

Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



NALOGA 4: FOURIEROVA ANALIZA

MATEMATIČNO-FIZIKALNI PRAKTIKUM

DIMITRIJE PEŠIČ

VPISNA ŠTEVILKA: 28201072

PREDAVATELJ: PROF. DR. BORUT PAUL KERŠEVAN

LJUBLJANA, 10.10.2023

1 Uvod

Naloga je od nas zahtevala, da izračunamo Fourierov obrat Gaussove porazdelitve in nekaj enostavnih vzorcev, npr. mešanic izbranih frekvenc. Za slednje smo primerjali rezultate, ko je vzorec v intervalu periodičen (izbrane frekvence so mnogokratniki osnovne frekvence), z rezultati, ko vzorec ni periodičen (kako naredimo Gaussovo porazdelitev ‘periodično’ za FT?). Opazuj pojav potujitve na vzorcu, ki vsebuje frekvence nad Nyquistovo frekvenco. Napravi še obratno transformacijo (5) in preveri natančnost metode. Poglej, kaj se dogaja z časom računanja - kako je odvisen od števila vzorčenj? Nato smo še morali z Fourierom analizirati 2.3 s dolge zapise začetka Bachove partite za violino solo.

2 Ozadje

Pri numeričnem izračunavanju Fourierove transformacije

$$H(f) = \int_{-\infty}^{\infty} h(t) \exp(2\pi i f t) dt \quad (1)$$

$$h(t) = \int_{-\infty}^{\infty} H(f) \exp(-2\pi i f t) df \quad (2)$$

je funkcija $h(t)$ običajno predstavljena s tablico diskretnih vrednosti

$$h_k = h(t_k), \quad t_k = k\Delta, \quad k = 0, 1, 2, \dots, N-1. \quad (3)$$

Pravimo, da smo funkcijo vzorčili z vzorčno gostoto (frekvenco) $f = 1/\Delta$. Za tako definiran vzorec obstaja naravna meja frekvenčnega spektra, ki se imenuje *Nyquistova frekvenca*, $f_c = 1/(2\Delta)$: harmonični val s to frekvenco ima v vzorčni gostoti ravno dva vzorca v periodi. Če ima funkcija $h(t)$ frekvenčni spekter omejen na interval $[-f_c, f_c]$, potem ji z vzorčenjem nismo odvzeli nič informacije, kadar pa se spekter razteza izven intervala, pride do *potujitve* (*aliasing*), ko se zunanji del spektra preslika v interval.

Frekvenčni spekter vzorčene funkcije (3) računamo samo v N točkah, če hočemo, da se ohrani količina informacije. Vpeljemo vsoto

$$H_n = \sum_{k=0}^{N-1} h_k \exp(2\pi i k n / N), \quad n = -\frac{N}{2}, \dots, \frac{N}{2}, \quad (4)$$

ki jo imenujemo diskretna Fourierova transformacija in je povezana s funkcijo v (1) takole:

$$H\left(\frac{n}{N\Delta}\right) \approx \Delta \cdot H_n.$$

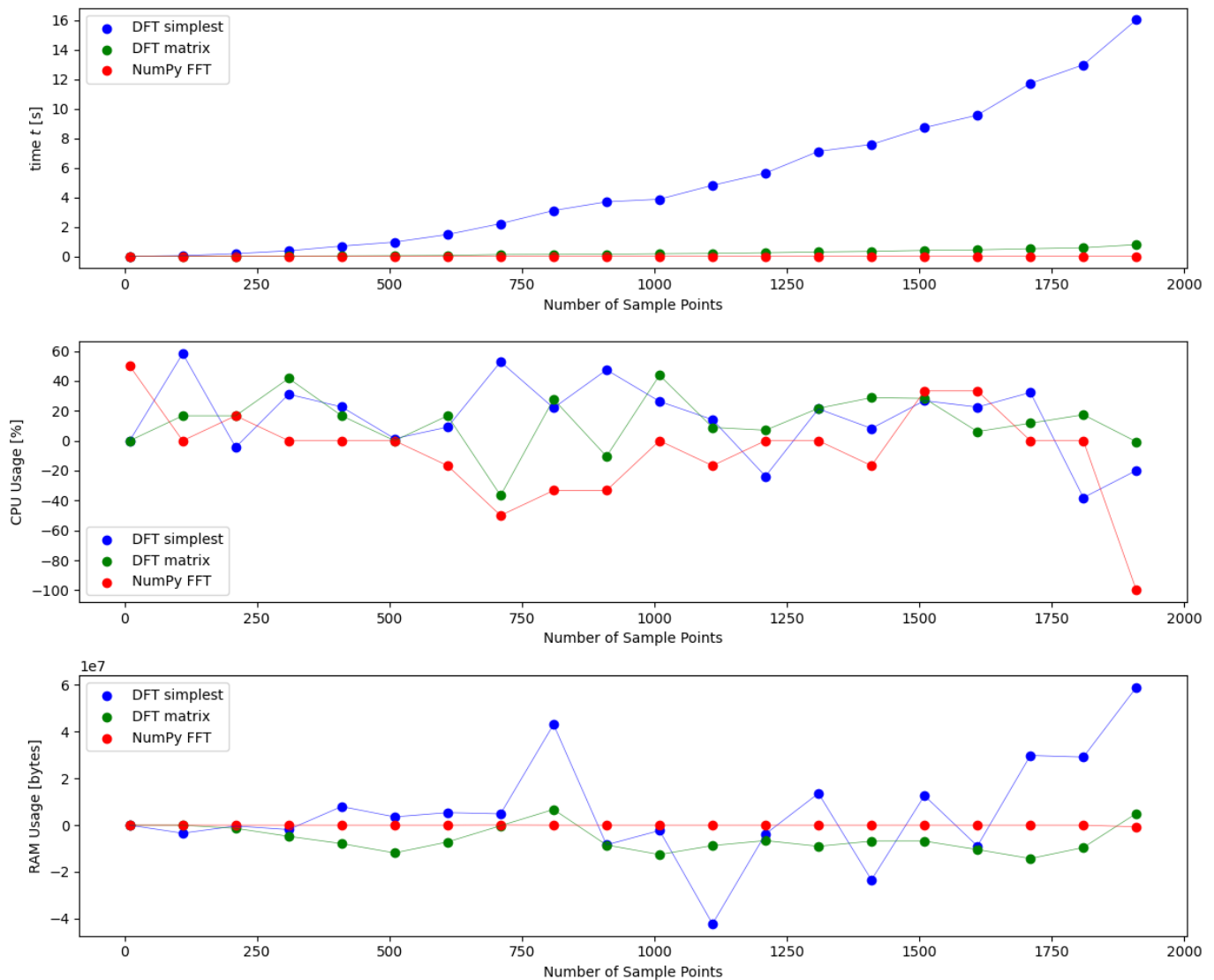
V tesni zvezi s frekvenčnim spektrom je tudi moč. *Celotna moč* nekega signala je neodvisna od reprezentacije, Parsevalova enačba pove

$$\sum_{k=0}^{N-1} |h_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |H_n|^2$$

Z obratno transformacijo lahko tudi rekonstruiramo h_k iz H_n

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n \exp(-2\pi i k n / N) \quad (5)$$

(razlika glede na enačbo (4) je le predznak v argumentu eksponenta in utež $1/N$).



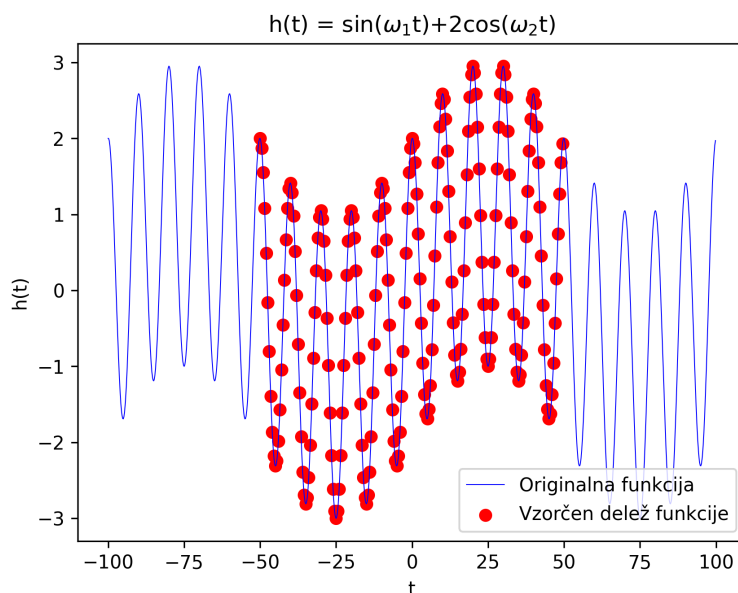
Slika 1: Čas izvajanja, poraba procesorja in pomnilnika.

3 Implementacija

Fourierovo transformacijo sem vpeljal na dva načina, eno z dvema *for* zankama in eno z uporabo matričnih operacij, ki jih podpira *Numpy* knjižnica. Za prvo '*preprostejšo*' metodo sem pričakoval odvisnost $\mathcal{O}(n^2)$, za drugo '*matrično*' metodo sem pa pričakoval povišanje v uporabi pomnilnika. Seveda je le to treba testirati, za primerjavo sem pa uporabil vgrajeno hitro fourierovo transformacijo *numpy.fft.fft()*.

Na grafih 1 so prikazani grafi časa izvedbe, porabe procesorja in porabe pomnilnika v odvisnosti od števila vzorčenj.

Kot je razvidno, je najpočasnejša najpreprostejša metoda, matrična metoda pa karnekaj časa drži tempo z hitro fourierovo transformacijo, ki je seveda bliskovita in ima minimalen čas izvedbe. Grafa za procesor in pomnilnik nam pa ne poveta veliko, saj se ne opazi drastične razlike v porabi le teh, razlike v porabi pa so le artefakt poslušanja glasbe in 93 hkrati odprtih zavihkov na brskalniku. Poskusil sem izračunati merilo uspešnosti tudi za višje frekvence vzorčenja do 44100Hz, vendar je računalnik potreboval veliko časa, katerega žal, za čeznočno simulacijo, nisem imel, poleg tega se pa za frekvence nad 22000Hz program sesuje.



Slika 2: Začetni signal in vzorčen delež.

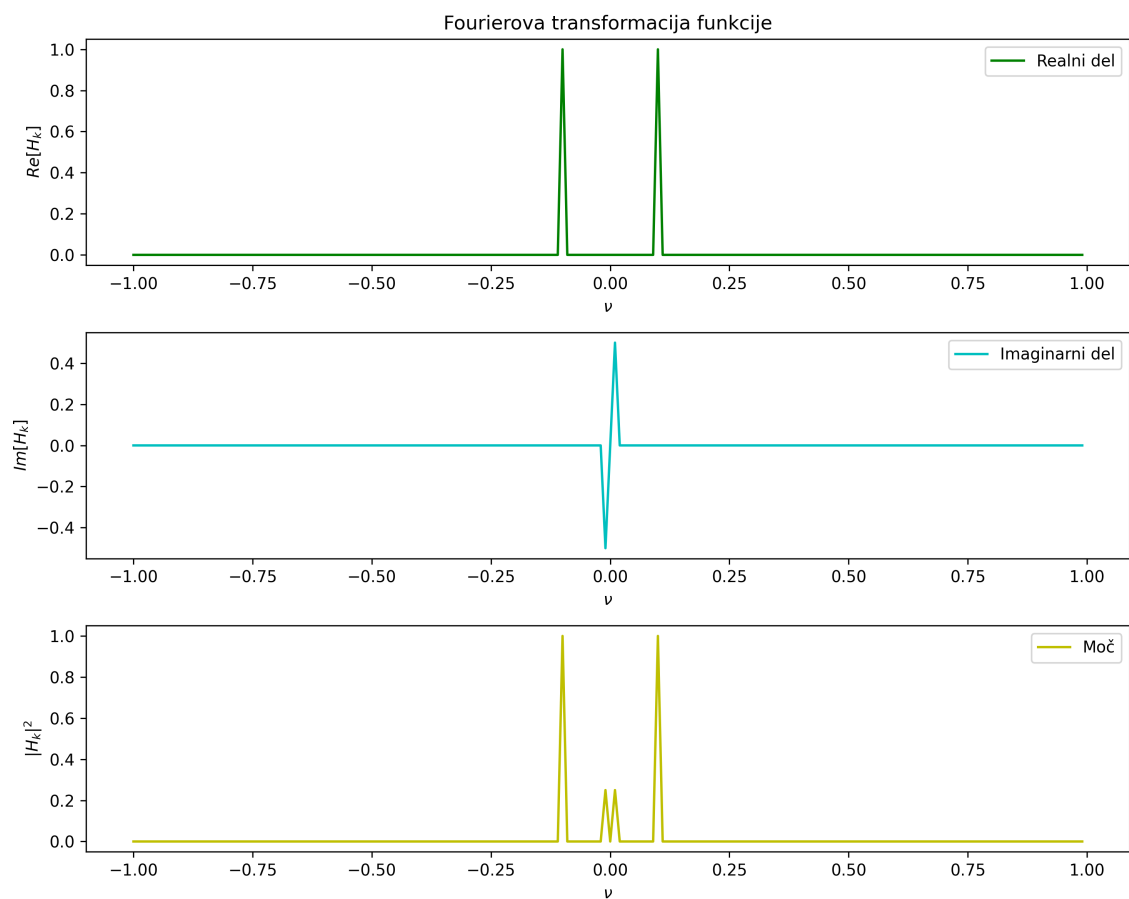
4 Rezultati

Sedaj ko imamo našo fourierovo transformacijo implementirano, se lahko lotimo uporabo transformacije na različnih funkcijah. Želel sem se izogniti uporabi vgrajene funkcije saj je hitra fourierova transformacija debata naslednje naloge, vendar mi hitro zmanjka pomnilnika za to matrično metodo in časa za preprosto metodo, zato sem vzel `numpy.fft.fft()`.

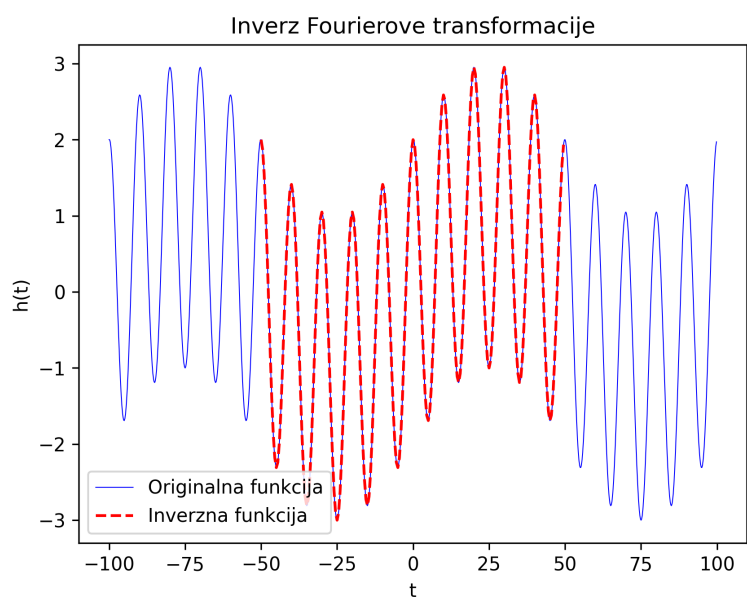
Vzorčil sem signal $h(t) = \sin(\omega_1 t) + 2 \cos(\omega_2 t)$, kjer sta $\omega_1 = 2\pi/100$, in $\omega_2 = 2\pi/10$. Vzorčil sem z periodo ± 100 in številom točk $n = 200$. Na grafu 2 je prikazana začetna funkcija in vzorčen delež, na grafu 3 so prikazani realni in imaginarni del ter moč, na grafu 4 pa je prikazana originalna funkcija in inverzna Fourierova transformacija (z drugimi besedami rekonstruirana funkcija).

4.1 Število točk

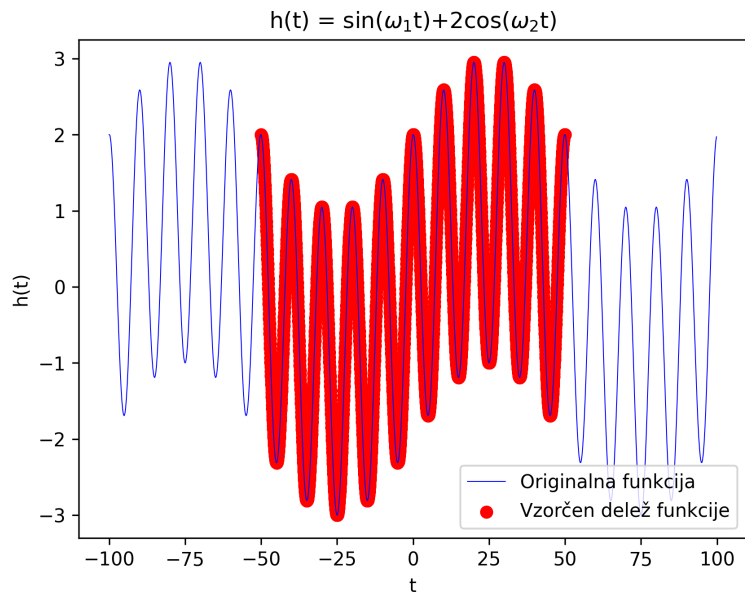
Če pošljemo število točk vzorčenja proti vse večjem številu, se naše delta funkcije 'izostrijo'. To je seveda logično saj pri numeričnem računanju le aproksimiramo Fourierovo transformacijo za končno število členov namest neskončnega integrala. Artefakt prikazan na grafih 5, 6, 7



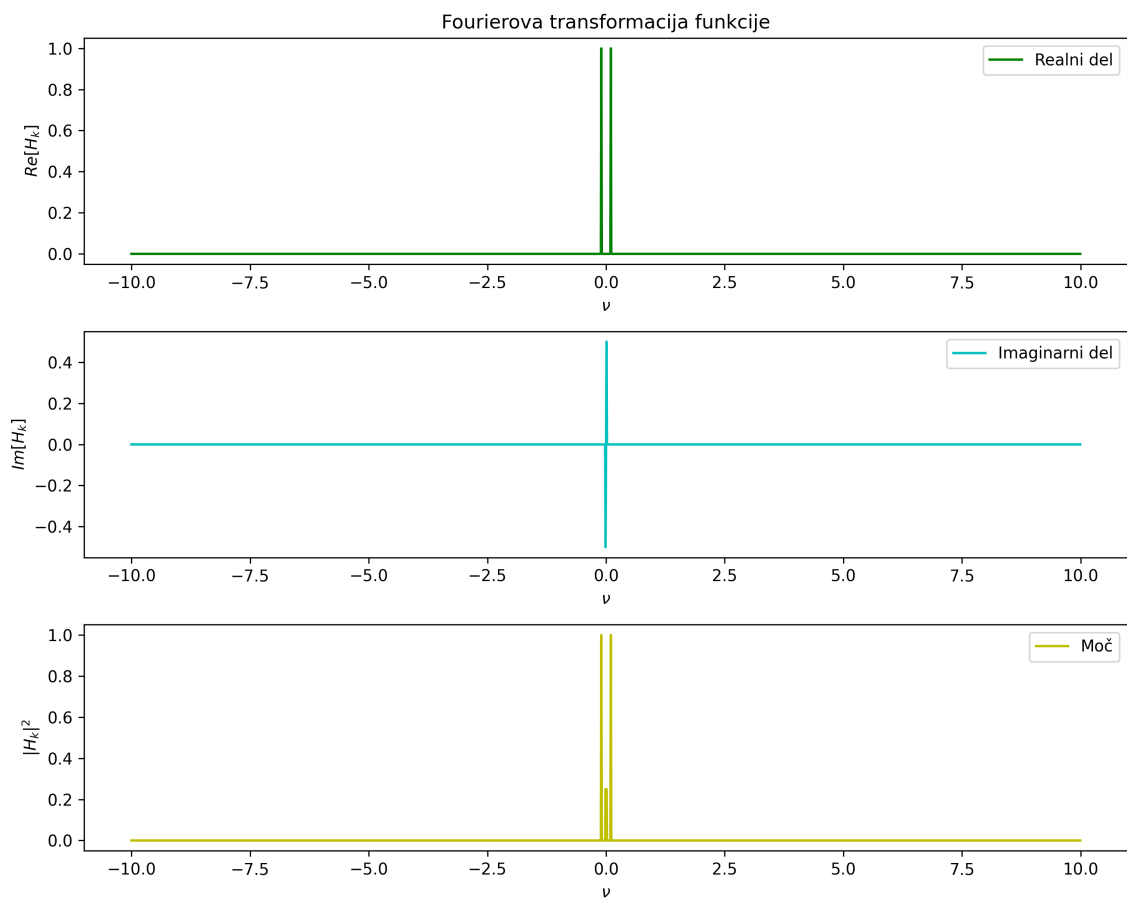
Slika 3: Realni in imaginarni del ter moč.



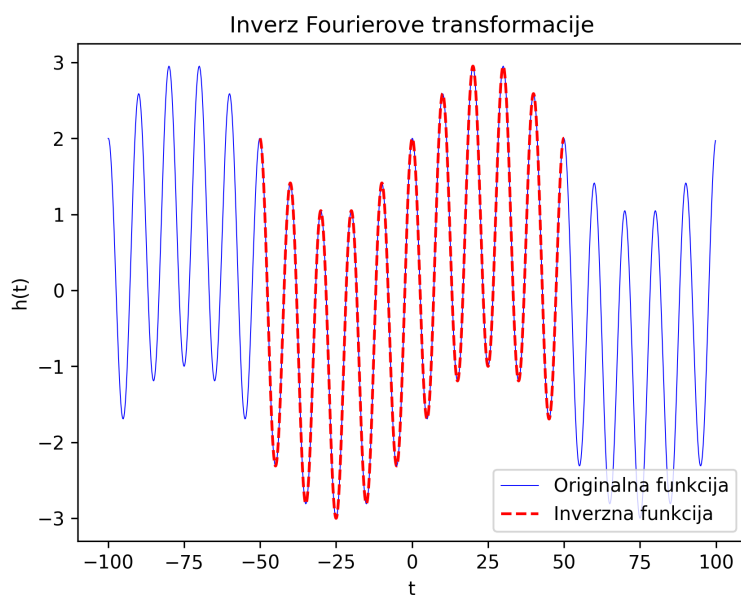
Slika 4: Inverzna transformacija.



Slika 5: Začetni signal in vzorčen delež.



Slika 6: Realni in imaginarni del ter moč.



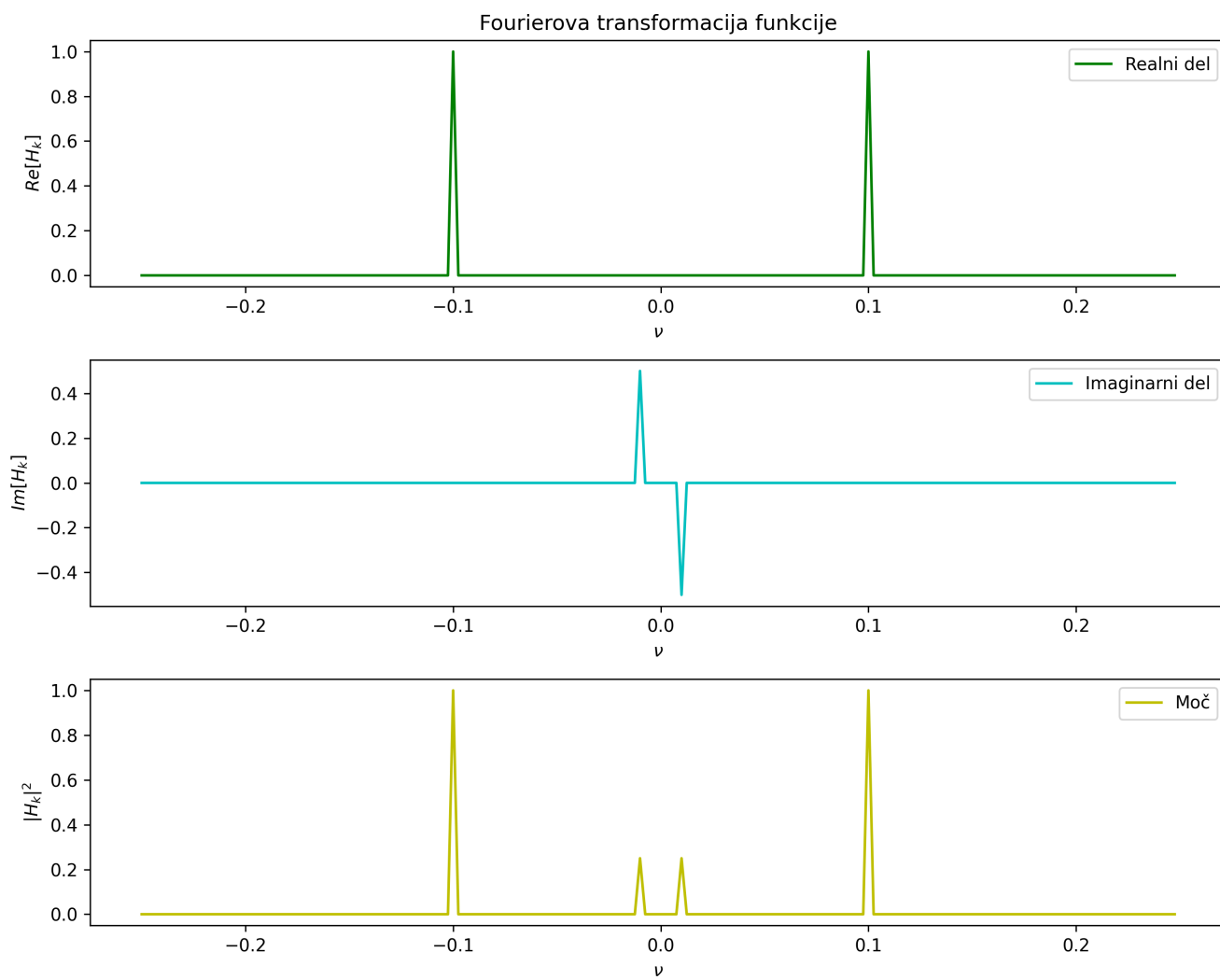
Slika 7: Inverzna transformacija.

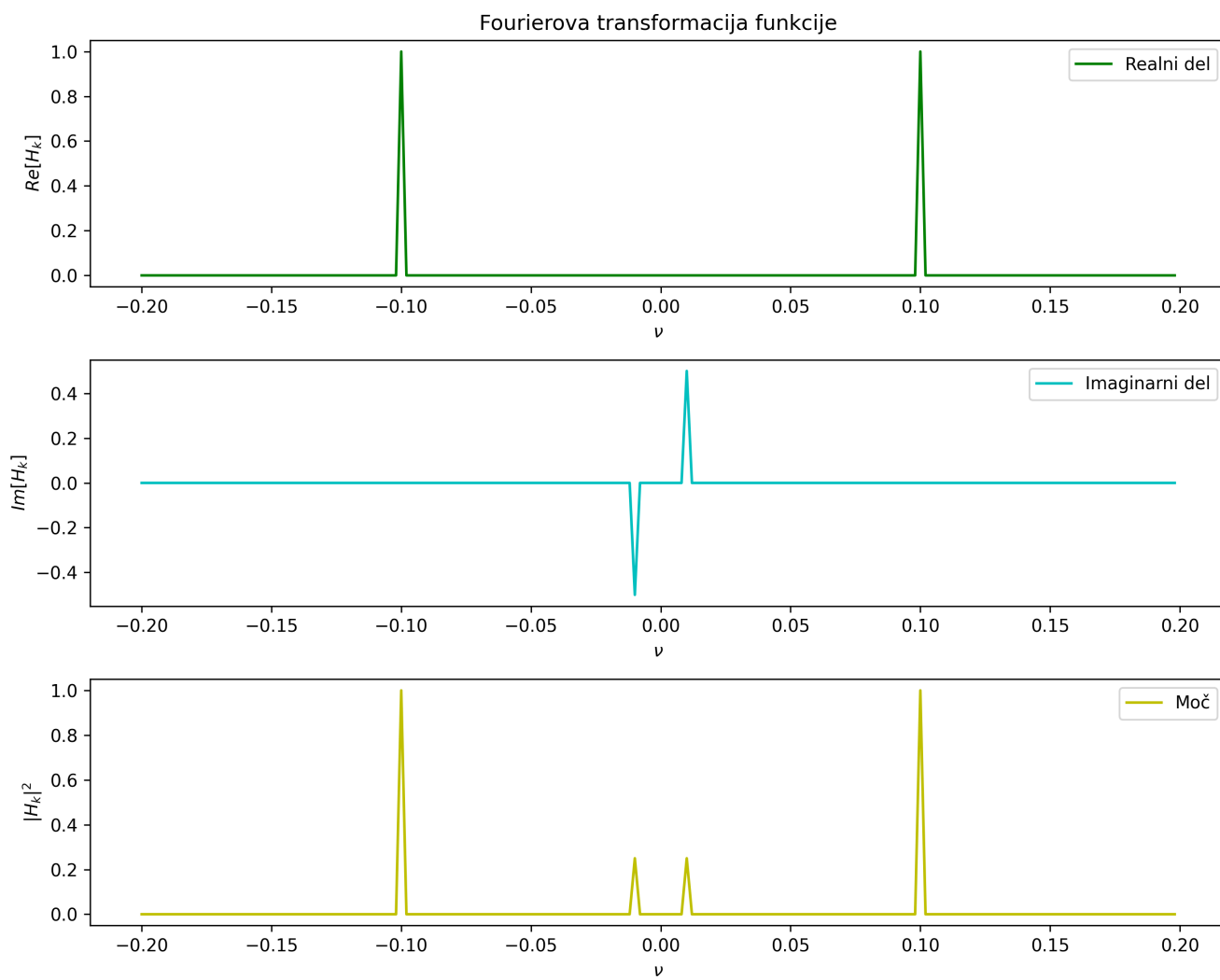
4.2 Potujevanje (Angl. *Aliasing*)

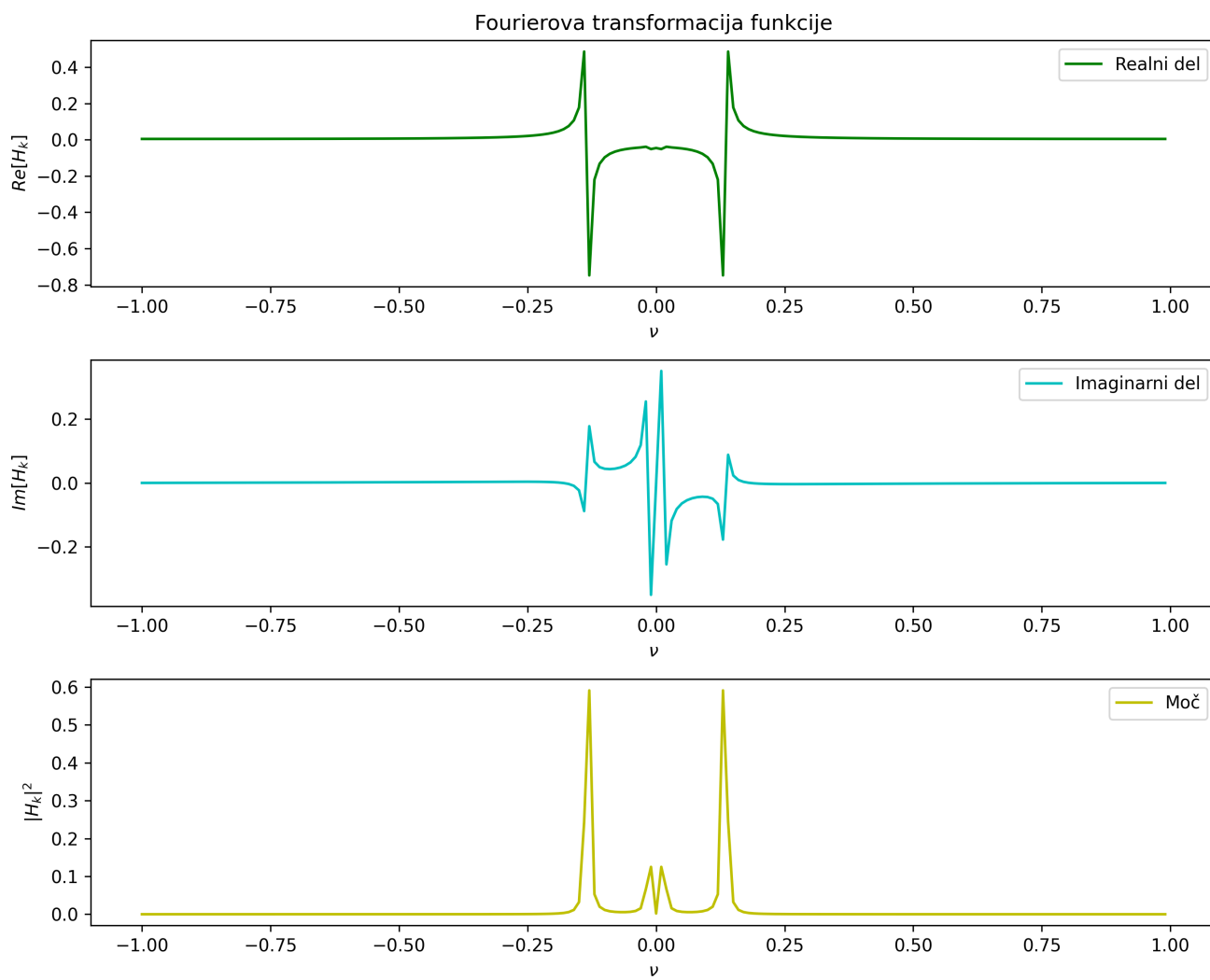
Nato sem si ogledal, kaj se zgodi če naredimo Fourierovo transformacijo enakih signalov, le z različnimi frekvencami, takšnimi, ki niso celoštevilski večkratniki. Opazimo, da je prišlo do razširitve črt in nekaj čudnosti v obeh komponentah transformacije.

Po Nyquist-Shannonovem teoremu [2], ki pravi, da mora biti periodičen signal vzorčen z frekvenco vzorčenja, ki ima vsaj dvakrat večjo frekvenco kot največja komponente frekvence vzorčenega signala, rekonstrukcija signala, ki ima frekvenčne komponente nad Nyquistovo frekvenco ni več možna, ker pride do potujevanja. To je razvidno iz grafov 4.2, 4.2, 4.2 kot 'zaviti uhlji' in oddaljevanje.

Za bodočo izboljšavo poročila bi lahko dodal še študijo popravljanja signala z umetnimi ničlami, tako imenovana 'zero-padding' metoda. Vendar je zdaj ura 6.30 zjutraj in več ne zmorem.

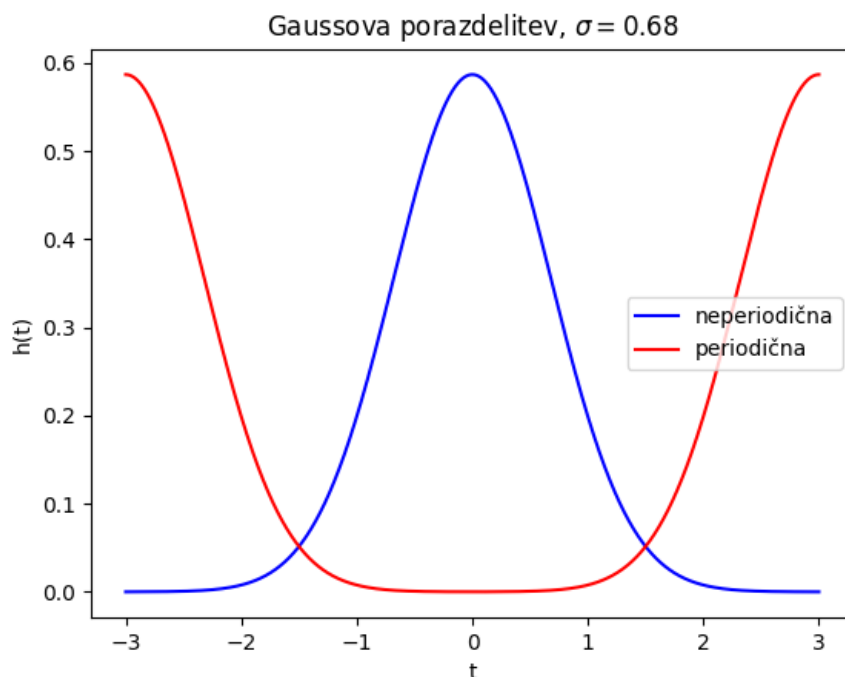






4.3 Gaussova porazdelitev

Kot smo se učili pri Matematiki IV 1, za Gaussovo porazdelitev velja, da je njena Fourierova transformiranka tudi Gaussova porazdelitev. Seveda situacija ni tako preprosta. Kot opisano v poglavju zgoraj, DFT privzame da je koordinatno izhodišče prvi element in računa le z pozitivnimi vrednostmi tako da razporedi frekvence od 0 do f_c , nato pa od $-f_c$ do 0, privzame pa tudi periodičnost, kar pomeni da ko zapeljemo iz desnega roba naj bi prišli v koordinatno izhodišče. Gaussova funkcija ni periodična, zato uporaba DFT povzroči hitro osciliranje kot je razvidno na sliki 10 z zeleno. Tega artefakta se znebimo tako, da ali vnaprej zamaknemo celotno Gaussovo porazdelitev s funkcijo `numpy.roll`, kot je razvidno na grafu ?? (graf 10 se nanaša na začetno porazdelitev predstavljeno na grafu ??, vendar sem dodal ?? saj je bolj razvidno kako izgleda naš postopek premika), ali pa oscilirajoč rezultat DFT skaliramo s faktorjem $g(x) = f(x - a) \implies G(\omega) = F(\omega)e^{2i\pi \cdot a\omega}$. Jaz sem se odločil za prvotni način.



Slika 8: Začetno stanje Gaussove porazdelitve.

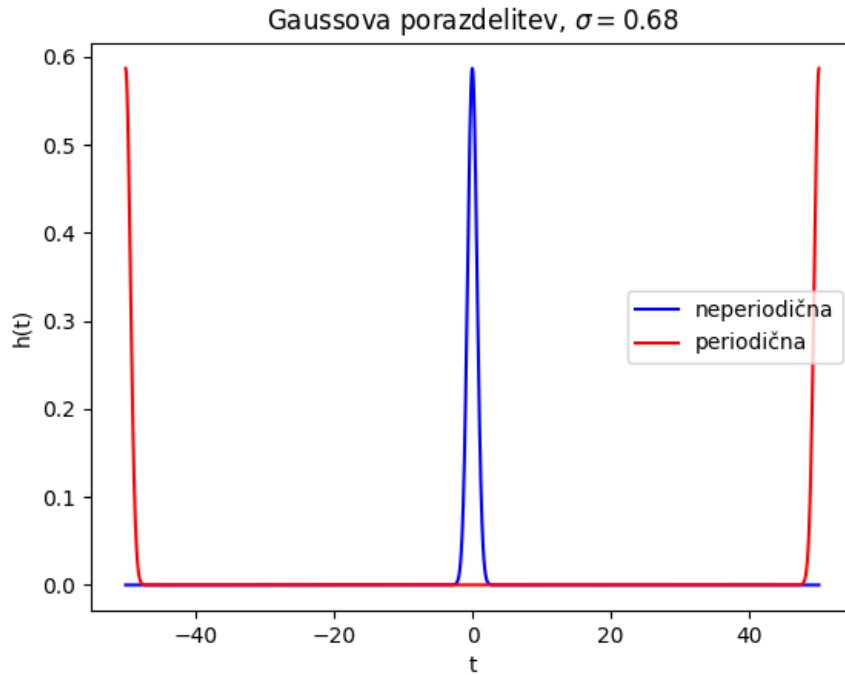
Na grafu 10 je prikaza Fourierova transformacija Gaussove porazdelitve, realna in imaginarna komponenta, ter 'moč'. Grafi so približani za lepšo razvidnosti.

5 Bachove partite

S poslušanjem zapisov v formatu `.mp3` sem ugotovil, da gre za degradacijo zvoka. Naša ušesa slišijo frekvence od 20Hz do 20kHz, temu primerno so zvočni zapisi vzorčeni pri 44100Hz. Zato Bachova partita pri le tej frekvenci vzorčenja zveni najbolj bistro in ostro, ko pa znižujemo frekveno vzorčenja zveni kot da se zvok oddaljuje, je v drugi sobi ali dušen.

Signal iz začetnih taktov partite je bil vzorčen pri 44 100 Hz, 11 025 Hz, 5512 Hz, 2756 Hz, 1378 Hz in 882 Hz.

Na grafih 11, 12, 13, 14, 15, 16 so prikazani rezultati za posamezne frekvence.



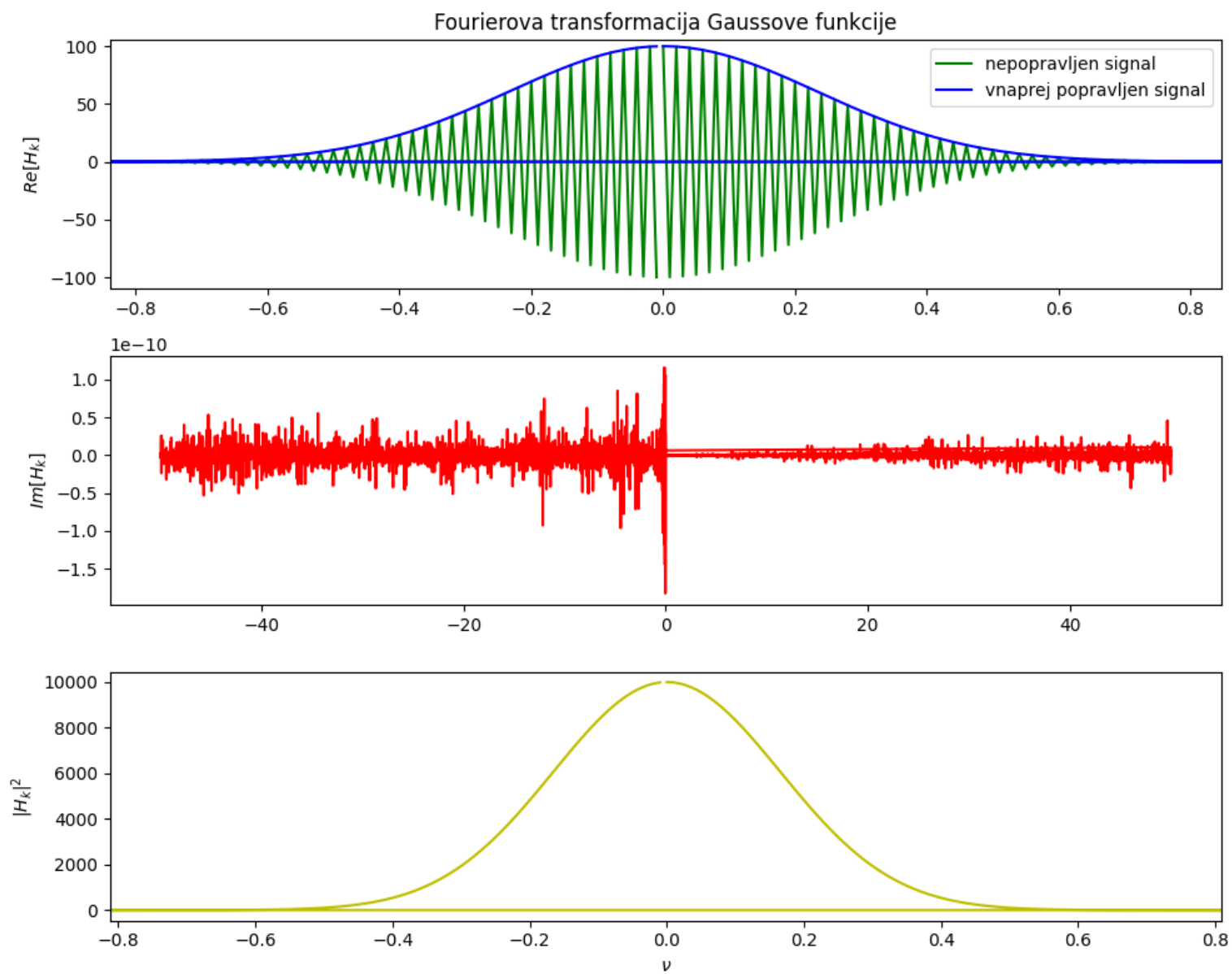
Slika 9: Začetno stanje Gaussove porazdelitve.

6 Zaključek

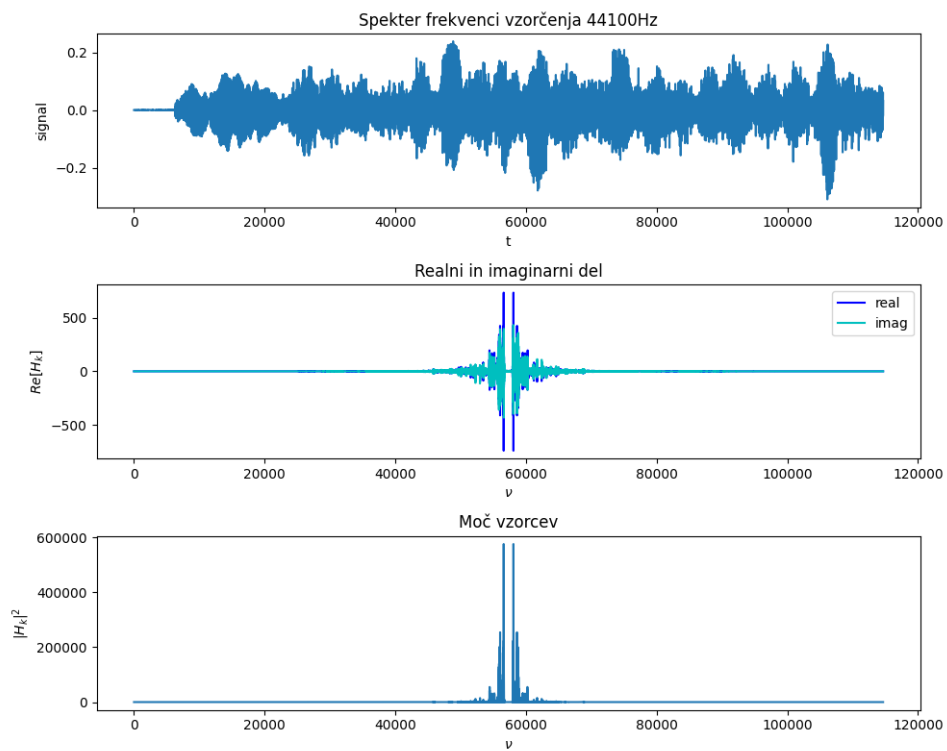
Cilj naloge je bil, da se spoznamo z delovanjem Fourierove transformacije, eno najboljših orodij, na katerih temelji današnja družba. Osebnost mi je bil že od prvega letnika koncept Fourierove transformacije in razvoja izjemno zanimiv čeprav še dandanes težko razumljiv. Na predavanjih ste omenili tudi uporabo pri *High frequency trading* na borzi. Poleg fizike se zanimam tudi za finančni svet, vendar mi je za zdaj zmanjkalo časa, da bi pri tej nalogi še kaj v tej smeri raziskal. V svojem nenaspanem stanju sem tudi žal spregledal kakšen lep način prikaza toliko podatkov, to vidim kot potrebno izboljšavo v nadaljnje.

Literatura

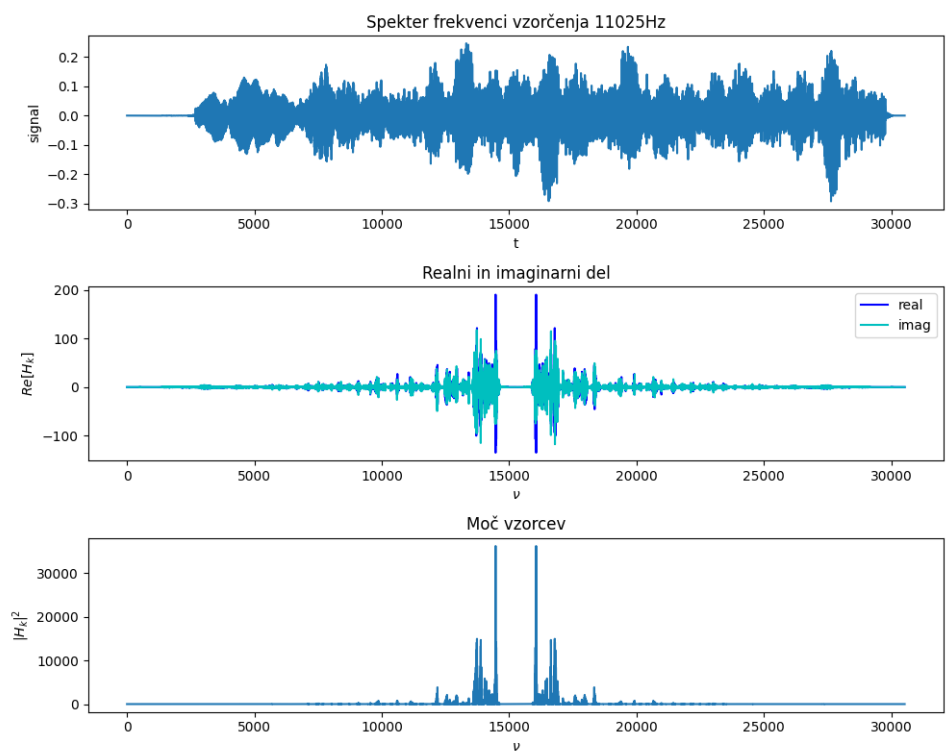
- [1] Simon Bukovšek, *Skripta predavanj Matematika 4, Ljubljana 2023*.
- [2] <https://en.wikipedia.org/w/index.php?title=Nyquist%E2%80%93Shannon%20sampling%20theorem&oldid=1000000000>
sampling theorem



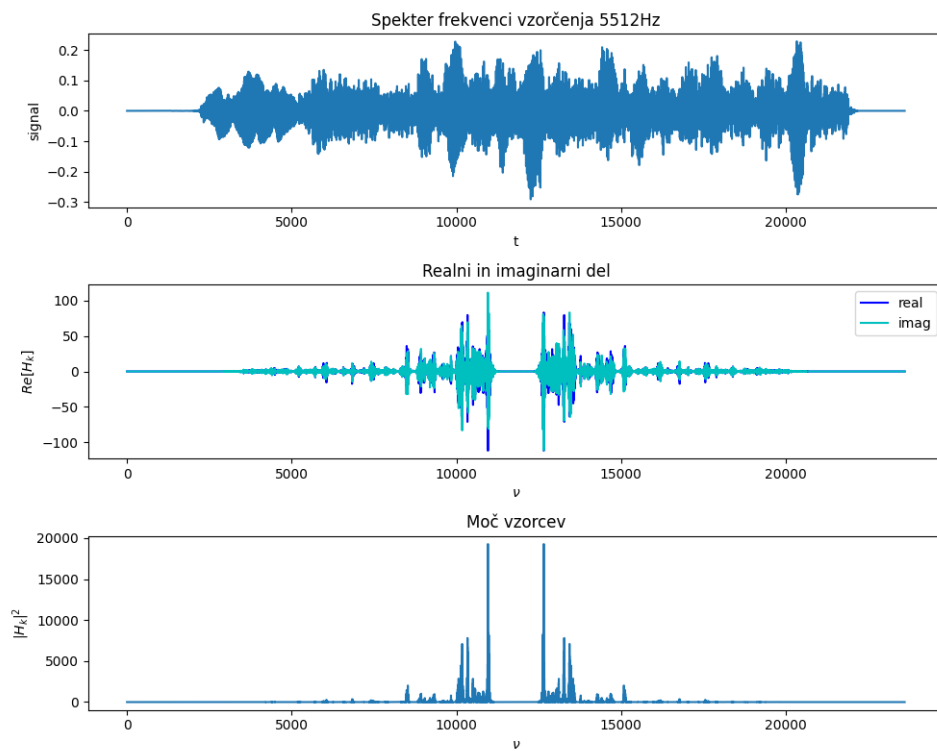
Slika 10: Fourierova transformacija Gaussove porazdelitve.



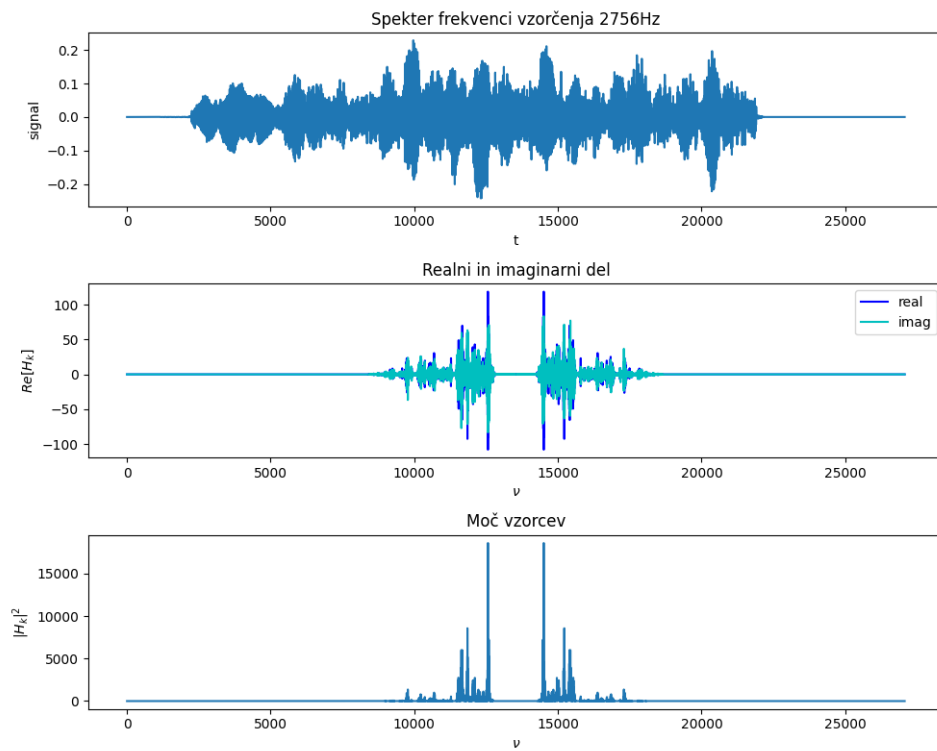
Slika 11: Fourierova transformacija Bachove partite.



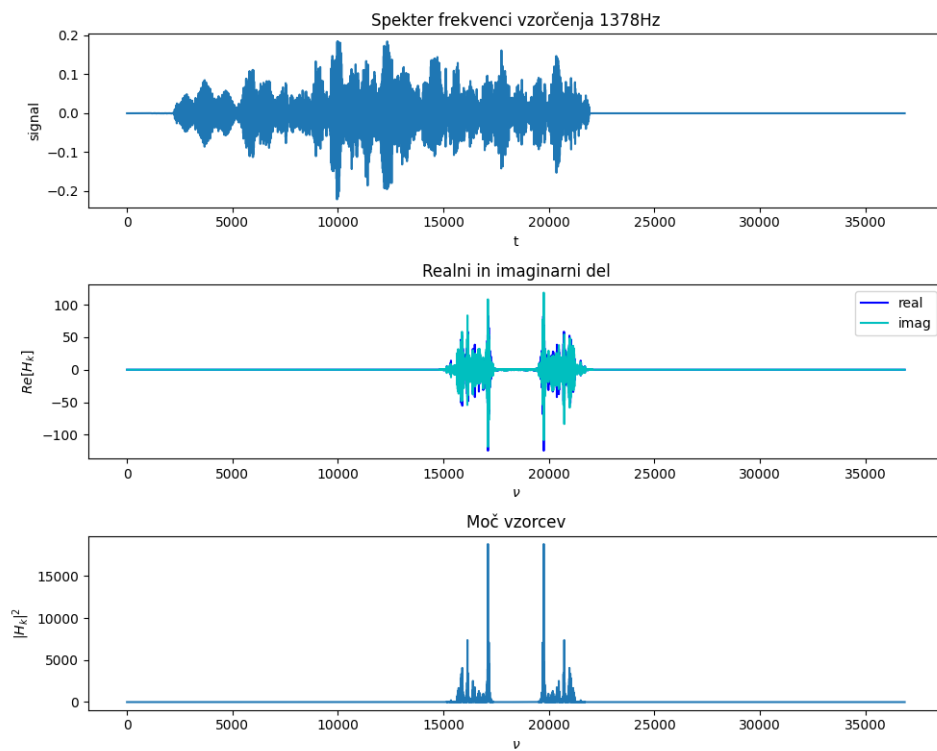
Slika 12: Fourierova transformacija Gaussove porazdelitve.



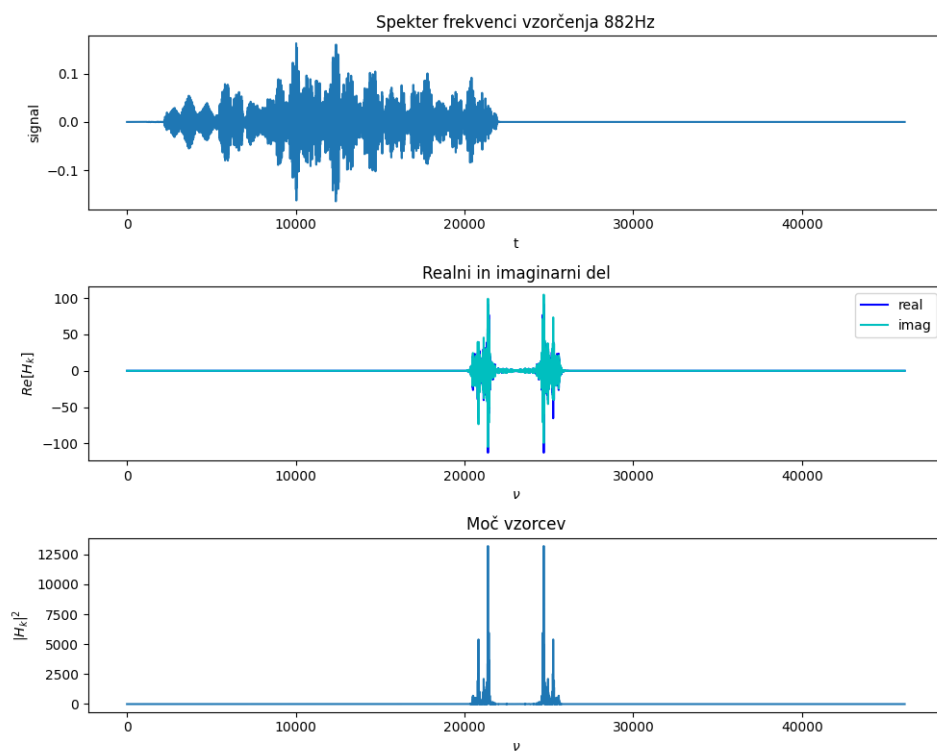
Slika 13: Fourierova transformacija Gaussove porazdelitve.



Slika 14: Fourierova transformacija Gaussove porazdelitve.



Slika 15: Fourierova transformacija Gaussove porazdelitve.



Slika 16: Fourierova transformacija Gaussove porazdelitve.