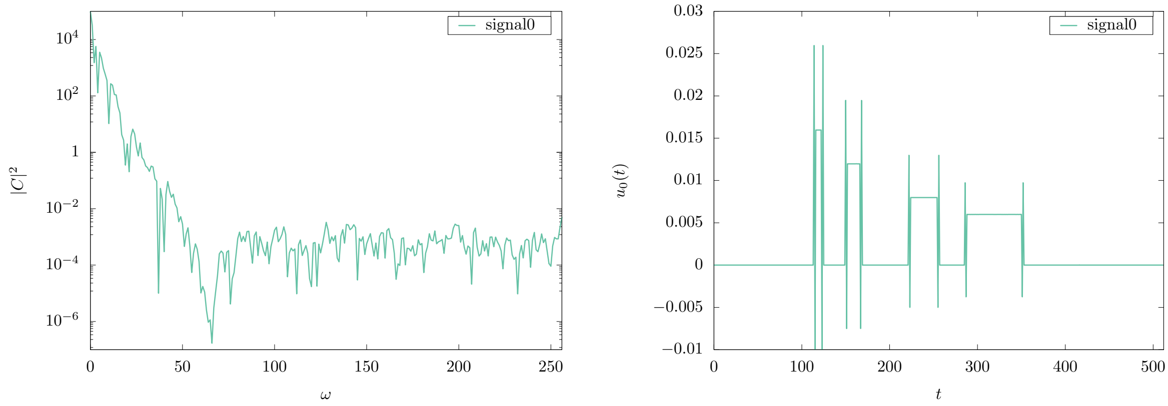
3.2 Reševanje in rezultati

Na spodnjem grafu 3.1 so prikazani vsi štirje signali – vidimo, da je prvi res nezašumljen, ostali trije pa imajo precej šuma.



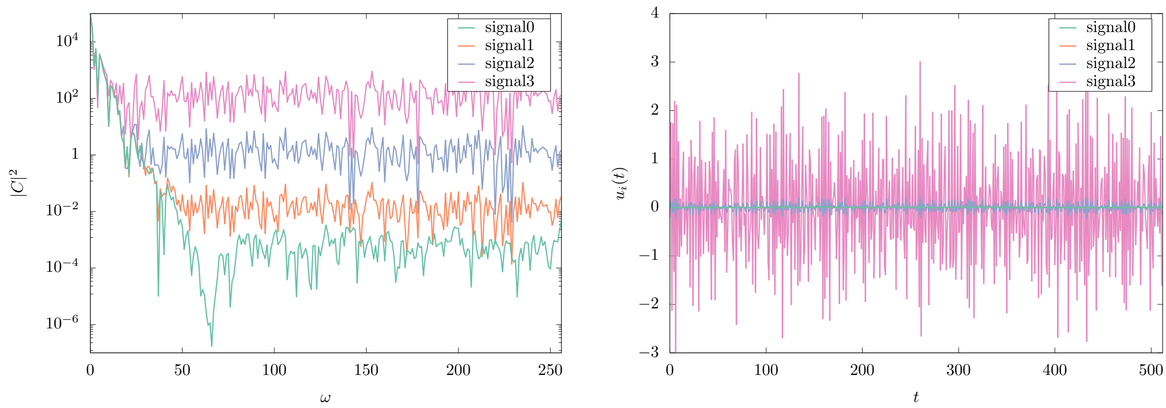
Slika 3.1: Na levi je prikazan prvi signal – c_0, na desni pa ostali trije signali.

Zgoraj omenjeni postopek najprej uporabimo na nezašumljenem signalu `signal0` in poskusimo dobiti iz njega vhodni signal u_0. Na grafu 3.2 levo vidimo njegov frekvenčni spekter, na desni pa rekonstruiran vstopni signal. Ta je sestavljen iz štirih škatlastih signalov – prvi je najvišji in najožji, ostali pa se nato nižajo in širijo.



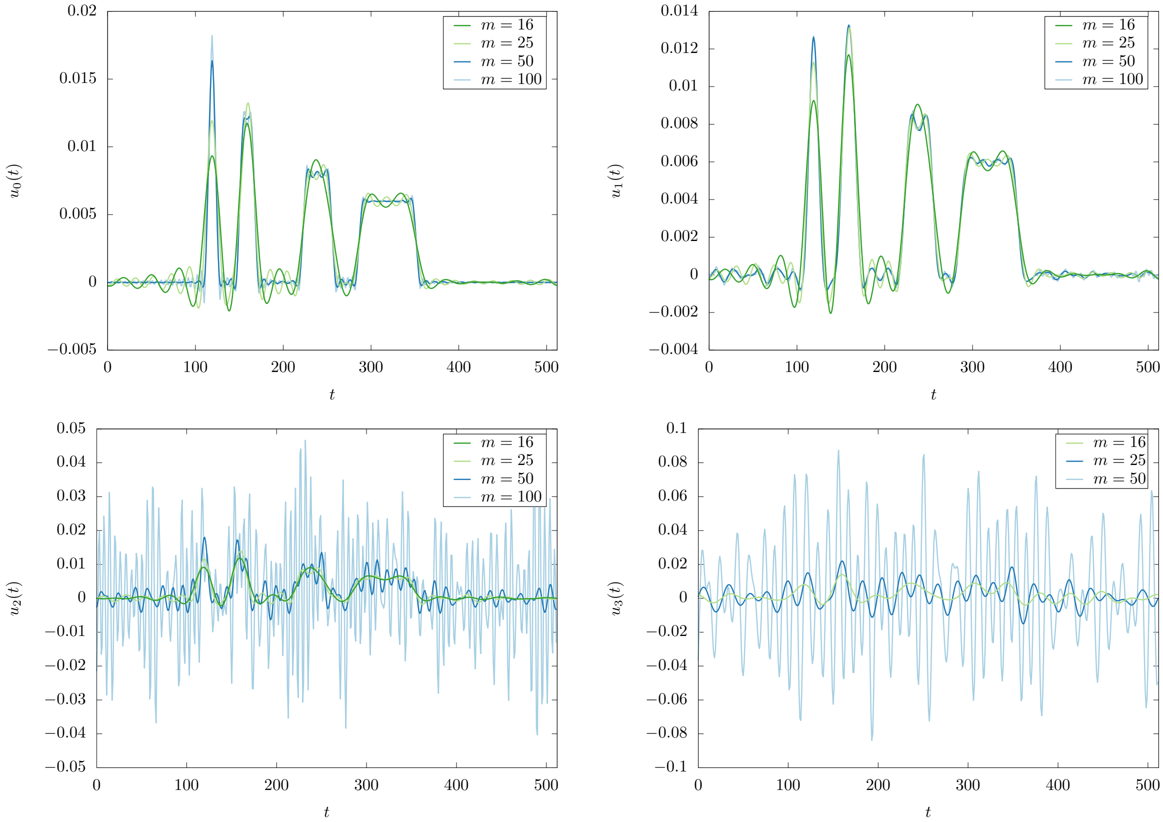
Slika 3.2: Frekvenčni spekter signala c_0 in dekonvuliran vstopni signal $u_0(t)$.

Z Wienerjevim filtrom rekonstruiramo še ostale signale in dobimo graf 3.3. Moči vsakega spektra so vedno višje, dobljen vhodni signal pa kljub uporabi filtra ni podoben nezašumljenem signalu.



Slika 3.3: Spektri in dekonvulirani vstopni signali.

Filter Φ poskusimo izboljšati še tako, da območja, kjer se nam zdi, da dobimo samo šum – na začetku in koncu meritve – postavimo na nič. Tako za različne širine teh intervalov (označene zm) pridelamo grafe 3.4. Vidimo, da s tem čistemu signalu *zaobljimo* obliko zmanjšamo amplitudo stopnic. Pri zašumljenih signalih vidimo, da smo na tak način bolj ali manj uspešno rekonstruirali vstopni signal.



Slika 3.4: Vstopni signali dobljeni z rezanjem filtriranega območja – m je njihova širina.

4 Čiščenje slike

V arhivu `lena_slike.tar.gz` imamo slike podobe Lene, ki so 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 $= 0, 4, 8, 16$). Po filtriranju z uporabo Wienerjevega filtra rekonstruiramo slike. V arhivu so tudi slike z dodano periodično motnjo (`*_nx.pgm`).

4.1 Pristop

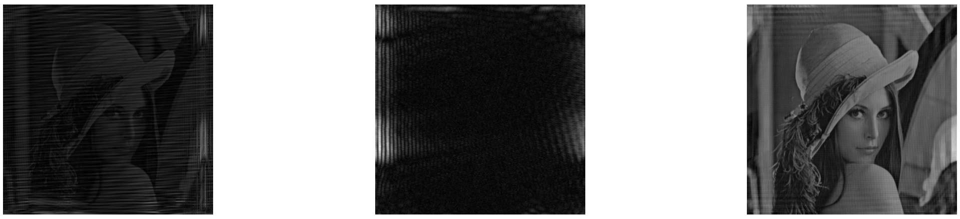
Problema se lotimo na enak način kot v prejšnjem odseku, le da tukaj Wienerjev filter prepišemo v drugo obliko:

$$\Phi(\omega) = \frac{1}{R(\omega)} \frac{|R(\omega)|^2}{|R(\omega)|^2 + \frac{N(\omega)}{S(\omega)}} \tag{10}$$

Sliko po stolpcih preberemo in vsak posamezen stolpec, na enak način kot v prejšnjem delu, rekonstruiramo. Zopet upoštevamo nek *cut off* šuma in poskusimo očistiti sliko. Slike brez dodanega gaussovega šuma lahko kar rekonstruiramo brez uporabe Wienerjevega filtra.

4.2 Dobljene slike

Najprej si poglejmo slike brez dodanega Gaussovega šuma:



Slika 4.1: Očiščene slike brez dodanega šuma – od leve proti desni: `k1-n0`, `k2-n0` in `k3-n0`.

vidimo, da je najlepše očiščena tretja slika, druga slika precej neprepoznavna, prva je sicer zatemnjena, vendar razločimo sliko. Poglejmo si najprej očiščene vse slike s tretjim jedrom.

Slika 4.2: Očiščene slike s tretjim jedrom – v zgosnji vrsti: `k3-n4`, `k3-n8` in v spodnji: `k3-n16` ter `k3-nx`.

Poskusimo očistiti tudi slike s prvim jedrom, vendar že za najmanjši šum pride rešitev precej zatemnjena in nerazpoznavna.

Slika 4.3: Poskus čiščenja slike: `k1-n4`

Druge slike ne uspemo očistiti, poskusimo z različnimi variacijami filtra in širinami frekvenčnih območij, ki jih odrežemo, morda bi morali poskusiti s kakšnim drugačnim jedrom.