

Univerza v Ljubljani
Fakulteta za *matematiko in fiziko*



Hitra Fourierova Transformacija (FFT)

5. naloga pri Matematično-fizikalnem praktikumu

Avtor: Marko Urbanč (28191096)
Predavatelj: prof. dr. Borut Paul Kerševan

10.11.2021

Kazalo

1	Uvod	2
2	Naloga	3
3	Opis reševanja	3
4	Rezultati	8
4.1	Avtokorelacijske funkcije vseh signalov	8
4.2	Zašumljeni posnetki	9
4.3	Pulzarji	11
5	Komentarji in izboljšave	14
5.1	DSP in audio efekti	14
	Literatura	15

1 Uvod

Prejšnji teden smo si pogledali diskretno Fourierovo transformacijo, ki smo jo definirali kot:

$$\hat{f}_k = \sum_{n=0}^{N-1} f_n \exp(2\pi kn/N), \quad k = -\frac{N}{2}, \dots, \frac{N}{2}.$$

Ena od ključnih ugotovitev je bila, da ima ta postopek časovno zahtevnost N^2 , kar je lahko zelo dolgo, sploh za obdelavo zvočnih posnetkov, kjer sem si že pri prejšnji nalogi pravzaprav pomagal z izboljšano verzijo. Do izboljšane *Hitre Fourierove transformacije* pridemo v osnovi z razcepom:

$$\hat{f}_k = \hat{f}_k^{sod} + \exp(2\pi kn/N) \hat{f}_k^{lih},$$

kjer izrazimo transformiranko s transformacijama njenih sodih in lihih členov. Vsota vsake od transformacij je zdaj dolžine $N/2$. Če je N enak potenci števila 2, lahko brez težav postopek uporabim rekurzivno za dve podtransformaciji, dokler ne pridemo do takih nizov, ki imajo samo še en člen. Za transformacijo enega člena je transformacija identiteta. Skupna časovna zahtevnost tega postopka je le $N \log_2 N$.

Postopek Hitre Fourierove transformacije naredi še nekaj izboljšav. Da ne iščemo pripadnikov niza po vsej tabeli, si podatke preuredimo. Lahko je pokazati, da je v prvotni tabeli treba med seboj zamenjati podatke, katerih vrstna števila v binarnem zapisu so obrnjena: v novem redu jemljemo člene kar po vrsti. Tudi potenc eksponenta ne izražamo vedno znova s sinusi in kosinusi, pač pa jih računamo z rekurzijo. Tak ali podoben postopek je osnova vseh algoritmov hitre Fourierove transformacije.

Korelacija dveh funkcij je definirana podobno kot konvolucija. Korelacija periodičnih funkcij $g(t)$ in $h(t)$ s periodo T je definirana kot:

$$\phi_{gh}(\tau) = \frac{1}{T} \int_0^T g(t + \tau) h(t) dt,$$

oziroma diskretno

$$\phi_{gh}(n) = \frac{1}{N} \sum_{k=0}^{N-1} g_{k+n} h_k.$$

Pravzaprav računamo skalarni produkt funkcij, ki sta časovno premaknjeni. Če je za določeno vrednost ta funkcija višja kot v okolici nam to pove, da sta si funkciji podobni, ampak ju moramo le zamakniti, da se to vidi. V primeru da sta funkciji oz. signala, za katera to izvednotimo, enaka, računamo njuno *avtokorelacijsko funkcijo*. Ta nam podaja mero za to, ali signal ostaja s pretekanjem časa sam sebi podoben. Če je signal slabo koreliran potem korelacija $\phi_h h(n)$ relaksira h kvadratu povprečnega signala $\langle h \rangle^2$, kjer je

$$\langle h \rangle = \frac{1}{N} \sum_{k=0}^{N-1} h_k.$$

Iz lokalnih maksimov v avtokorelacijski funkciji sklepamo na periodičnosti, bodisi popolne ali približne. Pri periodičnih signalih je tudi avtokorelacijska funkcija striktno periodična, za stohastične procese pa je značilna eksponentna avtokorelacijska funkcija. Praktičen račun po zgornji formuli lahko postane za velike vzorce bistveno prepočasen. Korelacijo raje računamo s Fourierovo transformacijo \mathcal{F} . Wiener–Khinchinov teorem pravi, da je korelacija signalov (ki imata prvi statistični moment konstanten v času in drugi statističen moment končen za vse čase) podana z spektrom moči teh signalov [1]. Dobimo:

$$\phi_{gh}(n) = \frac{1}{N-n} \mathcal{F}^{-1} \left[\hat{g} \cdot (\hat{h})^* \right]$$

Za račun s FFT signale dolžine N najprej prepisemo v dvakrat daljše, periodično zaključene podatkovne nize, $\tilde{h}_n = h_n$, $\tilde{h}_{n+N} = 0$ za $n = 0, \dots, N-1$ in $\tilde{h}_{n+2N} = \tilde{h}_n$. Tedaj se avtokorelacija zapiše v obliki

$$\phi_{hh}(n) = \frac{1}{N-n} \sum_{k=0}^{2N-1} \tilde{h}_{k+n} \tilde{h}_k,$$

kar lahko izračunamo s FFT. Temu pristopu za vzorčenje z dodajanjem ničel imenujemo *zero padding*. Pomaga nam odpraviti potujitev zaradi neperiodičnosti signala.

2 Naloga

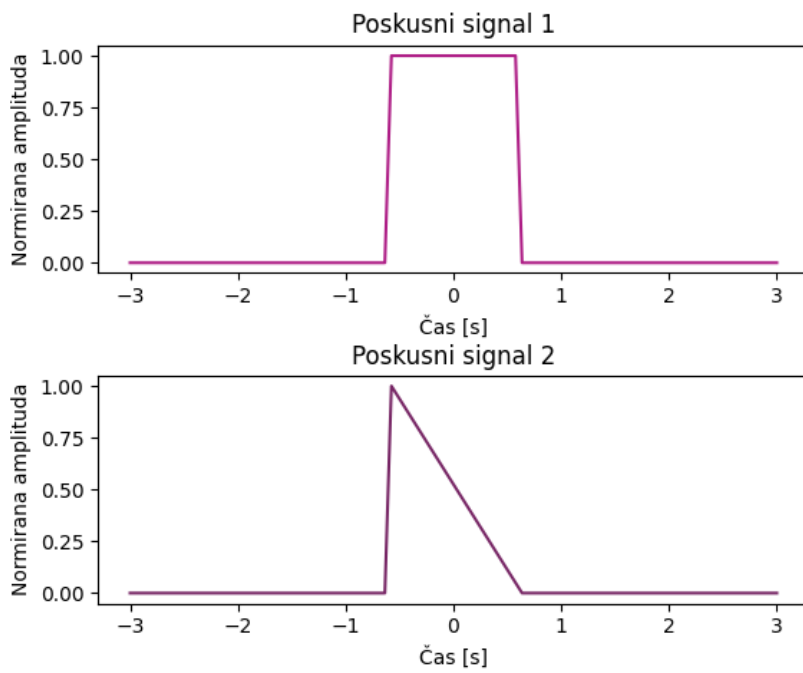
Naloga od nas zahteva, da za podane posnetke oglašanja sove velike uharice izračunamo njihove avtokorelacijske funkcije. Dani so tudi bolj zašumljeni posnetko z deročo reko in glasnimi črički. Z izračunom korelacijskih funkcij poskusimo ugotoviti katera sova se oglašča v zašumljenih posnetkih.

Dodatna naloga: Kot dodatno nalogo sem se, kot obljubljeno pri 4. nalogi, lotil obdelave posnetkov pulzarjev.

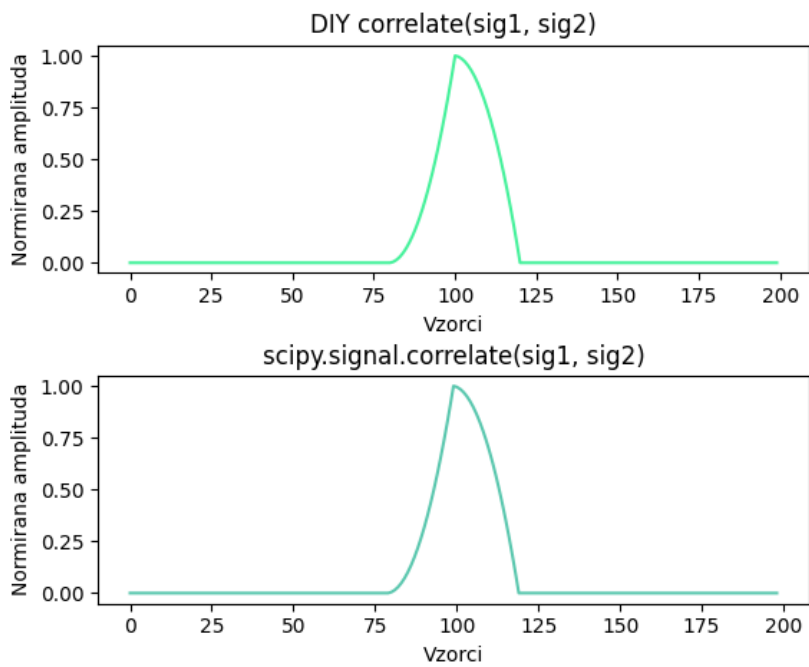
3 Opis reševanja

Kot je postala navada, sem se problema lotil v Pythonu spet z pomočjo vsestransko uporabnih paketov NumPy, SciPy in matplotlib. Prva dve knjižnici sem večinsko uporabljal za vse izračune in IO za .wav datoteke. Pogledal pa sem si tudi vgrajeno funkcijo `scipy.signal.correlate()`.

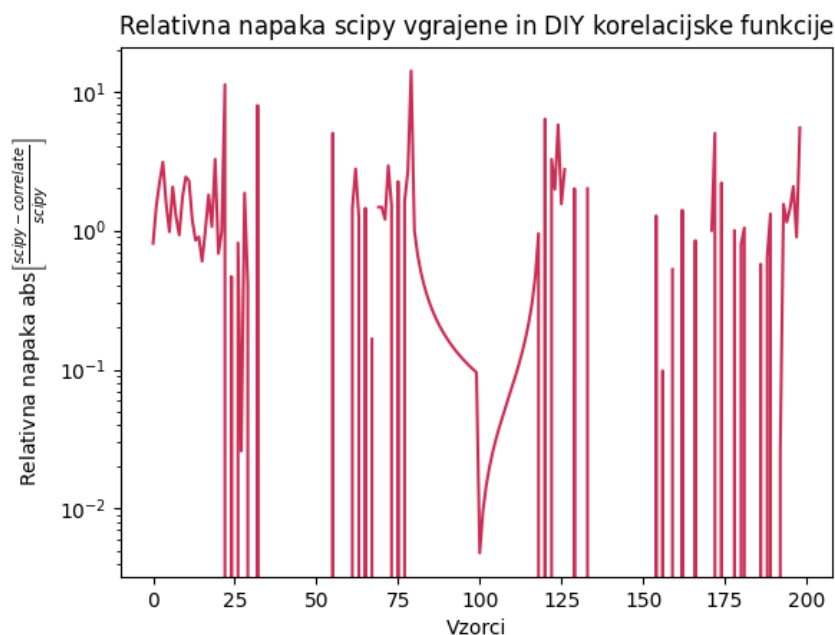
Napisal sem svojo funkcijo za korelacijo, ki deluje na metodo zero paddinga in inverzne Fourierove transformacije spektra moči. Želel sem preveriti, kako se primerja s prej omenjeno vgrajeno funkcijo. Zato sem zgeneriral dva tipična signala, ki se uporabljata za prikaz značilnosti konvolucije in korelacije, in primerjal relativno odstopanje glede na `scipy.signal.correlate()`.



Slika 1: Poskusna signala za korelacijo



Slika 2: Korelacija prejšnjih signalov



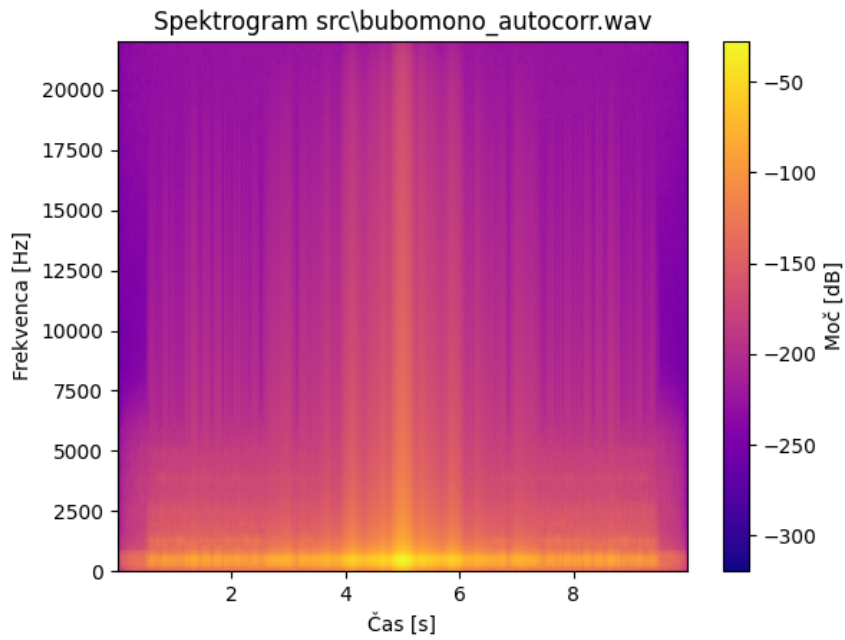
Slika 3: Relativna napaka med dvema metodama

Korelacija teh dveh testnih signalov nam da pričakovano funkcijsko obliko. Ne-pričakovana pa je natančnost, ki je v splošnem zadostna. V nekaterih točkah pa pravzaprav pride do kar velikih odstopanj. Mogoče bi bilo bolj smiselno računati napako glede na nenormirane amplitude. Pravzaprav nisem prepričan, kako to, da je pride do takega odstopanja. Sumim, da bi lahko šlo za kakšen izraz s smislu $\sim \frac{0}{0}$...

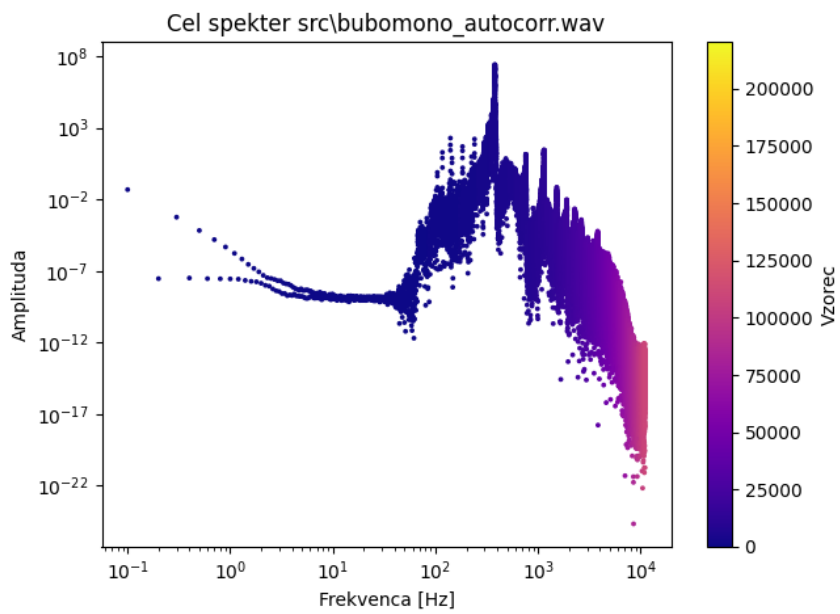
Tako sem se (nekoliko) prepričal v gotovost svoje funkcije za korelacijo in sem izračunal avtokorelacijske funkcije za vse podane signale. Tu sem pravzaprav nekaj kode za delo z .wav datotekami recikliral od prejšnje naloge, kar mi je prišlo zelo prav. Dobljeni grafi so že eden od glavnih rezultatov, ki jih zahteva naloga. Predstavljeni so v rezultatih 4.1

Zdelo se mi je zanimivo, da bi si pogledal spektre teh signalov bolj od blizu. Želel sem si narisati spektre in tudi spektrograme (torej spekter ob različnih časih), ki bi prišli še bolj prav za vizualizacijo podatkov pri obdelavi pulzarjev. Poskusil sem napisati svojo funkcijo za deljenje signala na intervale, da bi lahko narisal spekter ob različnih časih, ampak žal nisem prišel zelo daleč zaradi težav pri določanju intervalov. K sreči je za risanje spektrogramov vgrajena funkcija `matplotlib.pyplot.specgram()`, ki sama od sebe podatke razdeli na intervale za Fourierovo transformacijo.

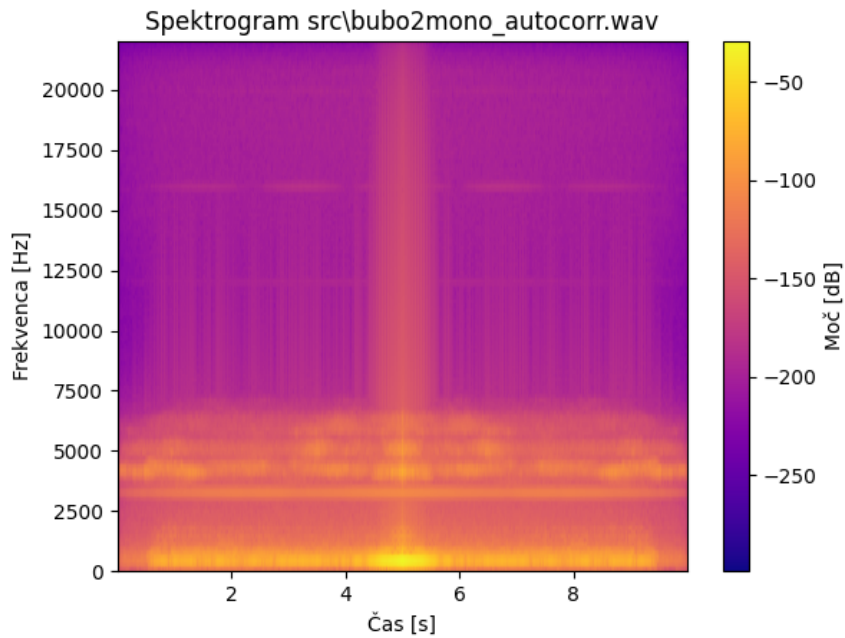
Tu predstavim primera za obe sovi. Zelo lepo se vidi osnoven oz. fundamentalen ton, ki ga sove zapojejo, kot pa tudi nekaj višjih harmonikov. Prva sova zapoje pri okoli 380 Hz, druga pa pri 335 Hz.



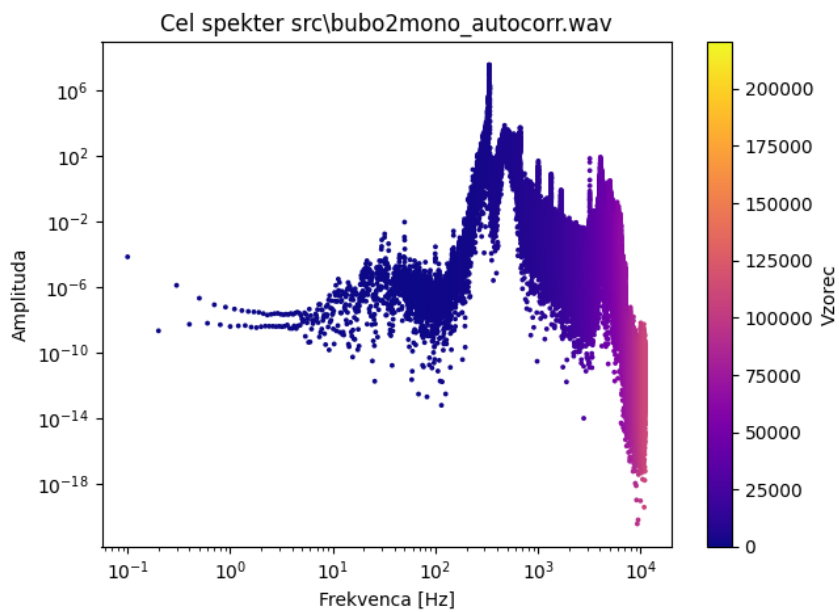
Slika 4: Spektrogram avtokorelacije signala bubomono.wav



Slika 5: Celoten spekter avtokorelacije signala bubomono.wav



Slika 6: Spektrogram avtokorelacije signala bubo2mono.wav

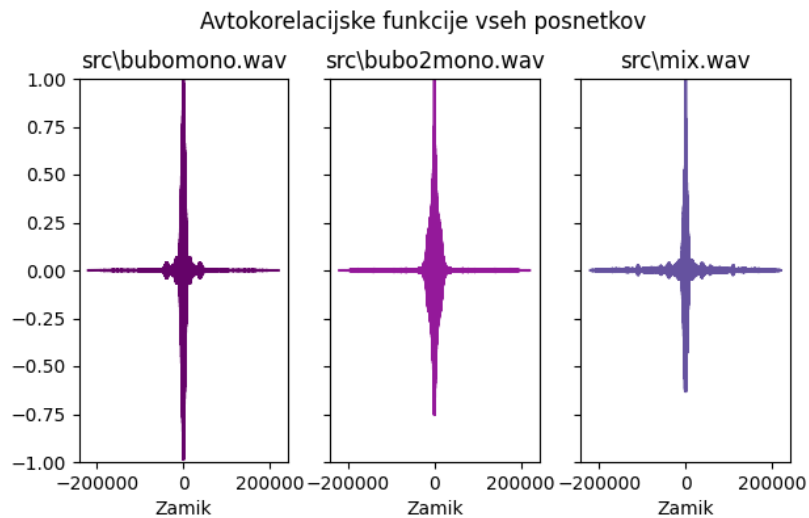


Slika 7: Celoten spekter avtokorelacije signala bubo2mono.wav

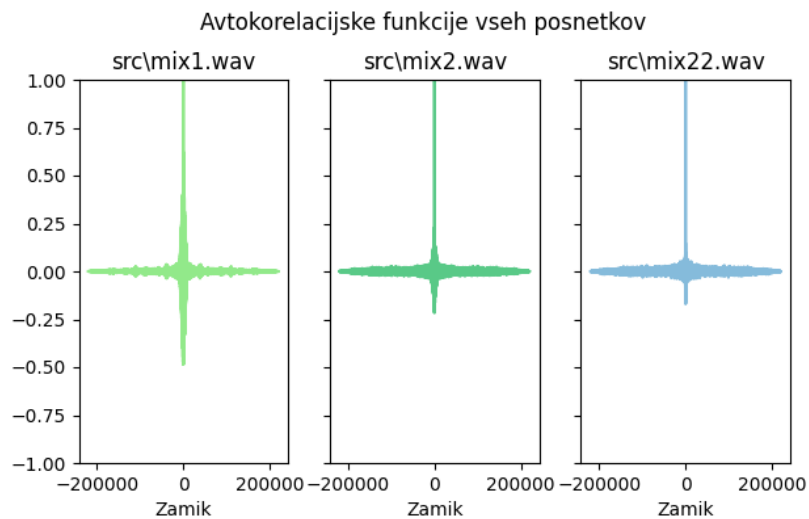
4 Rezultati

4.1 Avtokorelacijske funkcije vseh signalov

Lepo se vidi, da gre avtokorelacijska funkcija vedno bolj proti nekakšni delta funkciji, bolj kot je zašumljen signal. Za popolnoma naključen *beli šum* pričakujemo le delta funkcijo v izhodišču, ko ni zamika.



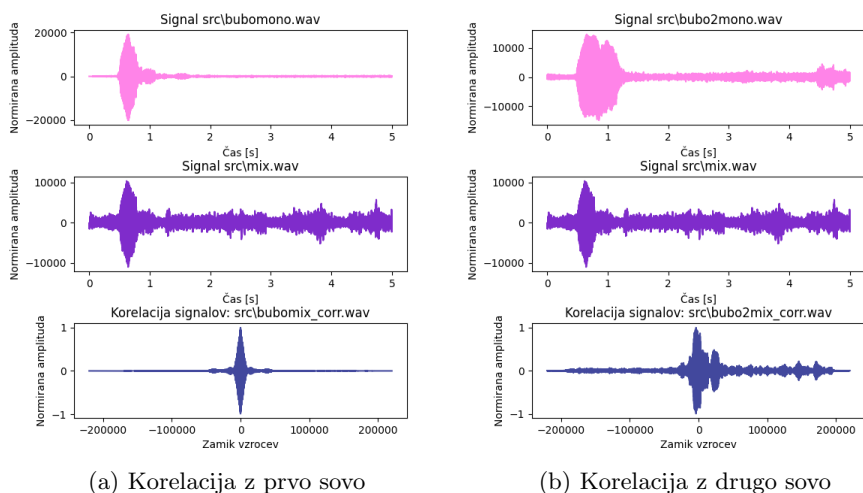
Slika 8: Avtokorelacijske funkcije signalov sov in dodanih čričkov



Slika 9: Avtokorelacijske funkcije z reko zašumljenih signalov

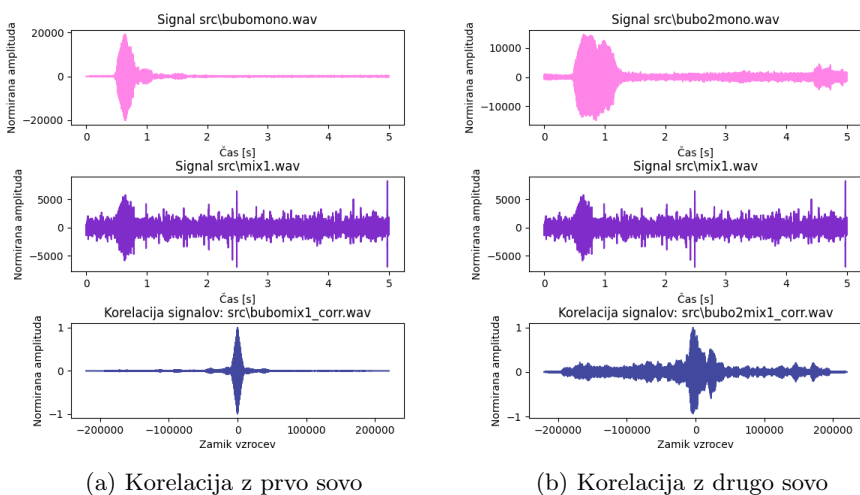
4.2 Zašumljeni posnetki

Z opazovanjem korelacijske funkcije lahko ugotovimo podobnosti signalov. Tako lahko učinkovito vidimo/slišimo skozi šum, kar pride zelo prav, ker so podatki v resničnosti pogosto zašumljeni.



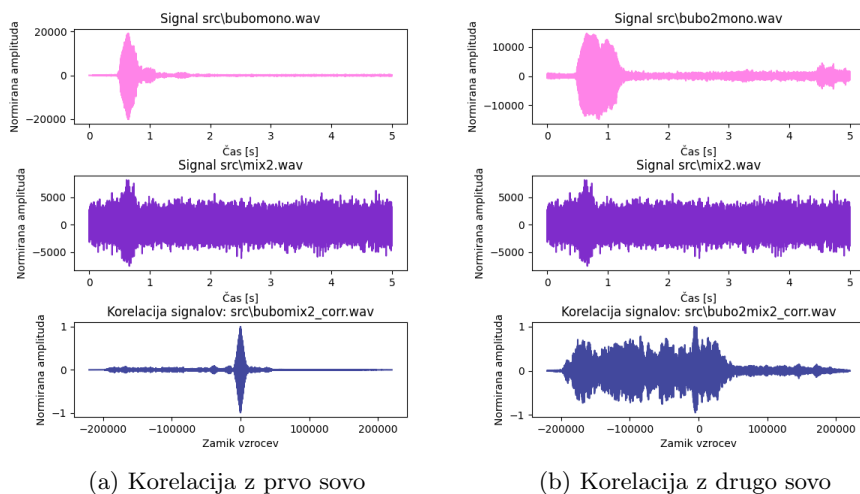
Slika 10: Korelacija z `mix.wav`

Jasno se vidi, da se v prvem s črčki zašumljenem posnetku sliši **prvo** sovo, ker je korelacijska funkcija podobna avtokorelacijski. Ta signala sta torej podobna. To lahko preverimo tudi če poslušamo oba posnetka in se prepričamo, da drži. V drugem zašumljenem posnetku, kjer je še potoček, po istem argumentu prepoznamo oglašanje **prve** sove.



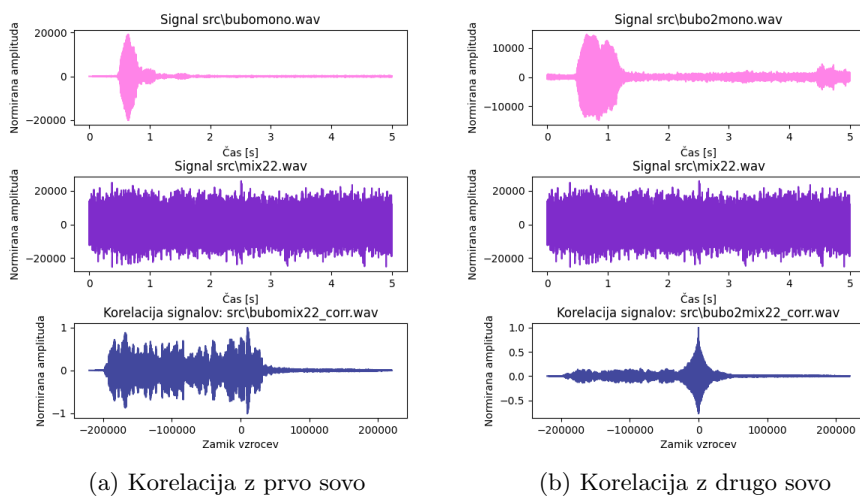
Slika 11: Korelacija z `mix1.wav`

Tretji posnetek je zašumljen z večjo reko. Tudi v njemu lahko prepoznamo oglašanje **prve** sove preko korelacijske funkcije.



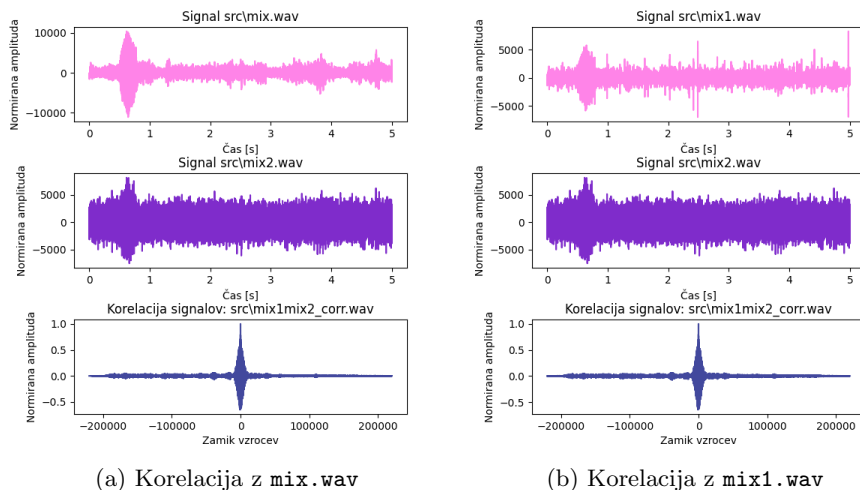
Slika 12: Korelacija z `mix2.wav`

Zadnji posnetek pa je zašumljen do te mere, da se sovo komaj sliši. Kljub temu preko korelacije uspemo “slišati” skozi šum in prepoznamo, da se oglašja **druga** sova.



Slika 13: Korelacija z `mix22.wav`

Praktično je pravzaprav tudi to, da sploh ni potrebno imeti izoliran zvok oglašanja sove, da lahko vse to ugotovimo. Če vzamemo s potočkom zašumljen posnetek lahko izračunamo korelacijo z tistim, kjer je zraven še reka. Signala vsebujeta isto oglašanje sove, torej sta si podobna. Pri prisotnem potočku pa ni problem že brez manipulacije zvokov ugotoviti za katero sovo gre. Enako velja posnetek s črički, kjer je to še lažje.

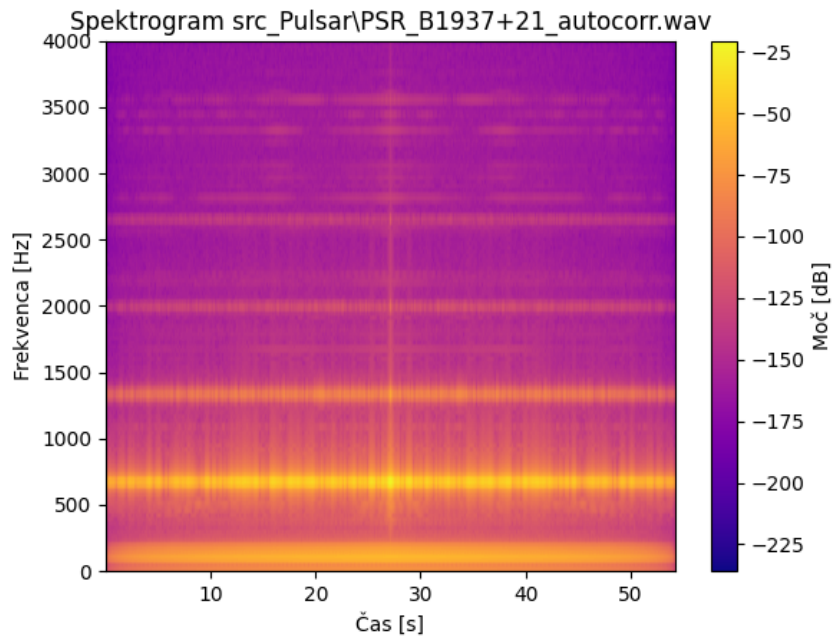


Slika 14: Korelacija z mix2.wav

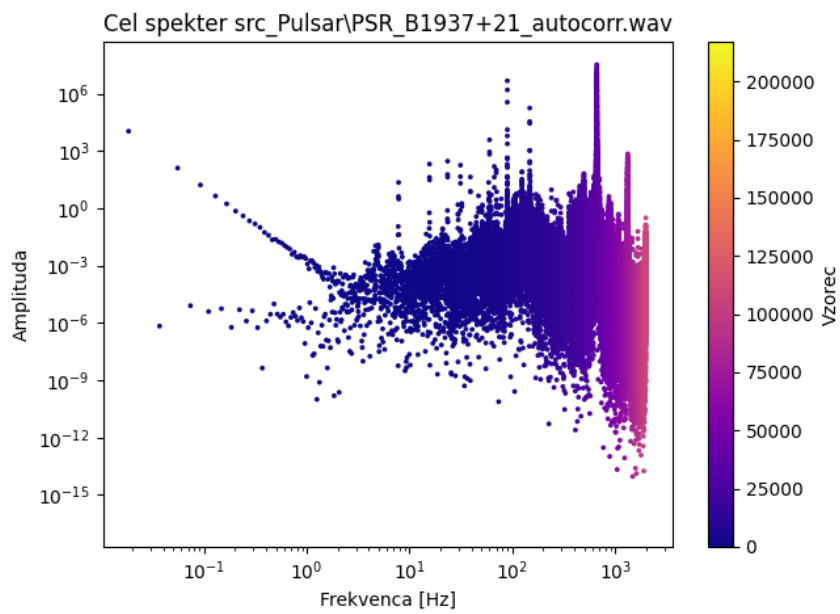
4.3 Pulzarji

Kot obljubljeni v prejšnji nalogi sem se lotil obdelave posnetkov pulzarjev. Pulzar, skovanka iz angleščine iz *pulsating radio source*, je astrofizikalni kompaktni objekt (tipično nevtronska zvezda, lahko pa tudi bela pritlikavka) z močnim magnetnim poljem. Pulzarji sevajo elektromagnetno sevanje iz svojih magnetnih polov. Njihov najbolj preprost model je, če si predstavljamo, da tvorijo polje kot neke sorte magnetni dipol. Kompaktni objekti so posmrtni ostani zvezd. Nevtronske zvezde so velike le v okolici 10 km, kar pomeni, da se zaradi ohranjanja vrtilne količine, zaradi ohranjanja vrtilne količine, ko se skrčijo iz velikosti zvezde le na nekaj kilometrov. Če so žarki usmerjeni proti nam pravzaprav detektiramo utripanje, kot utripanje nekega svetilnika. Te signale se opazuje z pomočjo radijskih teleskopov.

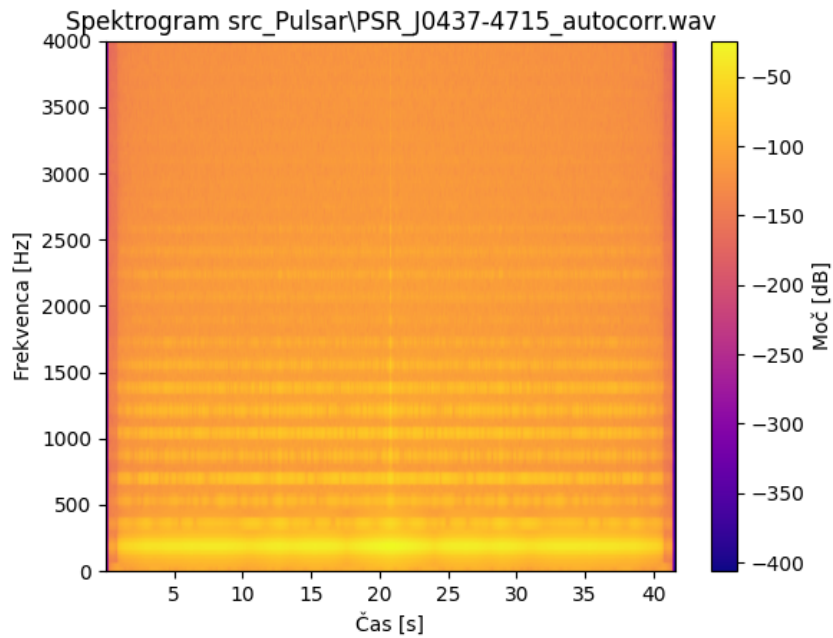
Na internetu sem našel nekaj posnetkov pulzarjev in njihovih frekvenc. Posnetki so večinoma zašumljeni, zato se mi je zdel zanimiv izziv poskusiti odpraviti šum preko avtokorelacije. Izbral sem si zelo hitra pulzarja **PSR B1937+21**, ki se vrti z frekvenco 642 Hz in **PSR J0437-4715**, ki se vrti z frekvenco 174 Hz. Pri obeh pulzarjih so bili dobljeni podatki gotovo že do neke mere počiščeni. Zelo lepo se vidi fundamentalni oz. osnovni ton in potem višje harmonike. Sem zadovoljen z rezultati. O tem konceptu sem želel govoriti že zadnjič, ko sem omenjal delo z sintesizerji. Še posebej tistimi, ki delujejo na princip frekvenčne modulacije.



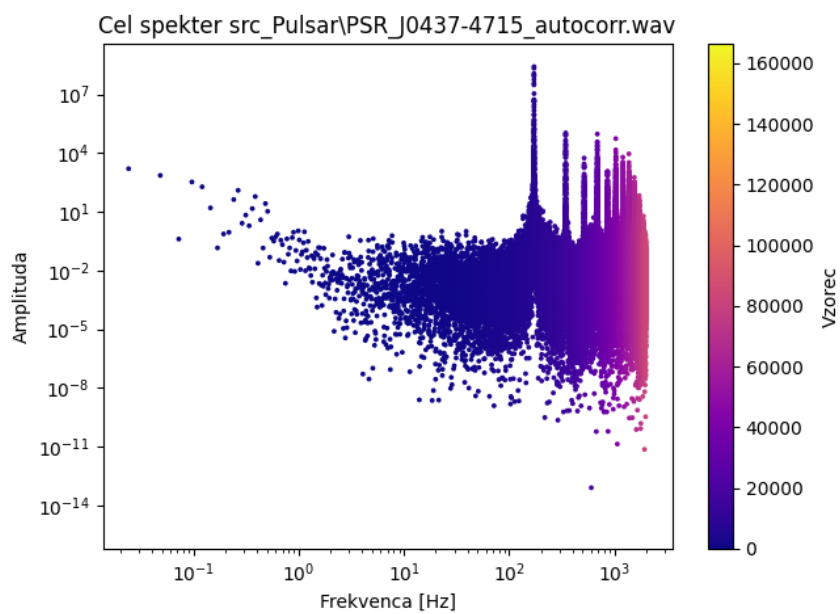
Slika 15: Spektrogram PSR J0437-4715



Slika 16: Cel spekter PSR J0437-4715



Slika 17: Spektrogram **PSR B1937+21**



Slika 18: Cel spekter **PSR B1937+21**

5 Komentarji in izboljšave

Zdi se mi, da sem večino pomembnih komentarjev že napisal sproti. Dolgo časa sem potreboval, da sem ustvari delujočo funkcijo za korelacijo. Največjo težavo pa sem imel z smiselnim prikazom podatkov, ker imam neuporabljenih grafov in slik veliko. Nisem želel samo lepiti en graf za drugim v .pdf, ker se mi to ne zdi ravno smiselno. Vseeno pa sem nekaj lepših plotov predstavil na tak način. Za dodatek sem si pogledal tudi kako se sliši signal po korelaciji tako, da sem ga spet izvozil kot .wav datoteka. (Več o tem v 5.1). Prav s tem razlogom so vse amplitude normirane, kar sem se hitro naučil, da je nujno potrebno, če ne želiš uničiti svojih ušes in zvočnikov. Priznam, da sam kar nekaj časa metal posnetke različnih stvari skozi program in užival v čudnih rezultatih.

Navdušen sem tudi nad rezultati, ki sem jih dobil pri obdelavi pulzarjev. Zdi se mi zelo dober prikaz aplikativne rabe predstavljene metode. Prejšnji teden sem poskusil podobno ampak mi zaradi šuma ni uspelo.

5.1 DSP in audio efekti

Naloga me je nekoliko navdihnila tudi za DSP (angl. *Digital Signal Processing*). Pri eni bolj zgodnji pokvarjeni verziji funkcije za korelacijo sem ugotovil, da pravzaprav ustvarja čudovite ambientalne teksture. Tudi pravilno napisana korelacija sama lahko nekoliko razmaže signale, kar je meni kot navdušencem nad (med drugim) noise rock-om/shoegaze zelo všeč. Vsekakor prilagam v email-u tudi nekaj posnetkov in njihovih izvornih datotek za primerjavo. Dobil sem idejo, da bi to lahko poskusil spremeniti v real time digitalen audio efekt za svoje inštrumente.. Ko bi le imel več časa, pa bi posnel shoegaze pesem o FFT za mafijski praktikum...



Slika 19: Potencialni album cover za FFT shoegaze

Literatura

- [1] Wikipedia. Wiener–Khinchin theorem — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Wiener%E2%80%99Khinchin%20theorem&oldid=1009093078>, 2021. [Online; accessed 14-November-2021].