Spektralna analiza in filtriranje

10. domača naloga pri predmetu Modelska analiza I

Domen Vaupotič

V tej nalogi se bomo ukvarjali s spektralno analizo in filtriranjem diskretnih signalov. V prvem delu bomo obravnavali diskretno Fourierovo transformacijo in vpliv izbire okenskih funkcij. Ponovimo najprej osnovne definicije. Fourierova transformacija je integralska transformacija:

$$S(f) = \mathcal{F}[s](f) = \int_{-\infty}^{\infty} s(t)\, e^{-i2\pi i f t}\, dt,$$

pri čemer je $s(t)$ signal v časovni domeni, f frekvenca, $\mathcal{F}$ Fourierova transformacija in $S(f)$ Fourierova tranformiranka signala (signal v frekvenčni domeni). Podajmo še inverzno transformacijo:

$$s(t) = \mathcal{F}^{-1}[S](t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(f)\, e^{i2\pi i f t}\, df.$$

V nalogi bomo obravnavali samo *realne signale*, zaradi česar pridobi Fourierova transformiranka Hermitsko simetrijo: $S(f) = \overline{S(-f)}$. Vpeljimo še dvostransko *spektralno gostoto moči* (PSD):

$$\mathcal{P}(f) = |S(f)|^2, \quad f \in \mathbb{R},$$

ta nam pove, koliko "moči" predstavlja posamezna frekvenčna komponenta v signalu. Celotna moč se pri transformaciji ohranja, kar zagotavlja Parsevalova enakost:

$$\int_{-\infty}^{\infty} |s(t)|^2\, dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} |S(f)|^2\, df.$$

Za realne signale je smiselno vpeljati še enostransko spektralno gostoto moči:

$$\mathcal{P}(f) = |S(-f)|^2 + |S(f)|^2, \quad f \in \mathbb{R}^+$$

in nato dvostransko PSD definirati kot $\mathcal{P}(f) = 2\,|S(f)|^2$.

Pri analizi signalov pa nimamo opravka z zveznimi funkcijami, temveč diskretnimi, zato vpeljemo *diskretno Fourierovo transformacijo* (DFT) za signal s sodo dolžino N:

$$S = \mathcal{F}_N[s] \;\rightarrow\; S_k = \sum_{j=0}^{N-1} s_j\, e^{-2\pi i j k/N},$$

$$s = \mathcal{F}_N^{-1}[S] \;\rightarrow\; s_j = \frac{1}{N} \sum_{j=0}^{N-1} S_k\, e^{2\pi i j k/N}.$$

Zapišimo še spektralno gostoto moči za diskretno transformiranko s sodo dolžino N:

$$\mathcal{P}_0 = |S_0|^2,$$

$$\mathcal{P}_k = \frac{1}{2}\left(|S_k|^2 + |S_{N-k}|^2\right), \quad k = 1, 2, \ldots, N/2 - 1,$$

$$\mathcal{P}_{N/2} = |S_{N/2}|^2.$$

Definirajmo najprej ustrezni funkciji za Fourierovo transformacijo in njen inverz.

```
ℱ[s_] := Table[Sum[s[[j + 1]] Exp[(-2 π i j k)/Length@s], {j, 0, Length@s - 1}], {k, 0, Length@s - 1}] // Chop
```

```
ℱℱ[S_] := 1/Length@S Table[Sum[S[[j + 1]] Exp[(2 π i j k)/Length@S], {j, 0, Length@S - 1}], {k, 0, Length@S - 1}] // Chop
```

Preverimo funkciji na enostavnem kratkem zaporedju.

```
ℱ[{1, 0, 3, 5 + 2 i}]
ℱℱ[%]
```

```
{9 + 2 i, -4 + 5 i, -1 - 2 i, -5 i}
```

```
{1, 0, 3, 5 + 2 i}
```

Primerjajmo sedaj rezultate z vgrajenima funkcijama **Fourier** in **InverseFourier**.

```
Fourier[{1, 0, 3, 5 + 2 i}]
InverseFourier[%] // Chop
```

```
{4.5 + 1. i, 0. - 2.5 i, -0.5 - 1. i, -2. + 2.5 i}
```

```
{1., 0, 3., 5. + 2. i}
```

Opazimo, da dobimo drugačno Fourierovo transformiranko, saj *Mathematica* razdeli predfaktor $1/N$ na obe smeri transformacije (na vsako predfaktor $1/\sqrt{N}$). Da spremenimo definicijo, moramo spremeniti nastavitve.

```
SetOptions[{Fourier, InverseFourier}, FourierParameters → {1, -1}];
Fourier[{1, 0, 3, 5 + 2 i}]
InverseFourier[%] // Chop
```

```
{9. + 2. i, -4. + 5. i, -1. - 2. i, 0. - 5. i}
```

```
{1., 0, 3., 5. + 2. i}
```

Preverimo, da za realne signale zares velja Hermitska simetrija $S_k = \overline{S_{N-k}}$:

```
Fourier[{1, 1, 2, 0, 1, 2}] // Chop
```

```
{7., 1., -2. + 1.73205 i, 1., -2. - 1.73205 i, 1.}
```

Sedaj definirajmo še spektralno gostoto moči in jo izračunajmo za nek realen signal.

```
𝒫[S_] := {Abs[S[[1]]]^2, Table[1/2 (Abs[S[[k + 1]]]^2 + Abs[S[[Length@S - k + 1]]]^2), {k, 1, Length@S / 2 - 1}],
   Abs[S[[Length@S / 2 + 1]]]^2} // Flatten
```

```
𝒫[ℱ[{1, 0, 3, 5, 2, 1}]]
```

```
{144, 36, 12, 0}
```

Enak rezultat dobimo tudi z vgrajeno funkcijo **PeriodogramArray**, ki ji nastavimo ustrezne predfaktorje. Funkcija sicer deluje za kompleksne signale, zato vrne dvostransko PSD, vendar takoj vidimo simetrijo.

```
PeriodogramArray[{1, 0, 3, 5, 2, 1}, FourierParameters → {1, -1}]
```

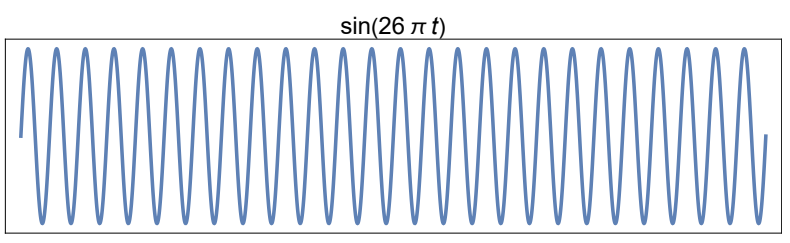
```
{144., 36., 12., 0., 12., 36.}
```

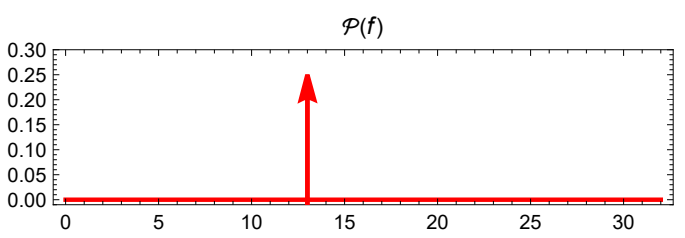
Oglejmo si sedaj najbolj osnovne primere DFT in najpogostejše težave, ki se pri tem pojavijo. Zamislimo si, da merimo harmonični pojav, ki daje sinusni signal $\sin(13*2\pi t)$, torej signal s frekvenco $f = 32$. Izračunajmo Fourierovo transformacijo:

```
FourierTransform[Sin[13 × 2 π t], t, f, FourierParameters → {0, -2 * Pi}]
```

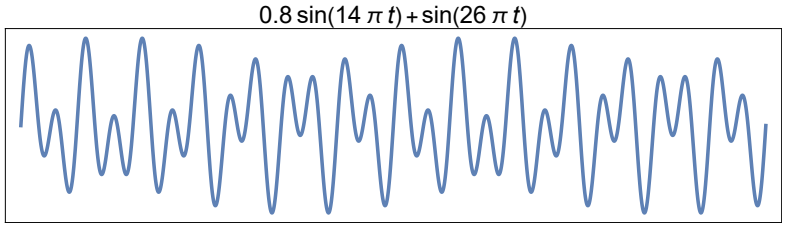
$$-\frac{1}{2}\, i\, \text{DiracDelta}[-13 + f] + \frac{1}{2}\, i\, \text{DiracDelta}[13 + f]$$

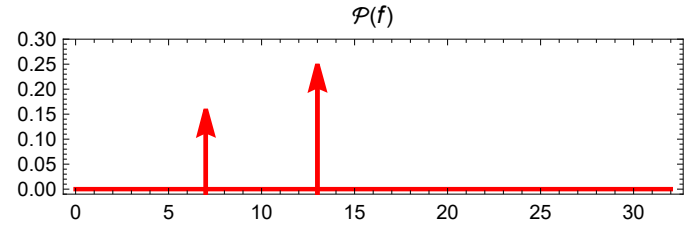
Spektralna gostota moči bo torej skoncentrirana pri frekvenci 13.





Podobno lahko izrišemo spektralno gostoto moči za kombinacijo sinusnih signalov.

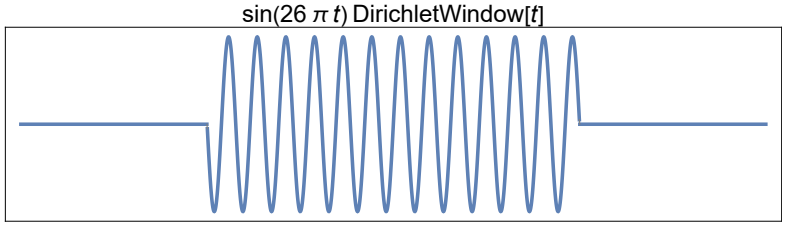


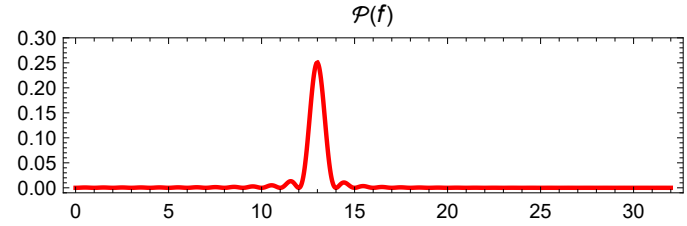


Sedaj si zamišljamo, da smo signal vzorčili *zvezno* na intervalu $t \in [-1/2, 1/2]$, s čimer smo zajeli polnih 13 period. Takšen signal lahko tvorimo tako, da izvorni signal pomnožimo s *kvadratno* (Dirichletovo) *okensko funkcijo*.

```
FourierTransform[Sin[13 * 2 π t] DirichletWindow[t], t, f, FourierParameters → {0, -2 * Pi}]
```

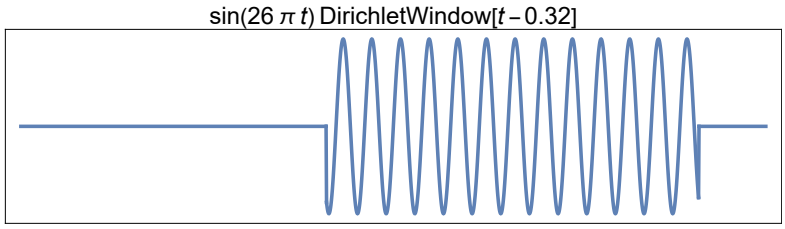
$$\frac{13\, i\, \mathrm{Sin}[f\, \pi]}{\left(-169 + f^2\right) \pi}$$

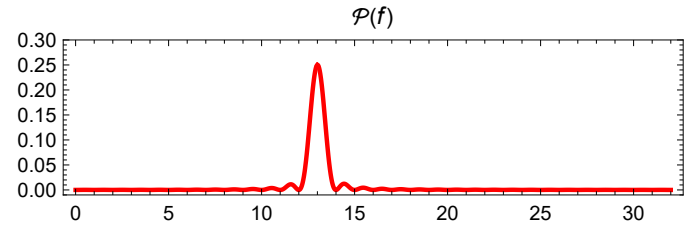




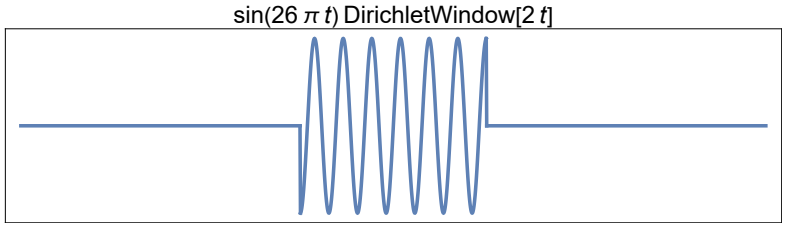
Dobili smo funkcijo, ki vsekakor *ni* periodična na celotnem definicijskem intervalu. Posledično dobimo Fourierovo transformacijo, v kateri zaznamo *spektralno puščanje* (spectral leakage). Gostota moči ni več zbrana pri eni frekvenci, temveč dobimo tudi vrhove pri drugih frekvencah. Množenje v časovni domeni je ekvivalentno konvoluciji v frekvenčni domeni, Fourierova transformacija Dirichletove okenske funkcije pa je funkcija sinc(x). Njen kvadrat torej vidimo v dobljeni spektralni gostoti moči.

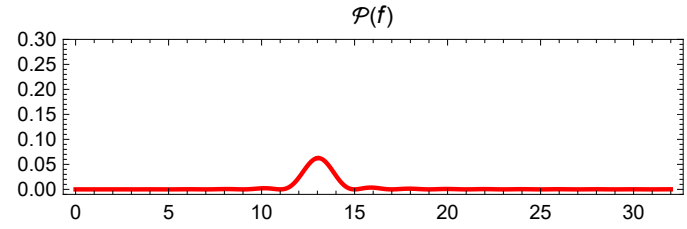
Okensko funkcijo lahko premaknemo in s tem ne vplivamo na spektralno gostoto moči.



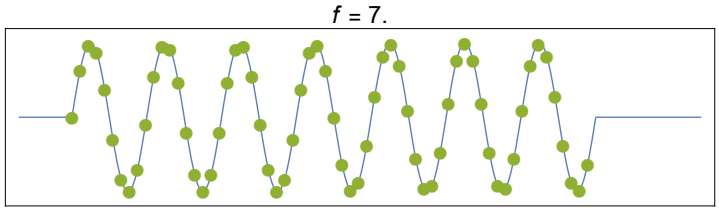


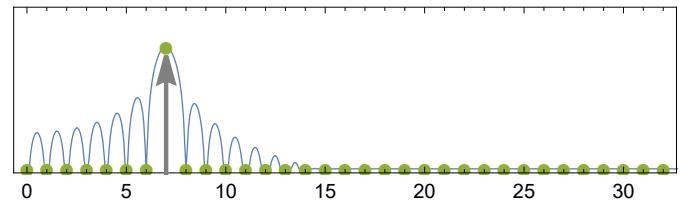
Če zajamemo krajši signal, pa dobimo manjšo spektralno gostoto in s tem slabšo frekvenčno ločljivost.

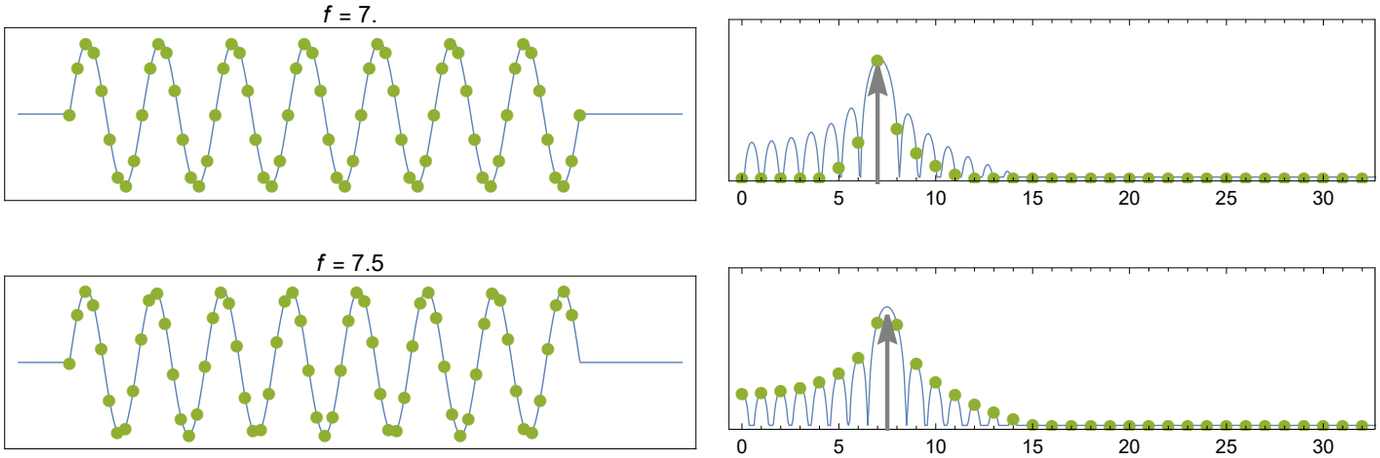




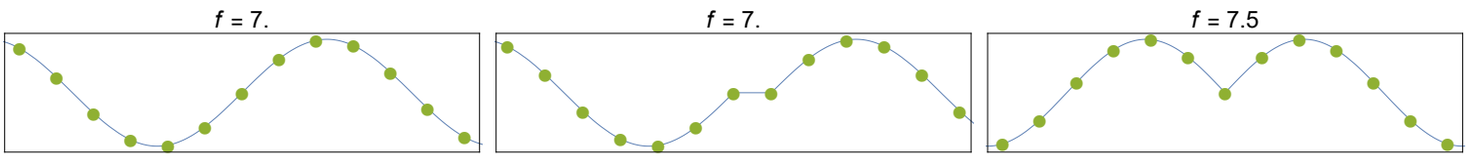
Sedaj se prestavimo v realni eksperimentalni svet in s tem v *diskretno vzorčenje*. Zamislimo si, da smo zgornji sinusni signal merili v enakomernih časovnih presledkih, pri čemer smo dobili 64 vzorcev (merilnih točk). Spodaj so prikazani trije različni primeri.



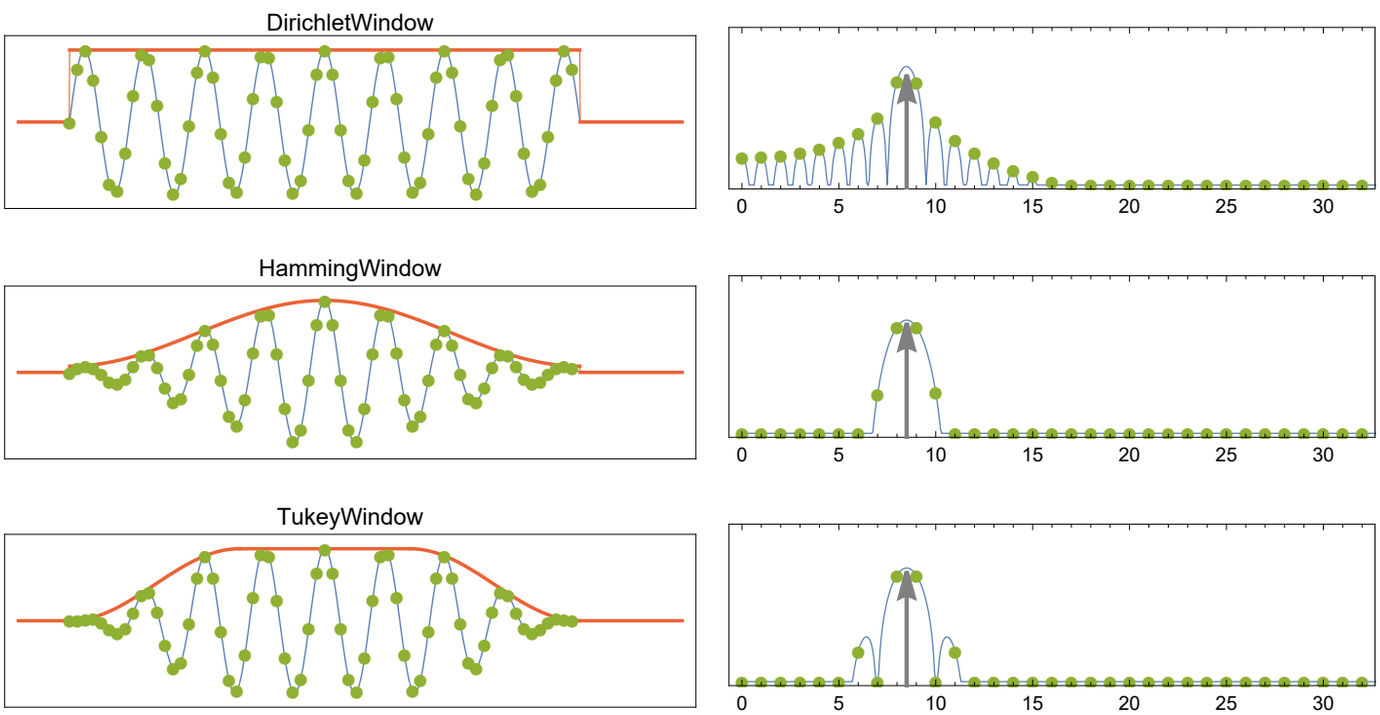




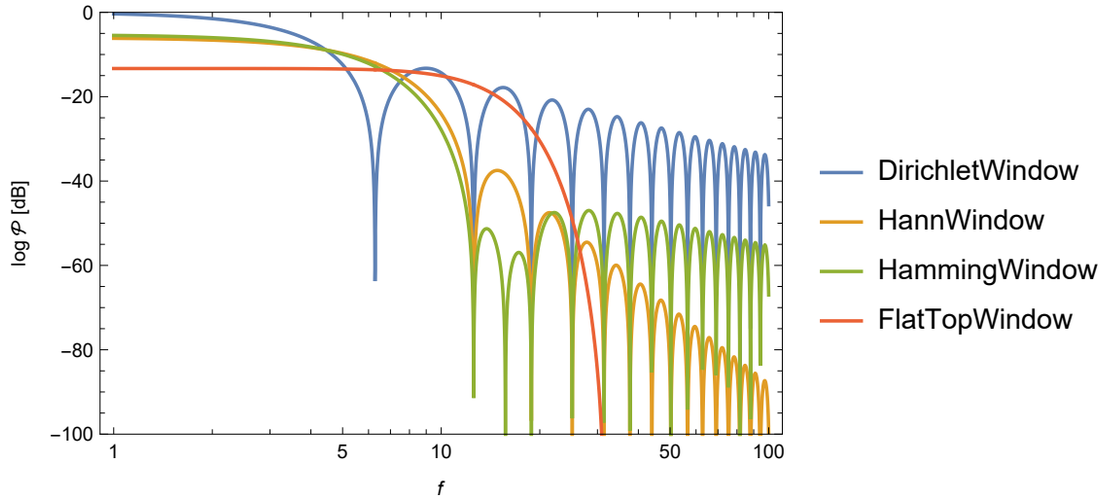
Opazimo lahko, da smo le v prvem primeru, ko smo vzorčili v takih časovnih presledkih, da se ponavljanje signala lahko enakomerno nadaljuje, dobili “točno” spektralno gostoto. V obeh drugih primerih je prisotno spektralno prelivanje. Da bomo bolje razumeli, kako pride do prelivanja, si vsakega izmed zgornjih primerov oglejmo tako, da zlepimo skupaj dve kopiji vzorca.



Opazimo ključni problem: le v prvem primeru se dve kopiji vzorca elegantno zlepita v sinusno krivuljo. S tem pa se pojavi tudi pot do rešitve. Signal moramo predobdelati tako, da se bodo njegove kopije lepo zlepljale druga z drugo. To dosežemo tako, da izberemo okensko funkcijo, ki nima tako ostrega prehoda kot Dirichletova, temveč na robovih počasi pojema k nič.

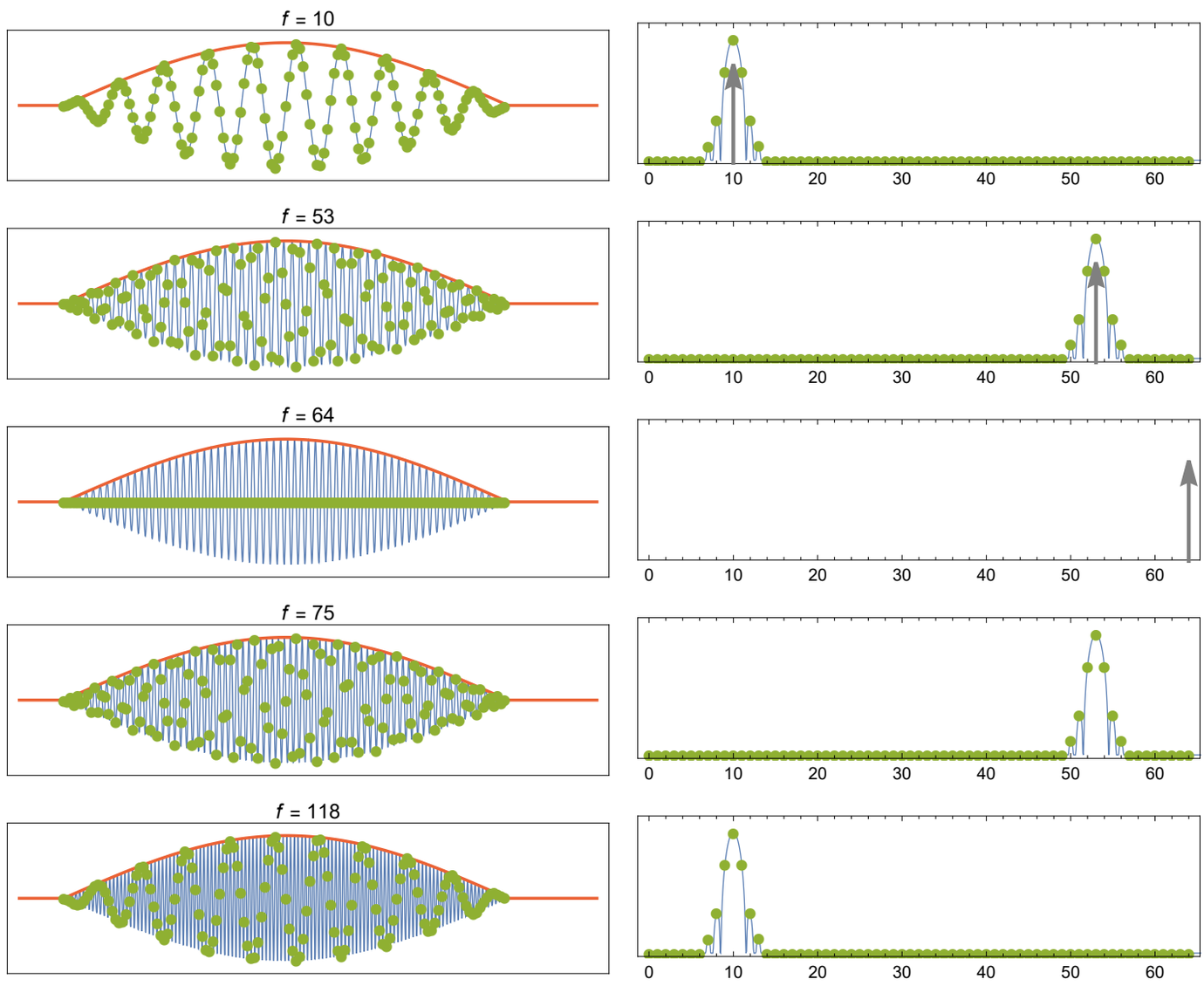


Opazimo zanimivo obnašanje! Obe izbrani okni sta sicer odpravili spektralno puščanje, vendar sta razširili spektralni vrh. Izbira okenske funkcije je torej zmeraj *kompromis* med želenim zmanjšanjem spektralnega prelivanja in neželenim razširjanjem vrha. Kadar vemo, da so v signalu vrhovi med seboj dobro ločeni in želimo čim bolje določiti njihovo amplitudo, tedaj uporabimo okensko funkcijo, ki ima sicer širok vrh, a majhno spektralno prelivanje (torej energija/amplituda enega vrha se ne bo “prenašala” v drug vrh), npr. okno *FlatTop*. Kadar pa želimo v signalu ločiti dva bližnja vrha, podobna po amplitudi, tedaj bomo uporabili npr. kar *Dirichletovo* okno.



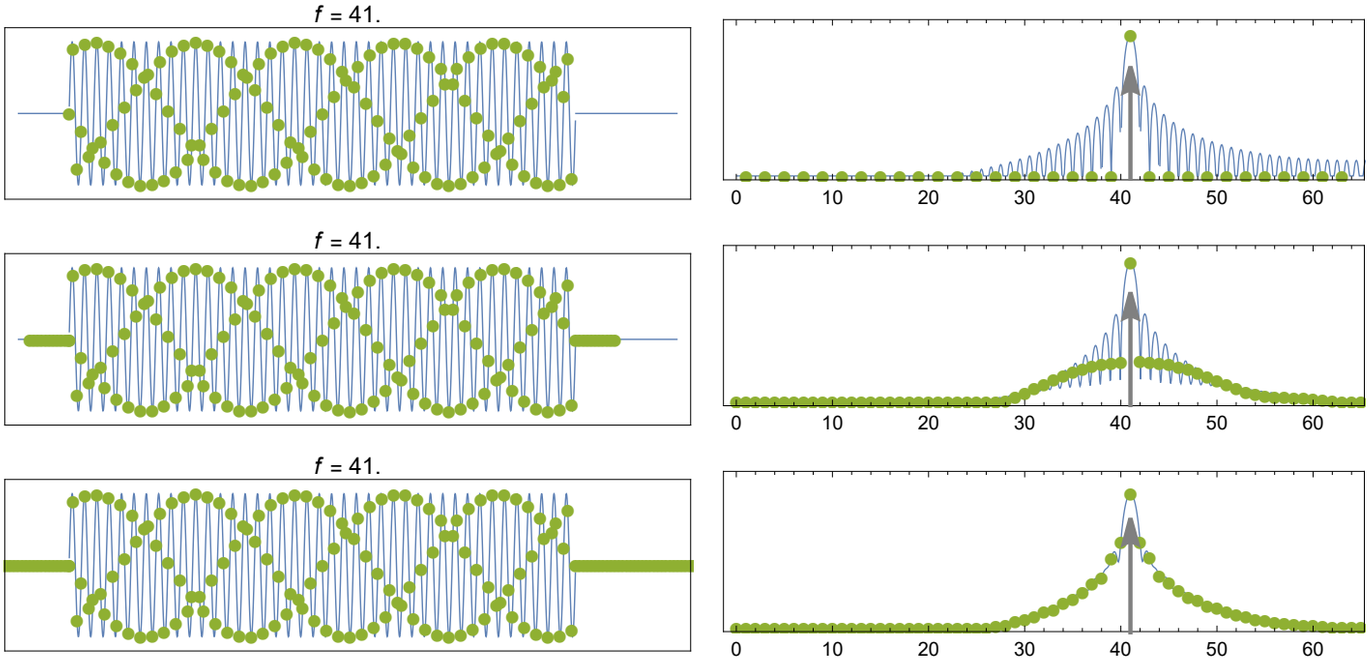
Izbira kompromisa se odlično vidi na zgornjem grafu, kjer je prikazana spektralna gostota pogostih okenskih funkcij: ožja kot je širina glavnega vrha, večje je spektralno prelivanje.

Oglejmo si sedaj še drug zanimiv pojav, ki pa zares nastane le pri diskretnem vzorčenju in ga ne moremo poustvariti v zvezni sliki.



Opazimo, da s povečevanjem nosilne frekvence signala naletimo na limito – najvišjo možno frekvenco, ki jo z dano velikostjo vzorca še lahko določimo. To frekvenco imenujemo *Nyquistova frekvenca* in je enaka polovični velikosti vzorca. V zgornjem primeru je v vzorcu 128 izmerkov, torej znača Nyquistova frekvenca 64. Če frekvenco povečujemo, se vrh v PSD “odbije” od Nyquistove frekvence in začne potovati nazaj – predstavljati se začne kot manjša frekvenca, od tod ime *potujitev*.

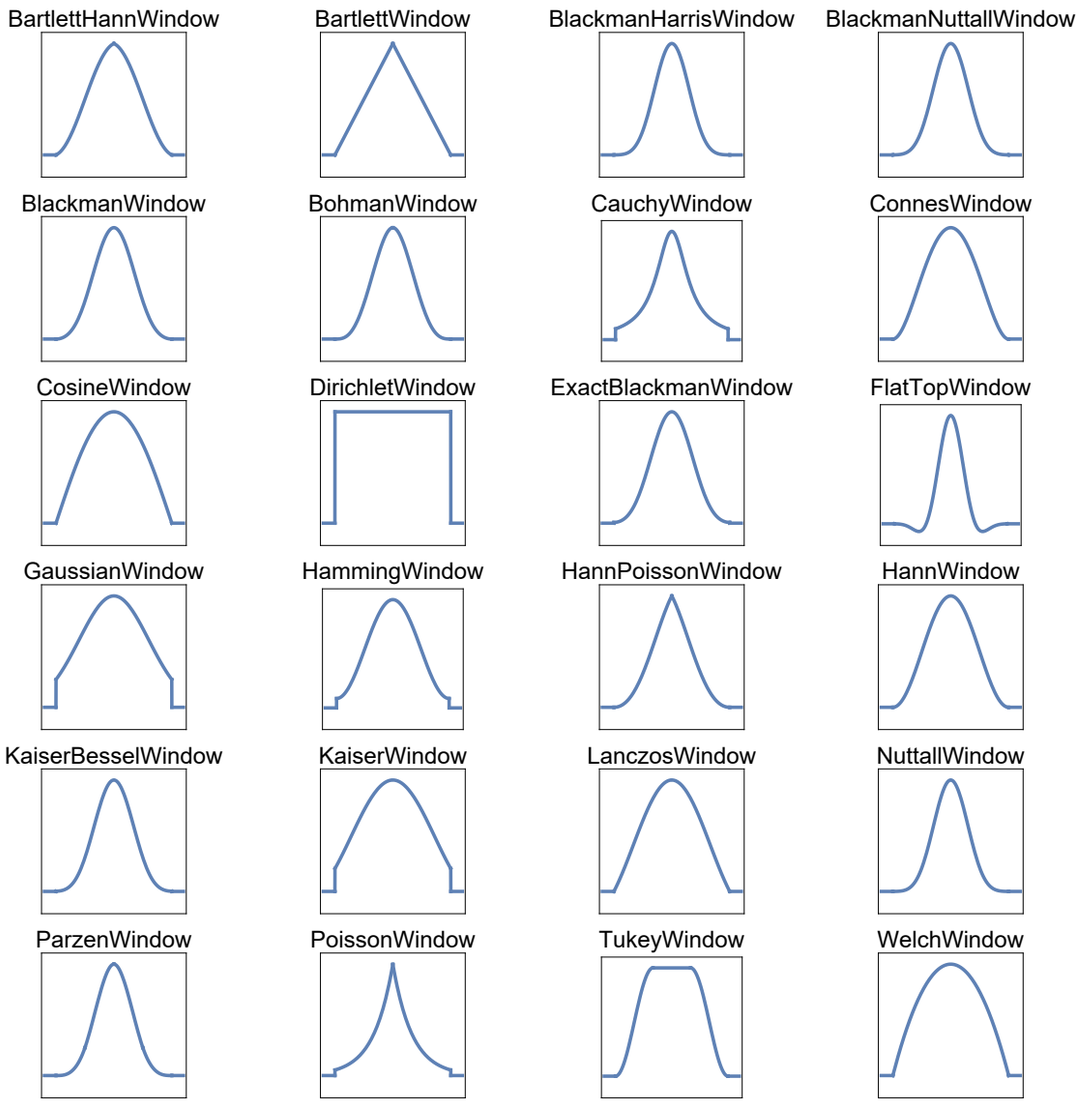
Oglejmo si še, kaj se dogaja s spektrom, če na obe strani vzorca dodamo ničle (*zero padding*).



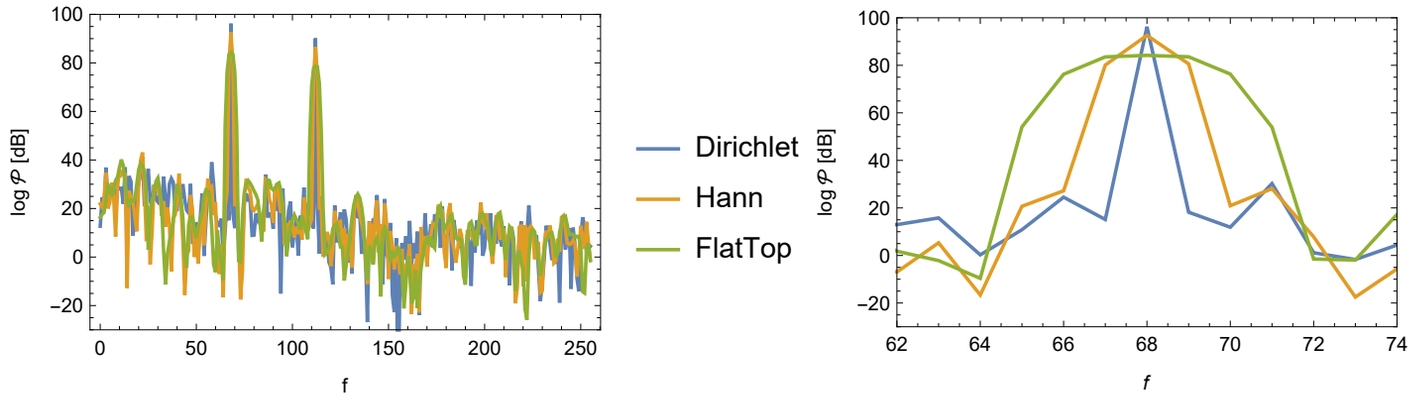
Opazimo, da z dodajanjem ničel sicer pridelamo več točk v frekvenčnem spektru (lahko bi rekli, da dobimo boljšo interpolacijo), vendar s tem ne povečamo frekvenčne ločljivosti, saj se vrh začne kvečjemu razširjati.

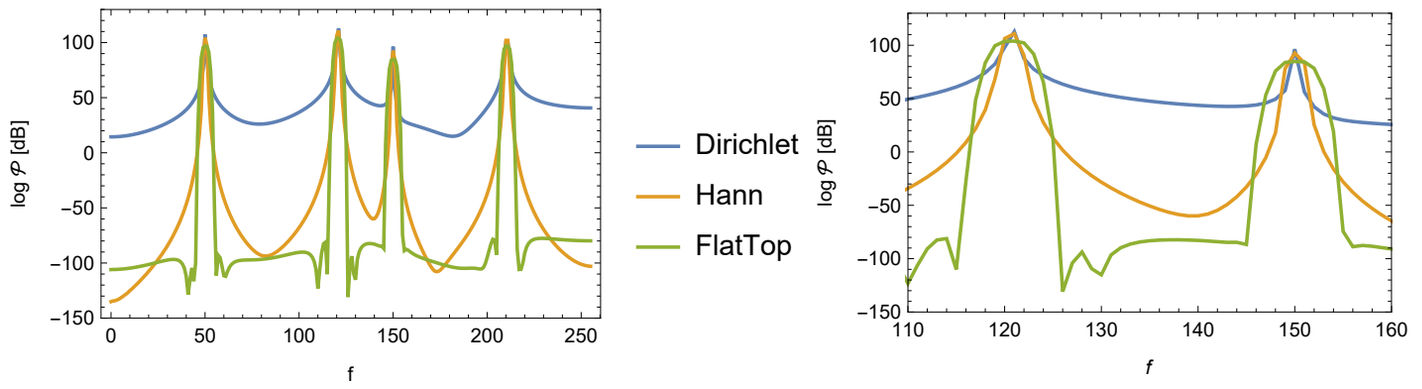
1. Frekvenčni spekter

Najprej si oglejmo nabor različnih oken, ki nam jih ponuja *Mathematica*. Opazimo, da je večina okenskih funkcij pozitivnih. Nekatera izmed oken imajo še dodatne parametre, s katerimi spreminjamo njihovo obliko.



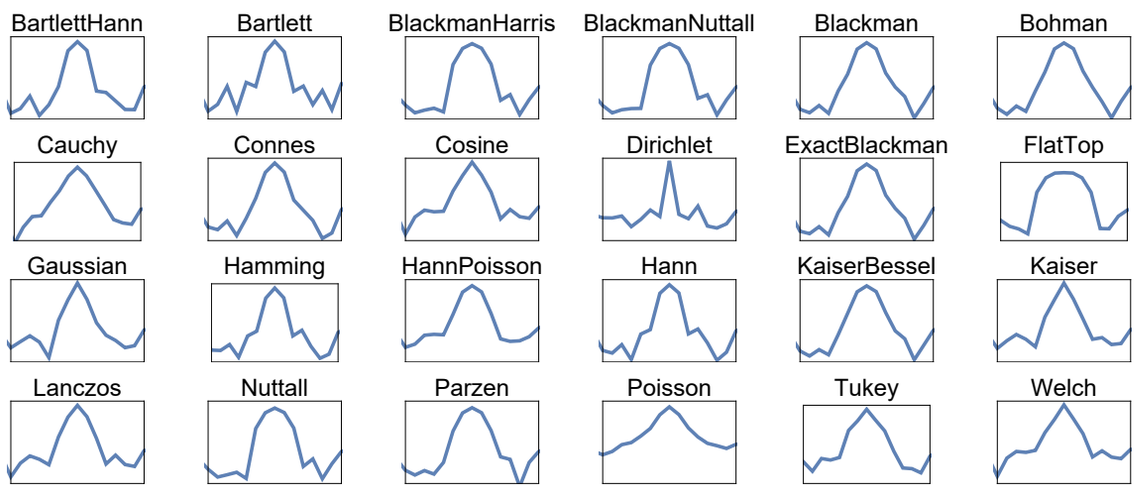
Omejimo nabor okenskih funkcij, ki jih bomo obravnavali. Vzemimo okno, s katerim lahko natančno določimo amplitude, a ima slabšo frekvenčno ločljivost (FlatTop), okno z zelo veliko frekvenčno ločljivostjo (Dirichlet) in okno z vmesnimi lastnostmi (Hann). Oglejmo si spekter moči za prvi signal naloge, ki je prikazan na spodnjih grafih.

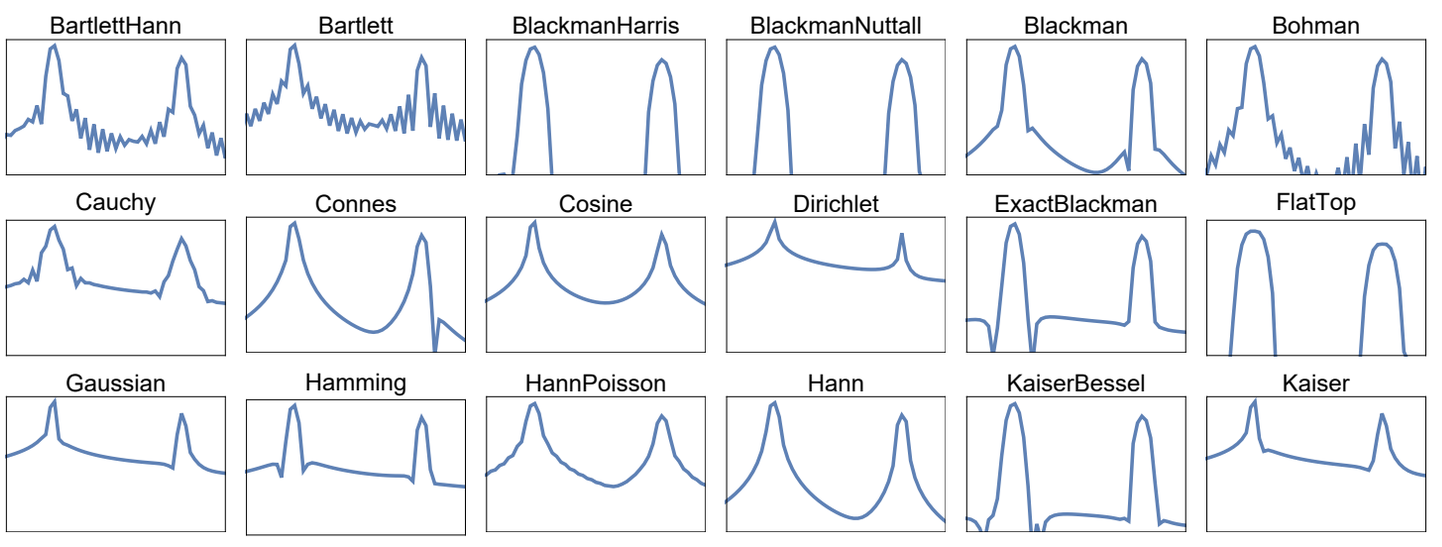


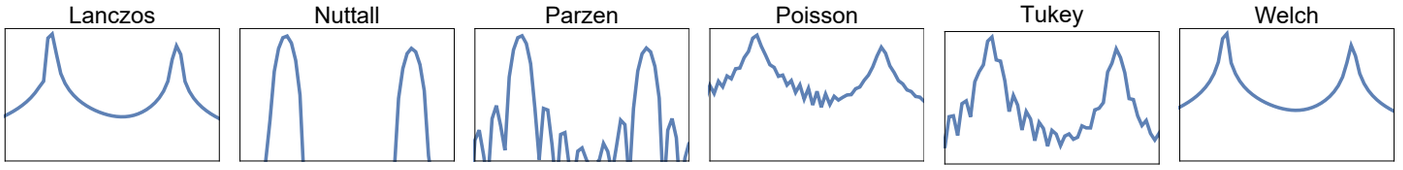


Zelo lepo se opazi omenjeni kompromis. Okno FlatTop ima najmanjše prelivanje, vendar znatno razširi vrh (in ga tudi zmanjša, kar imenujemo *scalloping loss*). Dirichletovo okno da najožji in najvišji vrh, vendar ima znatno prelivanje.

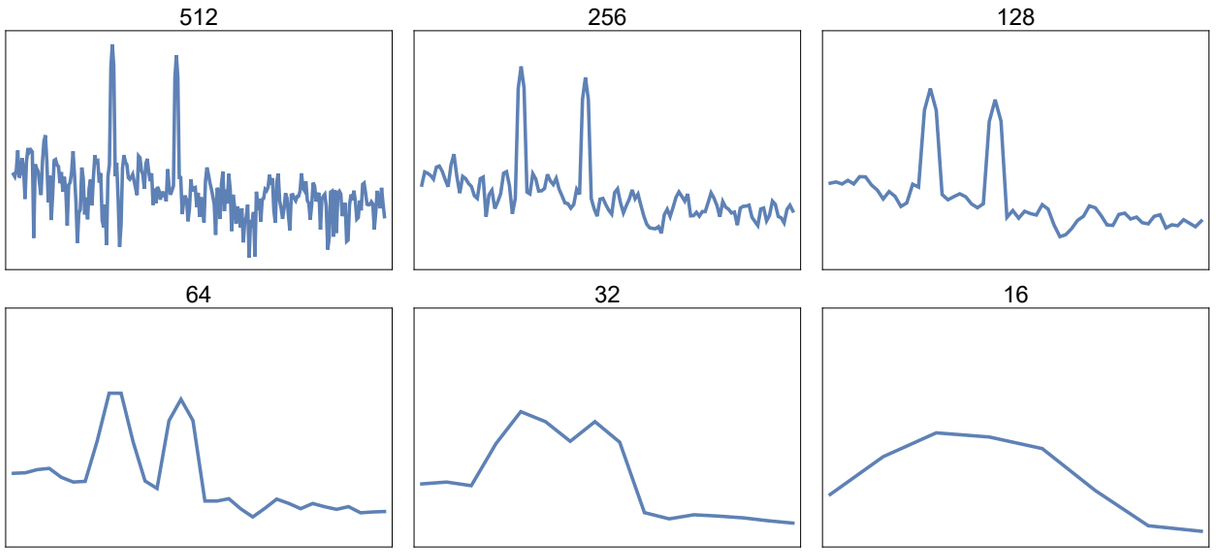
Na spodnjih dveh slikah so prikazana še ostala okna. Glede na naše potrebe pri obdelovanju signala se torej vsakič znova odločimo o ustreznem oknu, s katerim bomo dobili najboljše rezultate.







Če analiziramo krajše signale in rezultate povprečimo (Welchova metoda), dobivamo slabšo frekvenčno ločljivost.



2. Wienerjev filter

V drugem delu naloge se bomo ukvarjali s filtriranjem signalov. Definirajmo najprej konvolucijo dveh funkcij:

$$(u * r)(t) = \int_{-\infty}^{\infty} u(\tau)\, r(t-\tau)\, d\tau.$$

Sedaj si zamislimo, da vhodni signal $u(t)$ potuje čez napravo, ki ima prenosno funkcijo $r(t)$, pri čemer pa dodatno nastaja še šum $n(t)$. Izhodni signal, $c(t)$, bo torej oblike:

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

pri čemer smo s $s(t)$ označili konvoluirani signal. Prestavimo se sedaj v frekvenčno domeno:

$$C(f) = U(f)\, R(f) + N(f) = S(f) + N(f).$$

Najti želimo tako filtrirno funkcijo $\Phi(f)$, da bomo dobili čim bolje rekonstruirani signal $\tilde{U}(f) = [C(f)/R(f)]\, \Phi(f)$. Napaka, ki jo želimo minimizirati, je vsota kvadratov:

$$\epsilon = \int_{-\infty}^{\infty} \left| \tilde{U}(t) - U(t) \right|^2 dt \overset{\text{Plancherelov izrek}}{=} \int_{-\infty}^{\infty} \left| \tilde{U}(f) - U(f) \right|^2 df \overset{!}{=} \min.$$

Pišimo $\mathbb{E}[\bullet] = \int_{-\infty}^{\infty} \bullet\, df$.

$$\epsilon = \mathbb{E}\left[\left| \tilde{U}(f) - U(f) \right|^2 \right] = \mathbb{E}\left[\left| \frac{S+N}{R}\Phi - \frac{S}{R} \right|^2 \right] = \mathbb{E}\left[\left| \frac{1}{R} \right|^2 \left(|S|^2 \, |\Phi - 1|^2 + |N|^2 \, |\Phi|^2 + \text{mešani člen} \right) \right].$$

Mešani člen oblike $S\,N$ je enak nič, če *predpostavimo*, da sta signal in šum med seboj nekorelirana. Dobljeni funkcional variiramo in izrazimo iskani filter:

$$\delta\epsilon = 2 \left| \frac{S}{R} \right|^2 |\Phi - 1|^2\, \delta\Phi + 2 \left| \frac{N}{R} \right|^2 \Phi\, \delta\Phi \overset{!}{=} 0,$$

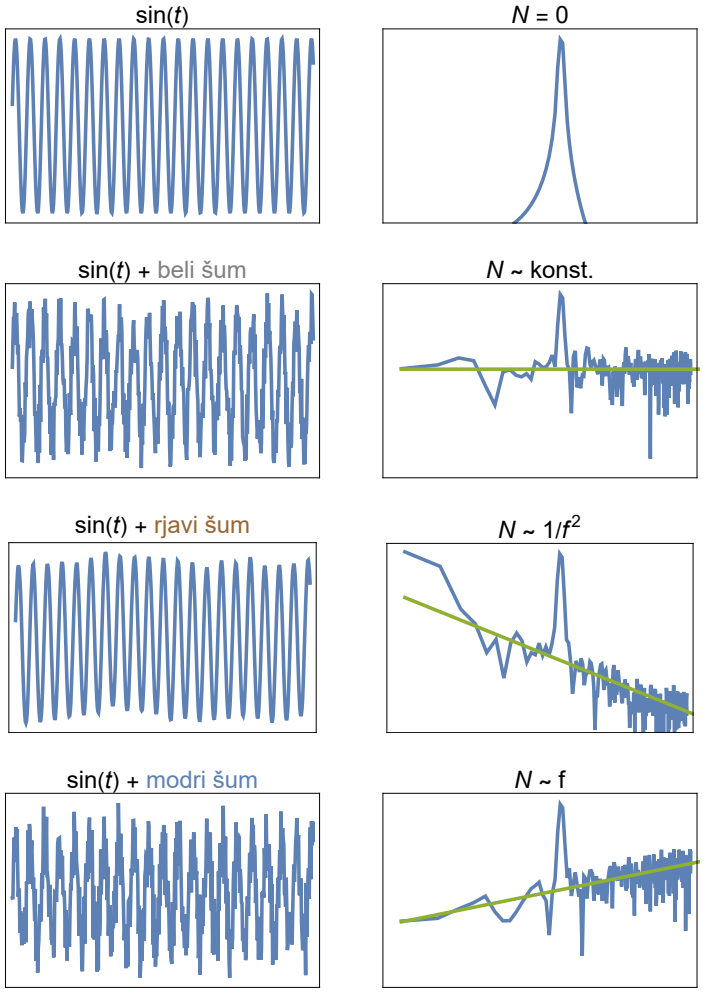
$$\Phi(f) = \frac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2}.$$

Čeprav smo dobili eksplicitni izraz za filtrirno funkcijo, imamo težavo: poznamo namreč le izhodni signal $C(f)$, ne pa tudi signala brez šuma ali samega šuma. Zato moramo ti dve funkciji *oceniti* iz spektralne gostote moči merjenega signala $C(f)$.

$$|S(f)|^2 + |N(f)|^2 \approx P_c(f) = |C(f)|^2.$$

Pri ocenjevanju modela za šum nam ni treba biti posebej natančni, saj bo napaka drugega reda.

Oglejmo si za začetek, kaj se zgodi s spektrom, ko sinusnemu signalu dodamo nek tip šuma. Prikazani so trije tipi šuma: Gaussov (beli) šum, rjavi šum (s frekvenčno odvisnostjo $1/f^2$) in modri šum (s frekvenčno odvisnos-tjo f).



V nalogi imamo podano eksponentno prenosno funkcijo:

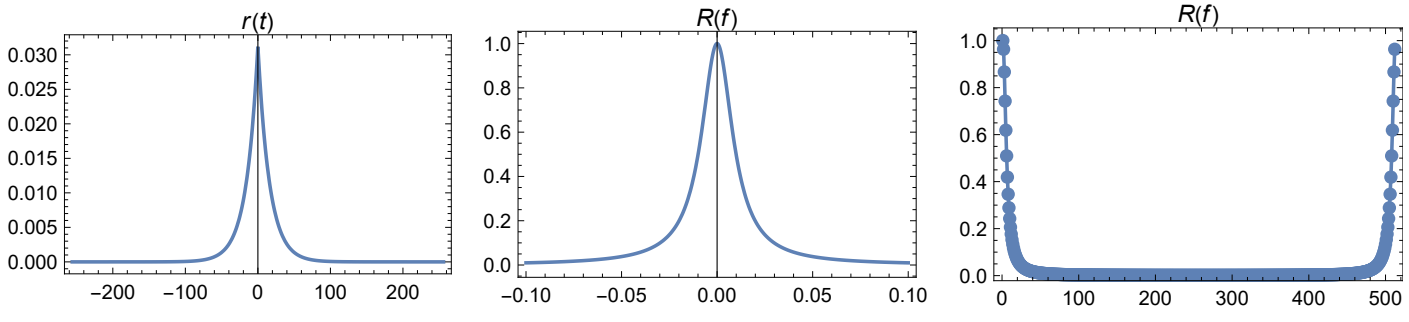
$$r(t) = \frac{1}{2\tau} e^{-|t|/\tau}, \quad \tau = 16.$$

Izračunajmo najprej njeno Fourierovo transformiranko:

```
FourierTransform[1/(2 τ) Exp[-Abs[t] / τ], t, f, FourierParameters → {0, -2 * Pi}]
```

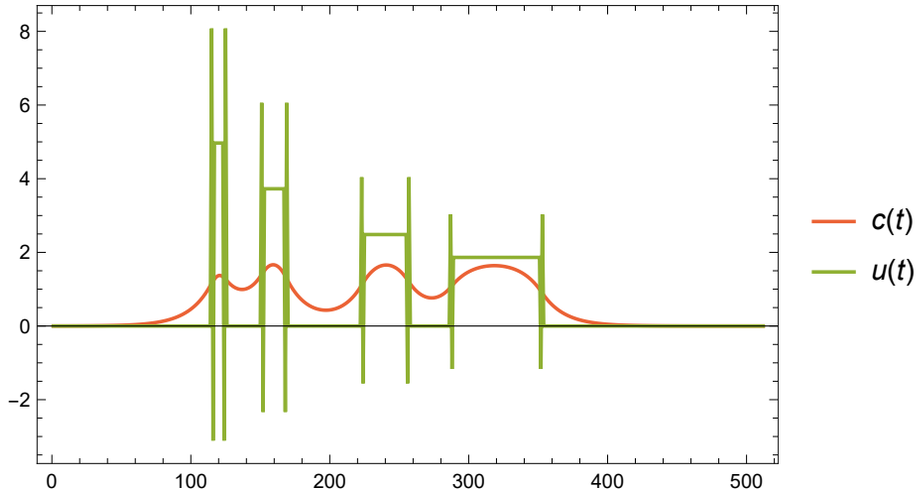
$$\frac{1}{1 + 4 f^2 \pi^2 \tau^2}$$

Pozorni smo, da pri diskretizaciji prenosno funkcijo ustrezno ovijemo.

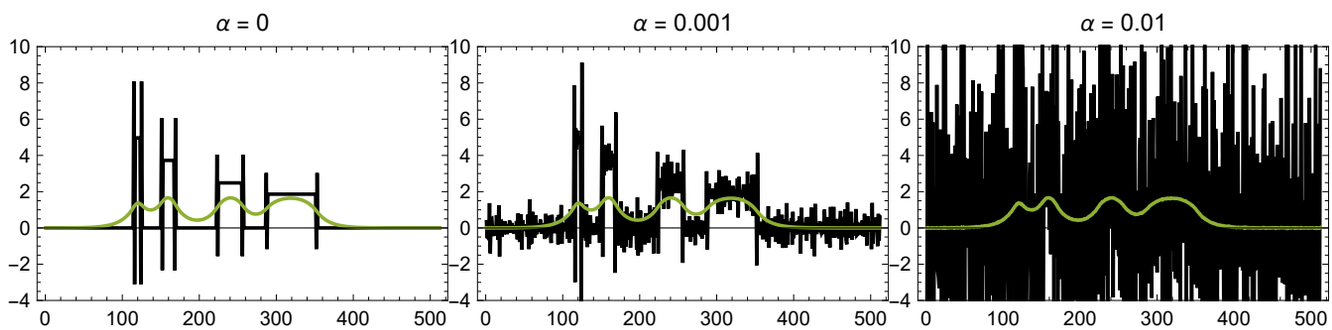


Oglejmo si sedaj najprej signal brez šuma. V tem primeru lahko dekonvolucijo direktno izvedemo brez težav:

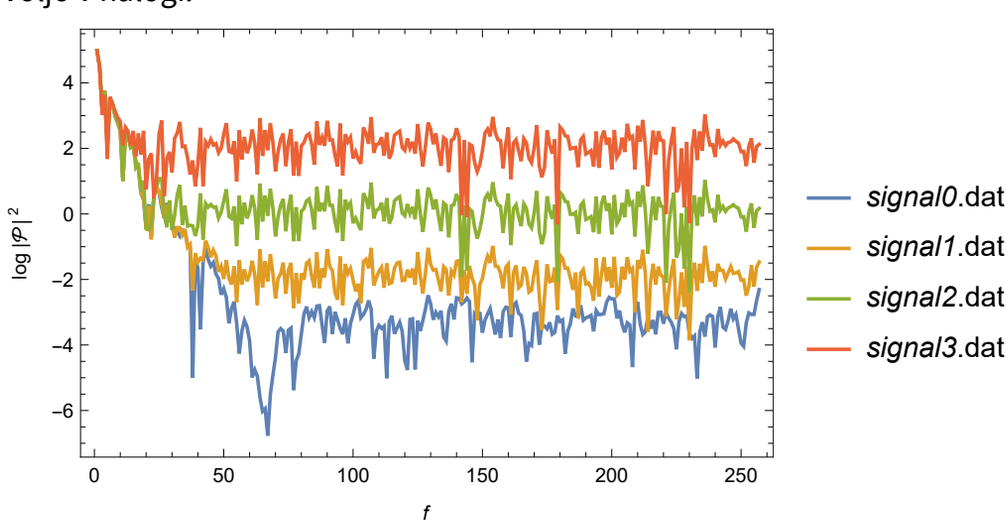
$$u(t) = \mathcal{F}^{-1}\left[\frac{C(f)}{R(f)}\right]$$



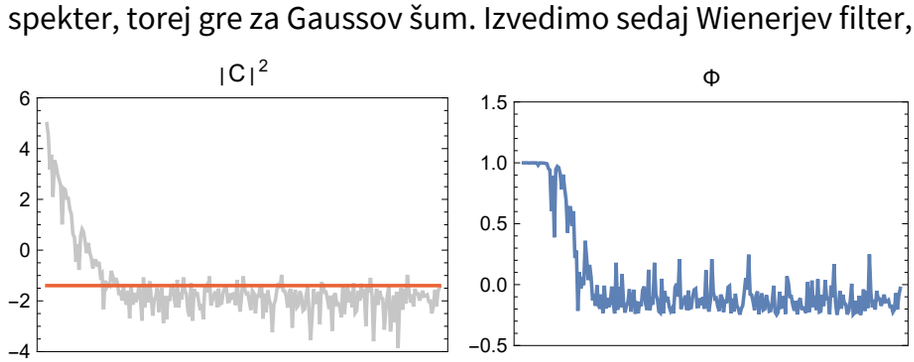
Naš signal torej predstavljajo štirje škatlasti sunki. Kaj se zgodi, če na tak čisti signal umetno dodamo Gaussovski šum z amplitudo α?

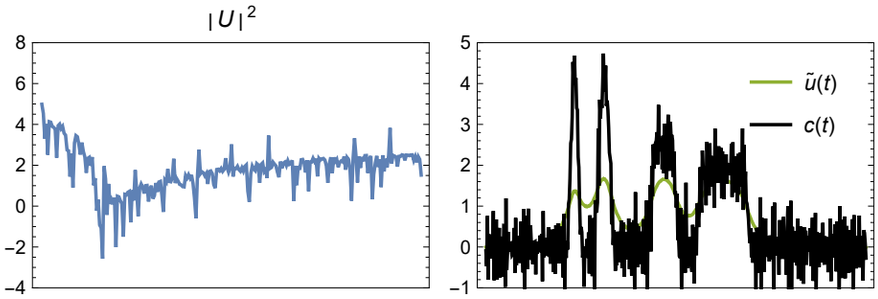


Čeprav smo signalu dodali komaj zaznaven šum (z očesom praktično neviden), postane dekonvolucija povsem neuporabna. Deljenje na območju, kjer je prenosna funkcija zelo majhna, močno ojači tiste frekvence – to pa je pogosto ravno pri visokih frekvencah, ki predstavljajo šum. Zaradi tega torej v splošnem pride do močne ojačitve šuma. Oglejmo si sedaj, kako izgleda spektralna gostota moči pri vseh štirih signalih, ki jih imamo na voljo v nalogi.

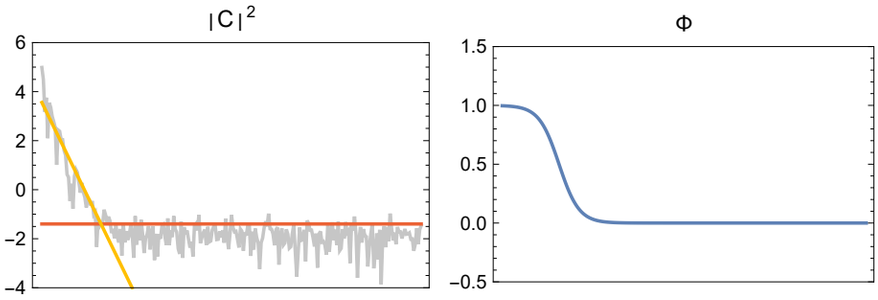


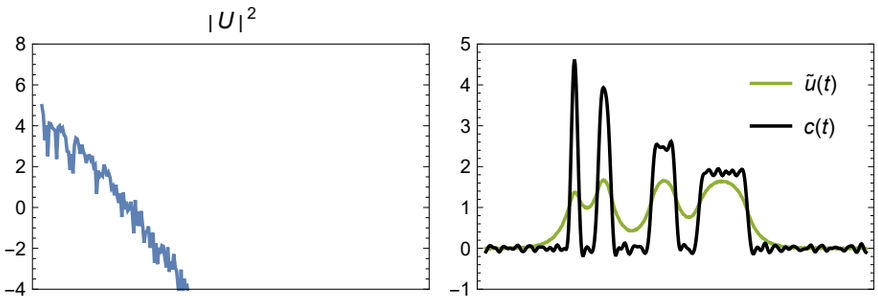
V spektru sta jasno vidni dve območji: območje signala in območje šuma. Videti je, da ima šum konstanten spekter, torej gre za Gaussov šum. Izvedimo sedaj Wienerjev filter, tako da na šum prilegamo premico.



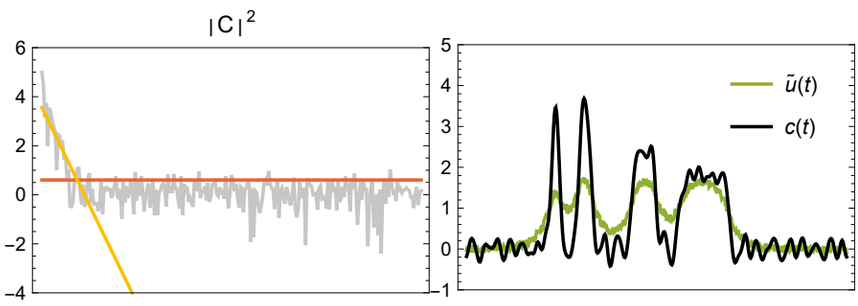


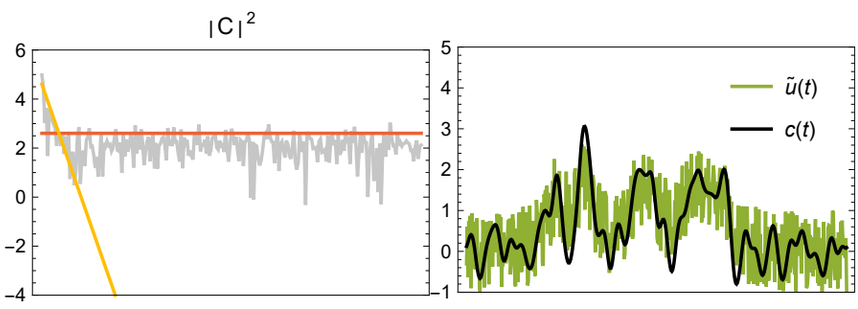
Signal smo razmeroma dobro rekonstruirali in prepoznamo lahko štiri škatlaste sunke, a je prisotnega še veliko šuma. V resnici je filtrirno funkcijo Φ bolje konstruirati tako, da ocenimo tudi potek signala *S*, npr. v danem primeru z eksponentno funkcijo.





Signal je zdaj več kot odlično rekonstruiran! Vrhovi so sicer manj zašiljeni, ker smo odstranili visoke frekvence, ki so nujne za hitre prehode. Enako storimo še za preostala dva zašumljena signala.

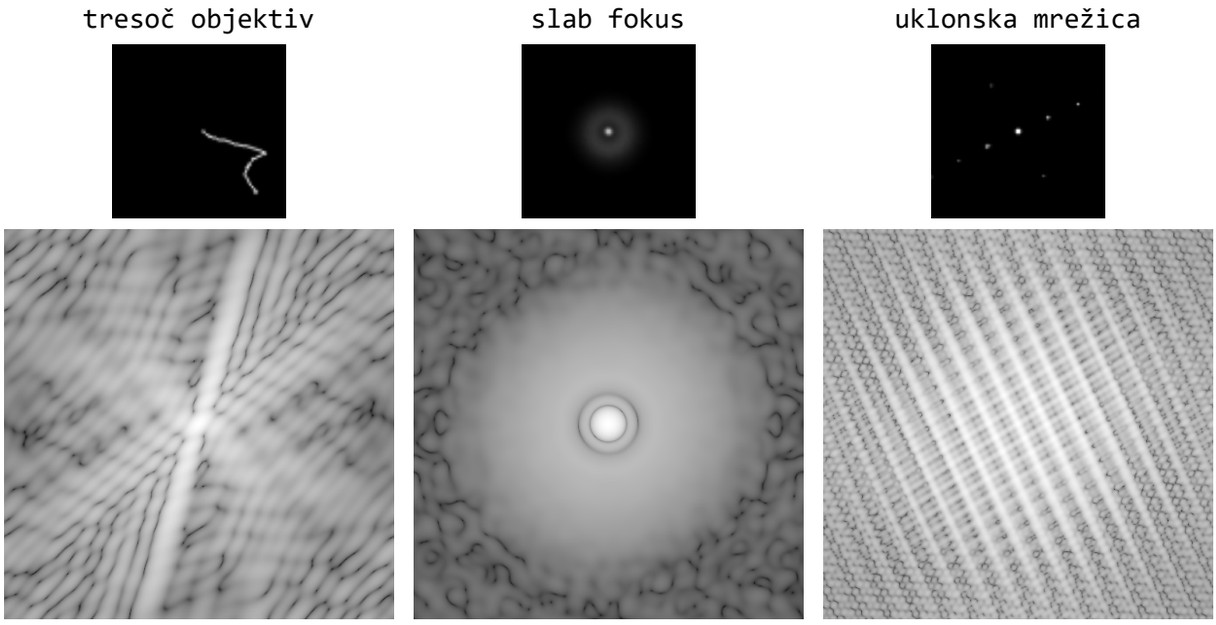




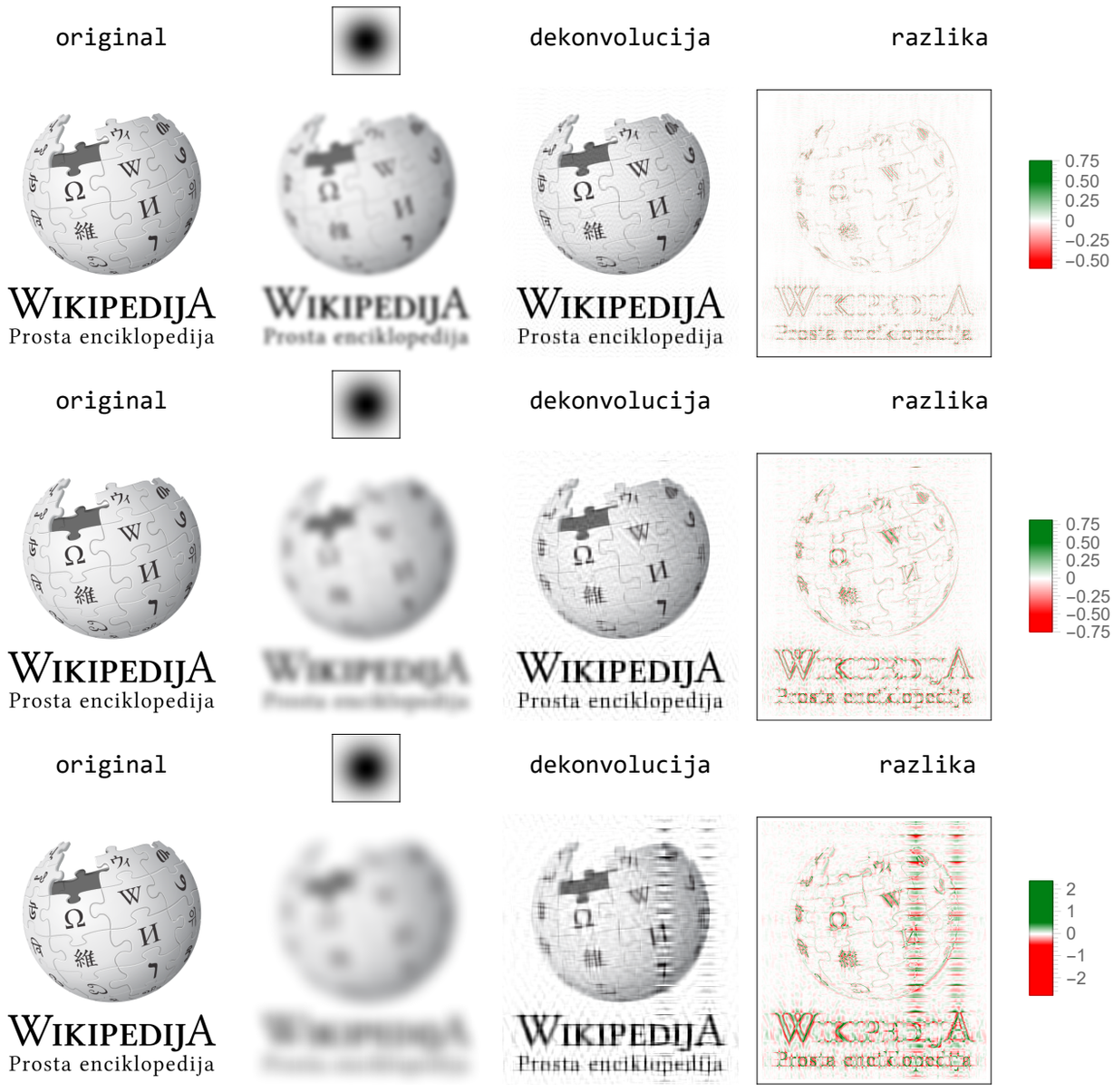
Opazimo, da je v zadnjem primeru šum na signalu že tako velik, da je končna rekonstrukcija razmeroma slaba, in v njej bi težko prepoznali škatlaste sunke.

3. Filtriranje slike

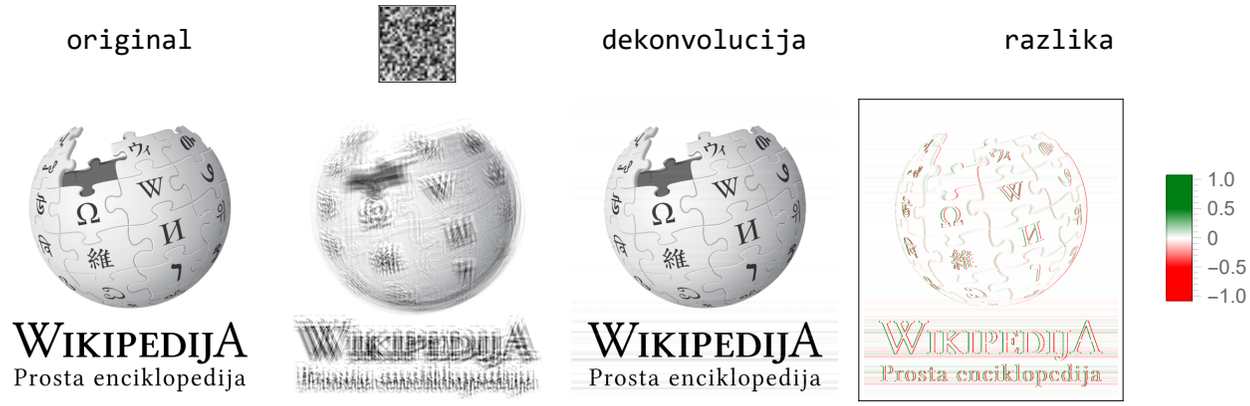
V zadnjem delu naloge bomo Wienerjev filter uporabili še na slikah. Popačitve slik so pogosto konvolucijske narave, zato jih lahko do neke mere odpravimo, če le poznamo ustrezno prenosno funkcijo (*psf*), s katero je popačitev nastala. Oglejmo si najprej tri prenosne funkcije iz naloge (prikazan je samo njihov osrednji del) in pripadajoče spektralne gostote moči.



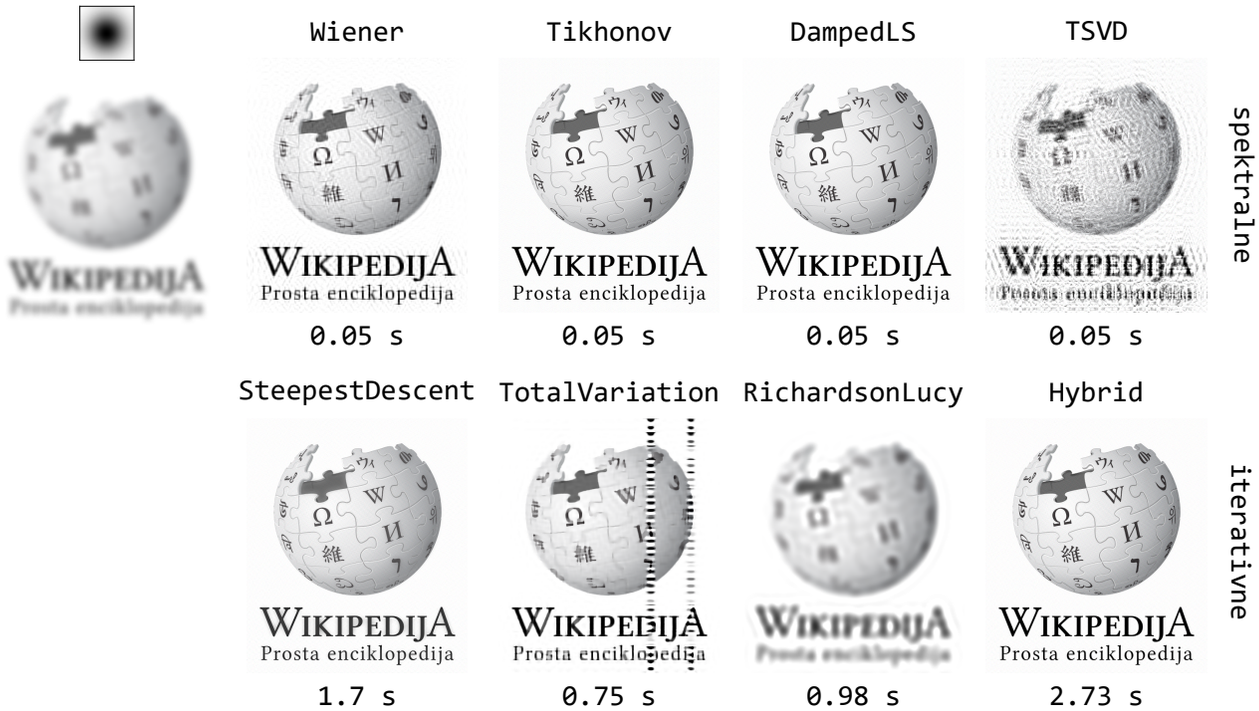
Preden se posvetimo že popačenim slikam iz naloge, si oglejmo, kaj lahko dekonvolucija naredi, če sami izvedemo popačitev. Za to bomo uporabili vgrajeni funkciji **`ImageConvolve`** in **`ImageDeconvolve`**, ki ji bomo ročno nastavili Wienerjevo metodo.



Opazimo, da dekonvolucija tudi pri znatno zamegljeni sliki lahko odlično rekonstruira prvotno sliko. Pri večjih jedrih se začnejo pojavljati artefakti. Zanimivo si je ogledati, da dekonvolucija deluje tudi pri naključnem jedru, ko postane konvoluirana slika zares izmaličena.



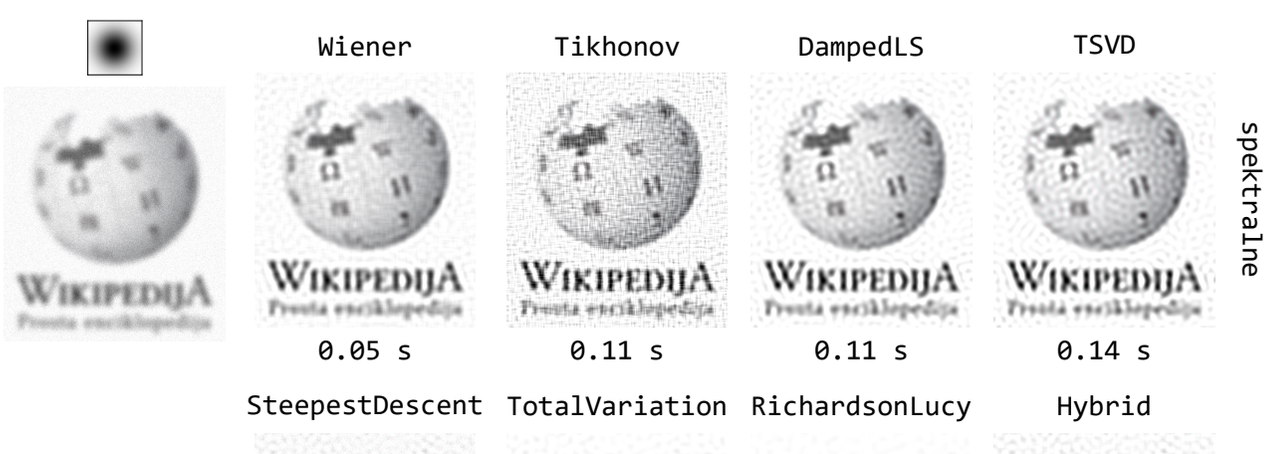
Pravzaprav pa lahko za dekonvolucijo uporabimo tudi drugačne metode. Oglejmo si vse vgrajene metode.



Opazimo, da dajo prve štiri metode, ki so *spektralne metode*, v danem primeru nekoliko boljše rezultate kot zadnje štiri metode, ki so *iterativne metode*. Spektralne so prav tako znatno hitrejše. DampedLS je drugo ime za Levenberg-Marquardtovo metodo, ki izvaja numerično nelinearno minimizacijo napake. Tikhonova regularizacija pa je način regularizacije, ki pomaga pri reševanju problema dekonvolucije (ta je namreč v splošnem slabo pogojen, med drugim težave predstavlja deljenje z majhnimi vrednostmi, kar ojača visokofrekvenčni šum).

Če k sliki dodamo še šum, pa postanejo razlike med metodami manjše. Naj še omenimo, da smo spektralnim metodam, ko sliki ni bil primešan šum, določili, naj ne uporablja regularizacije, medtem ko smo ji v prisotnosti šuma pustili proste roke, da sama avtomatsko oceni stopnjo šuma in s tem stopnjo regularizacije.

```
convDeconvMethod[wiki, GaussianMatrix[15], .05]
```



iterativne

1.58 s | 0.95 s | 1.17 s | 1.91 s

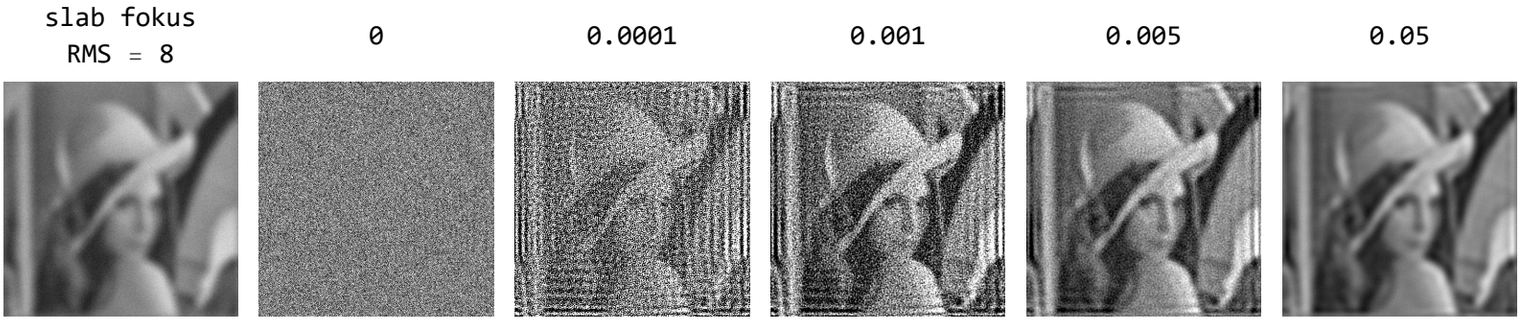
Sedaj se posvetimo Leni s fotografij iz naloge.

tresoči objektiv RMS = 0 | tresoči objektiv RMS = 4 | tresoči objektiv RMS = 8 | tresoči objektiv RMS = 16

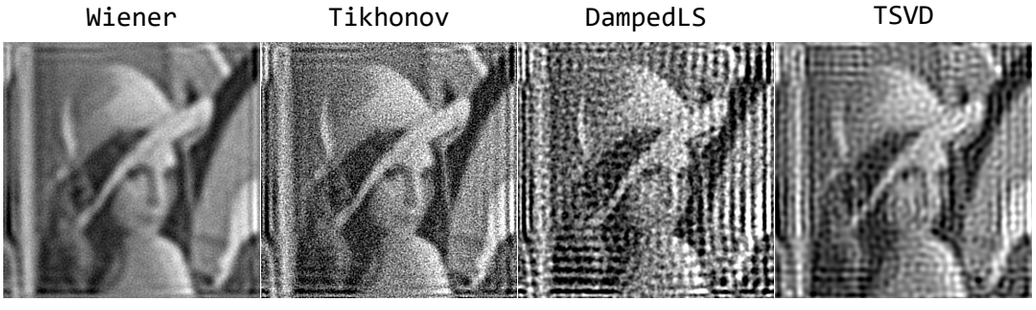
slab fokus RMS = 0 | slab fokus RMS = 4 | slab fokus RMS = 8 | slab fokus RMS = 16

uklonska mrežica RMS = 0 | uklonska mrežica RMS = 4 | uklonska mrežica RMS = 8 | uklonska mrežica RMS = 16

Opazimo, da smo pri vseh slikah uspeli doseči neko stopnjo izboljšanja. Pri slabem fokusu se zdi to izboljšanje najmanjše. Z ročnim nastavljanjem stopnje regularizacije lahko vplivamo na kompromis med zašumljenostjo slike in robnimi popačitvami.



Ali se morda bolj obnesejo ostale metode? Niti ne.

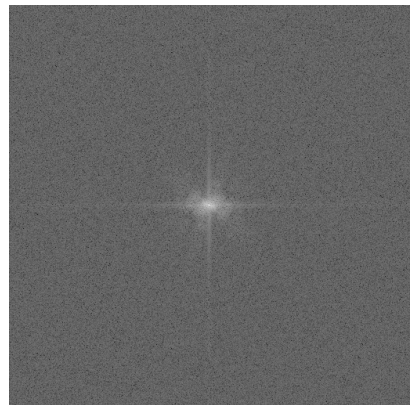


0.05 s 0.08 s 0.05 s 0.08 s

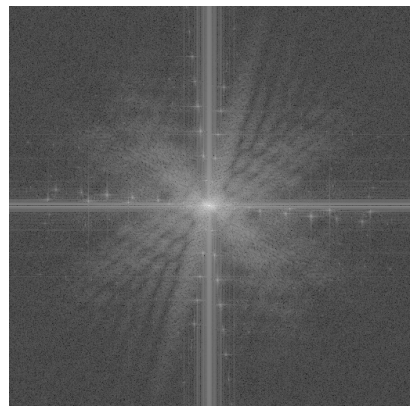
7.45 s 1.83 s 2.55 s 7.75 s

Za konec si oglejmo še primer periodičnega šuma. Ta je najbolje viden v periodogramu slike, kjer so prisotne enakomerno razmaknjeni vrhovi. Opazimo, da vrhovi ležijo v več različnih smereh, kar nakazuje, da je bil dodan periodični šum v več različnih smereh.

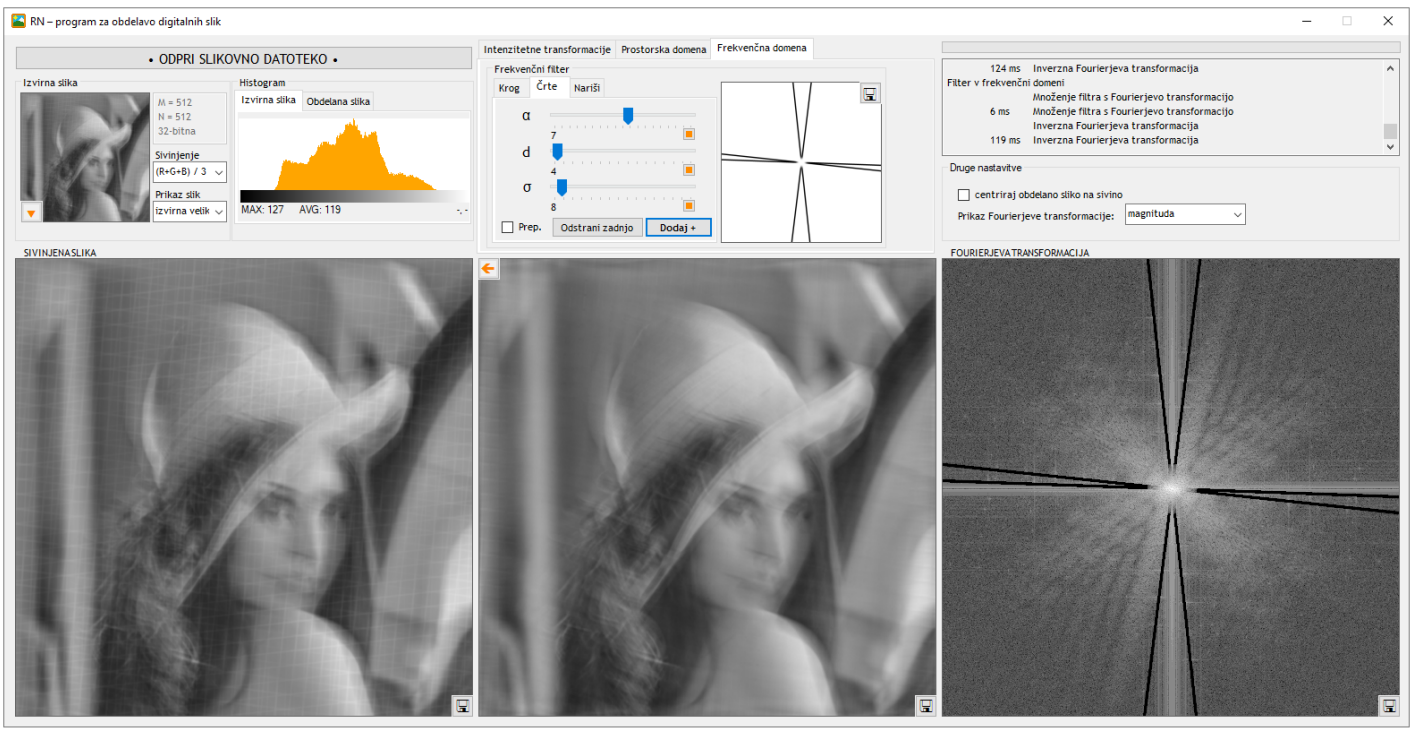
Gaussov šum



periodični šum



Za odstranjevanje periodičnega šuma bom uporabil program, ki sem ga izdelal v sklopu raziskovalne naloge v srednji šoli. Omogoča nam, da v spektru poljubno odstranjujemo želena območja (npr. z risanjem ali dodajanjem črt).



Filtrirano sliko sedaj spustimo skozi dekonvolucijo z ustreznim jedrom (tresoči objektiv), da dobimo končno restavrirano sliko, s katero smo lahko zadovoljni.