

Univerza v Ljubljani  
Fakulteta za *matematiko in fiziko*



# NALOGA 4: HITRA FOURIEROVA TRANSFORMACIJA IN KORELACIJSKE FUNKCIJE

MATEMATIČNO-FIZIKALNI PRAKTIKUM

DIMITRIJE PEŠIĆ

VPISNA ŠTEVILKA: 28201072

PREDAVATELJ: PROF. DR. BORUT PAUL KERŠEVAN

LJUBLJANA, 10.10.2023

## 1 Uvod

Naloga je od nas zahtevala, da izračunamo avtokorelacijsko funkcijo signalov, ki so močno zašumljeni z raznimi zvoki narave. Mi smo poskušali iz njih razbrati za katero sovo gre v z uporabo dveh nezašumljenih posnetkov.

Dodatna naloga je pa zahtevala uporabo avtokorelacije pri kakšnem drugem, poljubnem primeru.

## 2 Ozadje

Diskretno Fourierovo transformacijo smo definirali kot

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

oziroma

$$H_k = \sum_{n=0}^{N-1} W_N^{nk} h_n, \quad W_N = \exp(2\pi i / N).$$

Ta postopek ima očitno časovno zahtevnost  $N^2$ . Račun pa je mogoče izvesti tudi z bistveno manj operacijami. Osnovni premislek je razcep

$$H_k = H_k^{\text{sod}} + W_N^k H_k^{\text{lih}},$$

kjer smo transformiranko  $H$  izrazili s transformacijama njenih sodih in lihih členov, pri čemer je vsota vsake od transformacij zdaj dolžine  $N/2$ . Gornjo relacijo lahko uporabljamo rekurzivno: če je  $N$  enak potenci števila 2, lahko rekurzijo razdrobimo do nizov, ki imajo samo še en člen. Zanj je transformacija identiteta. Za obrat pri eni vrednosti frekvence (pri danem  $m$ ) je potrebno na vsakem koraku rekurzije le eno množenje s potenco  $W$ , korakov pa je  $\log_2 N$ . Skupna časovna zahtevnost je torej le še  $N \log_2 N$ .

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  $W$  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 (FFT).

Z neko transformacijo iz družine FFT bomo izračunali korelacijsko funkcijo dveh signalov. Korelacija periodičnih funkcij  $g(t)$  in  $h(t)$  s periodo  $T$  je definirana kot:

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

oziroma diskretno

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

Računamo torej skalarni produkt funkcij, ki sta časovno premaknjeni za  $\tau$  oziroma  $n$ . Če je za določeno vrednost premika ta funkcija višja kot v okolici, potem to pomeni, da sta si funkciji podobni, le da ju je treba premakniti, da se to vidi.

V primeru, da sta funkciji (signala), ki ju primerjamo, enaki, računamo njuno *avtokorelacijsko funkcijo*: ta je mera za to, ali signal ostaja s pretekanjem časa sam sebi podoben. Če je signal slabo koreliran (sam s sabo), korelacija  $\varphi_{hh}(n)$  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. Še bolj nas zanima, kako *hitro* se korelacija izgublja: računamo rajši reskalirano obliko avtokorelacije

$$\tilde{\varphi}_{hh}(n) = \frac{\varphi_{hh}(n) - \langle h \rangle^2}{\varphi_{hh}(0) - \langle h \rangle^2},$$

kjer je imenovalec nekakšno merilo za varianco signala,

$$\sigma^2 = \varphi_{hh}(0) - \langle h \rangle^2 = \frac{1}{N} \sum_{k=0}^{N-1} (h_k - \langle h \rangle)^2.$$

Pri zgornjih enačbah moramo še “peš” poskrbeti za periodično zaključenost signala pri  $n = N$ , torej da je perioda enaka velikosti vzorca. Če tega ne moremo narediti, je bolj pravilna definicija avtokorelacije

$$\varphi_{hh}(n) = \frac{1}{N-n} \sum_{k=0}^{N-n-1} h_{k+n} h_k.$$

Praktičen račun po zgornji formuli lahko postane za velike vzorce prezauden. Avtokorelacijo rajši računamo s FFT (DFT)  $\mathcal{F}$ , saj je korelacija obratna Fourierova transformacija  $\mathcal{F}^{-1}$  produkta Fourierovih transformacij  $\mathcal{F}$ , torej z  $G = \mathcal{F}g$  in  $H = \mathcal{F}h$  dobimo

$$\varphi_{gh}(n) = \frac{1}{N-n} \mathcal{F}^{-1} [G \cdot (H)^*]$$

oziroma

$$\varphi_{hh}(n) = \frac{1}{N-n} \mathcal{F}^{-1} [|H|^2].$$

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

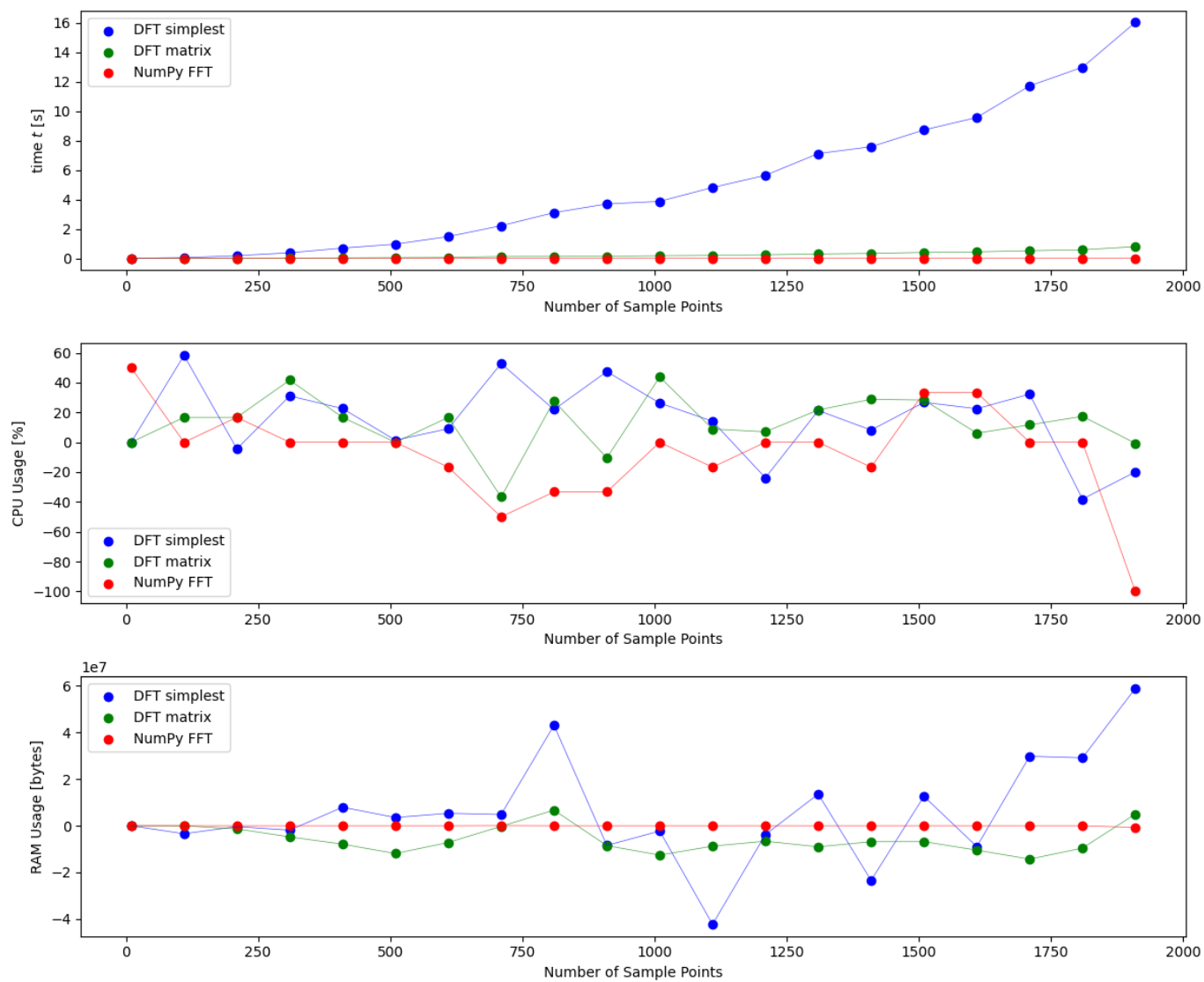
$$\varphi_{hh}(n) = \frac{1}{N-n} \sum_{k=0}^{2N-1} \tilde{h}_{k+n} \tilde{h}_k,$$

kar lahko izračunamo s FFT.

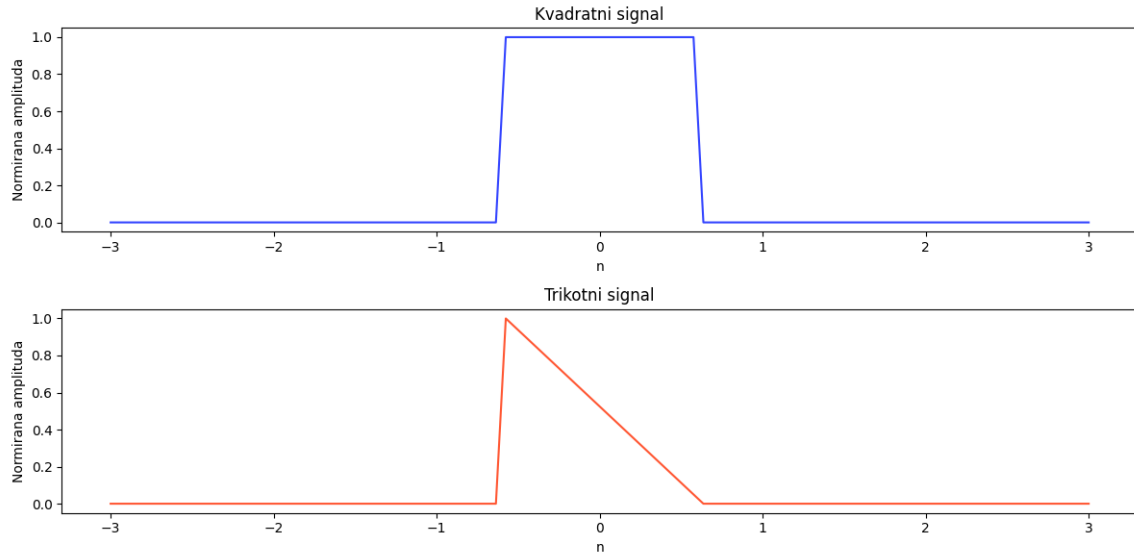
## 3 Implementacija

### 3.1 Hitrost

Računanje avtokorelacijske funkcije sem vpeljal z uporabo hitre Fourierove transformacije, saj sem že pri prejšnji nalogi prikazal nepraktičnost uporabe diskretne Fourierove transformacije. Prilagam isti graf 1 kjer sem testiral hitrost izvajanje dveh implementaciji DFT in implementacijo `numpy.fft.fft()` funkcije, poleg pa še porabo pomnilnika in procesorja v odvisnosti od števila vzorčenj.



Slika 1: Čas izvajanja, poraba procesorja in pomnilnika.



Slika 2: Vhodni kvadratni in trikotni signal.

### 3.2 Primerjava metod

Primerjal sem več različnih implementacij funkcij za izračun korelacij med signali. implementiral sem funkcijo 1

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

brez uporabe knjižnic samo z pythonovimi zankami. Nato sem z uporabo numpyjevih vektorskih operacij naredil funkcijo, ki je prvo vhodnemu signalu dodala zero-padding kot smo to omenjali pri prejšnji nalogi, nato pa izvedla operacijo opisano v formuli 2

$$\varphi_{gh}(n) = \frac{1}{N} \mathcal{F}^{-1} [G \cdot (H)^*], \quad (2)$$

v naslednjem poglavju pokažem zakaj sem uporabil normalizacijo  $\frac{1}{N}$  namesto  $\frac{1}{N-n}$ .

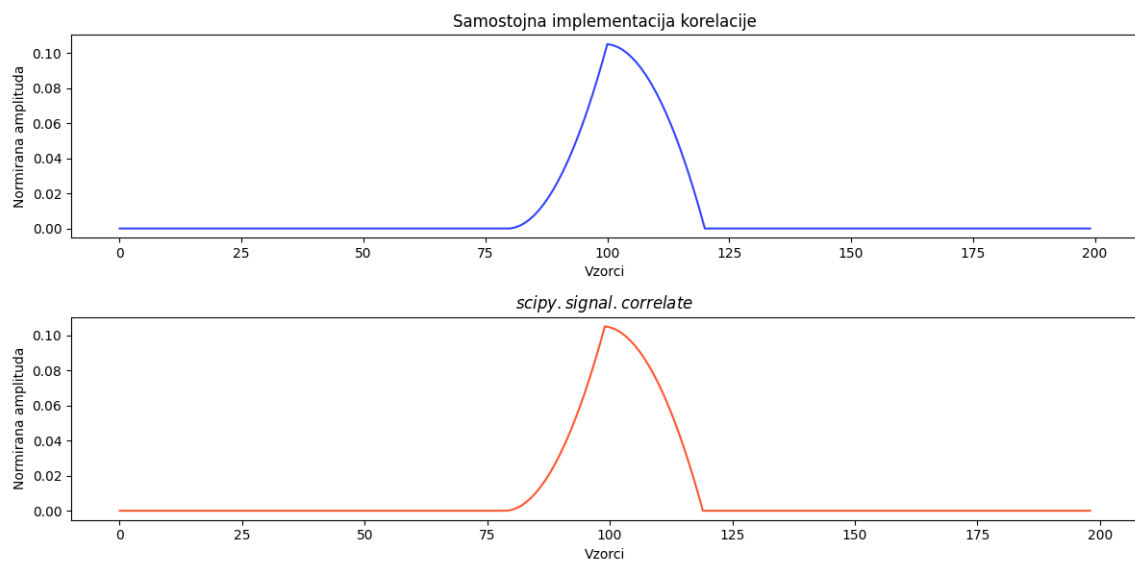
Za konec sem pa še implementiral kar standardno funkcijo iz numpy knjižnice, in sicer `numpy.correlate`, ki vzame dva vhodna signala in izračuna korelacijsko funkcijo.

Zgledoval sem se večinoma po predstavitve s predavanj in pa Wikipedia strani [5], zato sem si zadal cilj reproducirati graf z Wikipedije, in sicer na grafu 2 sta prikazana kvadratni in trikotni signal, na katerem izvednotim korelacijo z samostojno implementirano korelacijsko funkcijo in pa z `numpy.correlate`. To je pa prikazano na grafu 3.

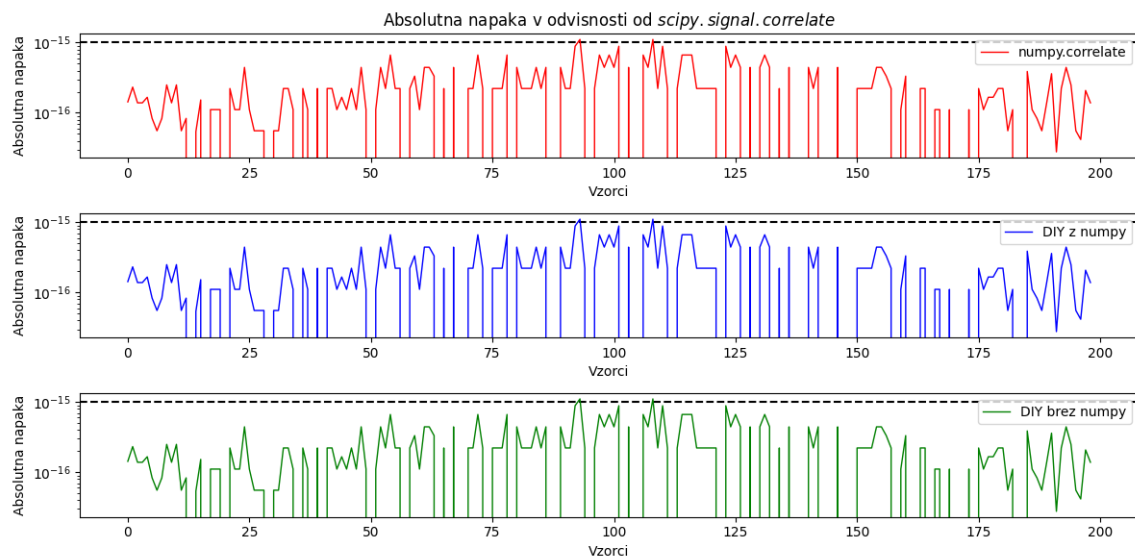
Izračunal sem še absolutno napako teh treh metod v odvisnosti od neodvisne implementacije iz scipy knjižnice, in sice funkcijo `scipy.signal.correlate`, kar je prikazano na grafu 4

### 3.3 Normiranje

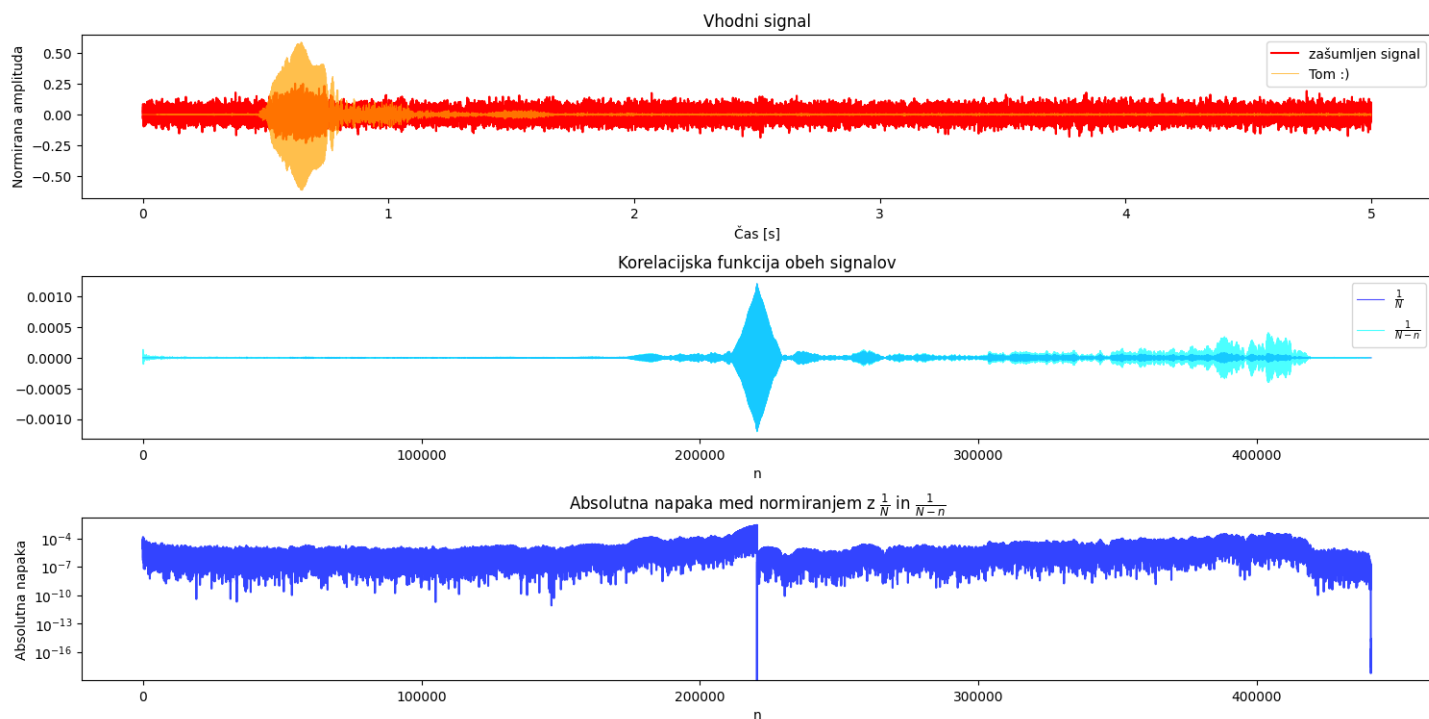
Preveril sem tudi razliko med normiranjem z  $\frac{1}{N}$  ali z  $\frac{1}{N-n}$ , to sem kar implementiral v funkciji 2. Za signal sem vzel zvok sove, ki jo v naslednjem poglavju poimenujem Tom in zašumljen signal te sove v okolici. Rezultat prikazan na grafu 5. Kot vidimo nastanejo takšni artefakti na koncih korelacijske funkcije pri uporabi normalizacije  $\frac{1}{N-n}$ . Za razliko pa pri normalizaciji  $\frac{1}{N}$  tega ni, zato sem naprej koristil to implementacijo.



Slika 3: Korelacija kvadratnega in trikotnega signala.



Slika 4: Absolutna napaka implementiranih metod v odvisnosti od *scipy.signal.correlate*



Slika 5: Različna normalizacija.

## 4 Detekcija sov

Zdaj ko imamo vse potrebno že implementirano, se lahko lotimo detekcijo sov v zašumljenih signalih. Za zabavo sem sovi poimenoval po slavnem duetu - **Tom Jerry**, ime mačkona Tom se nanaša na datoteko *bubomono.wav*, ime miši Jerry pa na datoteko *bubo2mono.wav*. Dobili smo še 4 različno močno zašumljene signale narave, iz katerih smo morali izluščiti ali gre za Toma ali Jerryja.

Na graf 6 sem narisal korelacijo med signalom Tom in Jerry, malo za zabavo, malo pa za boljšo predstavbo kaj pomeni slabo koreliran signal. Čeprav sta si signala dokaj podobna, saj gre za oglašanje dveh sov, nam graf nakazuje da to ni prava avtokorelacija, saj ne gre za isto sovo.

### 4.1 Tom

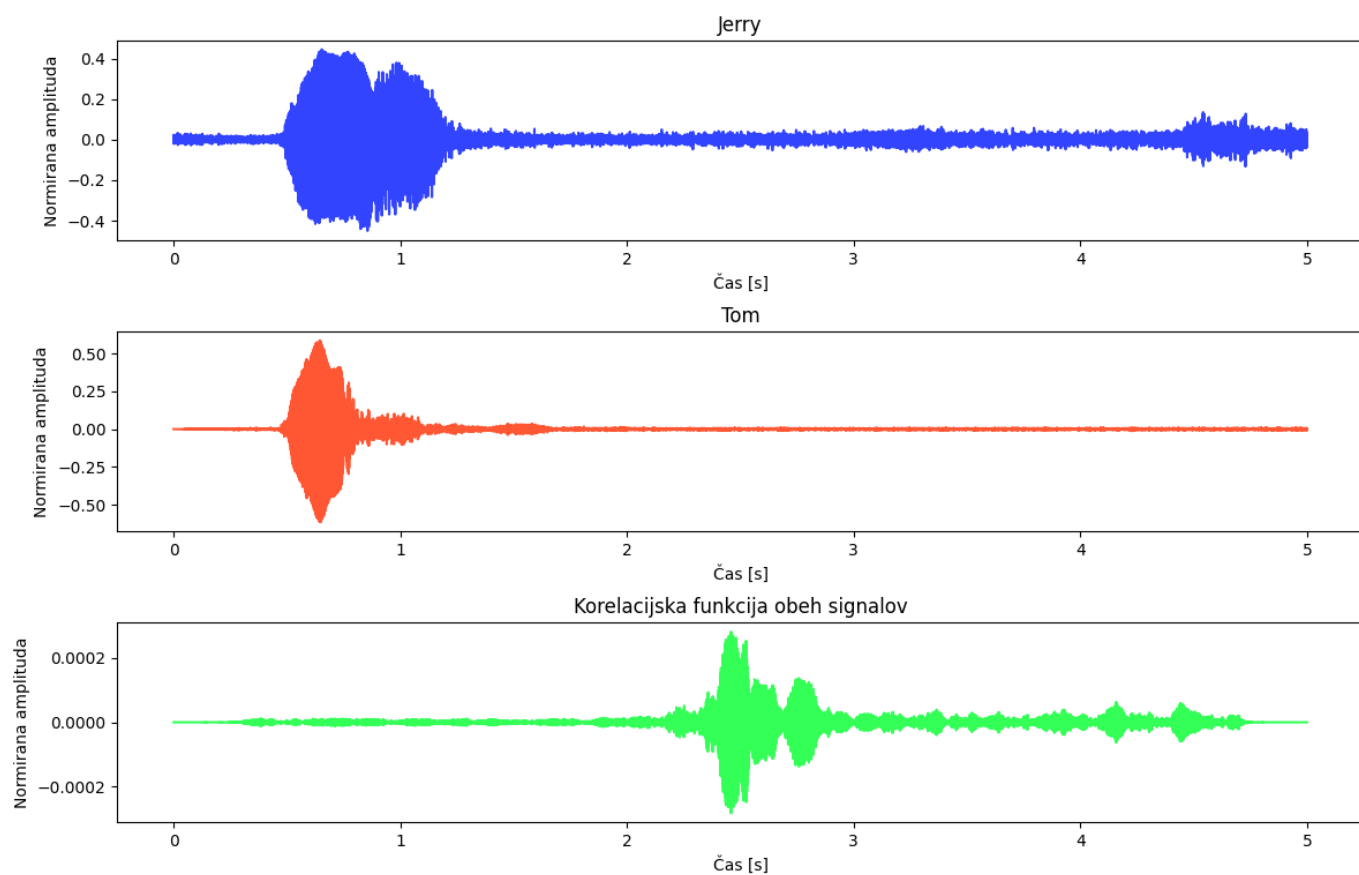
V tem podpoglavju se nahajajo grafi korelacij vseh signalov z signalo našega mačkona Toma. To so grafi 7, 8, 9 in 10.

Opazimo lahko, da se Tom oglaš (verjetno zaradi vse strašne bolečine, ki mu jo je povzročil Jerry) na vseh posnetkih razen na zadnjem 10.

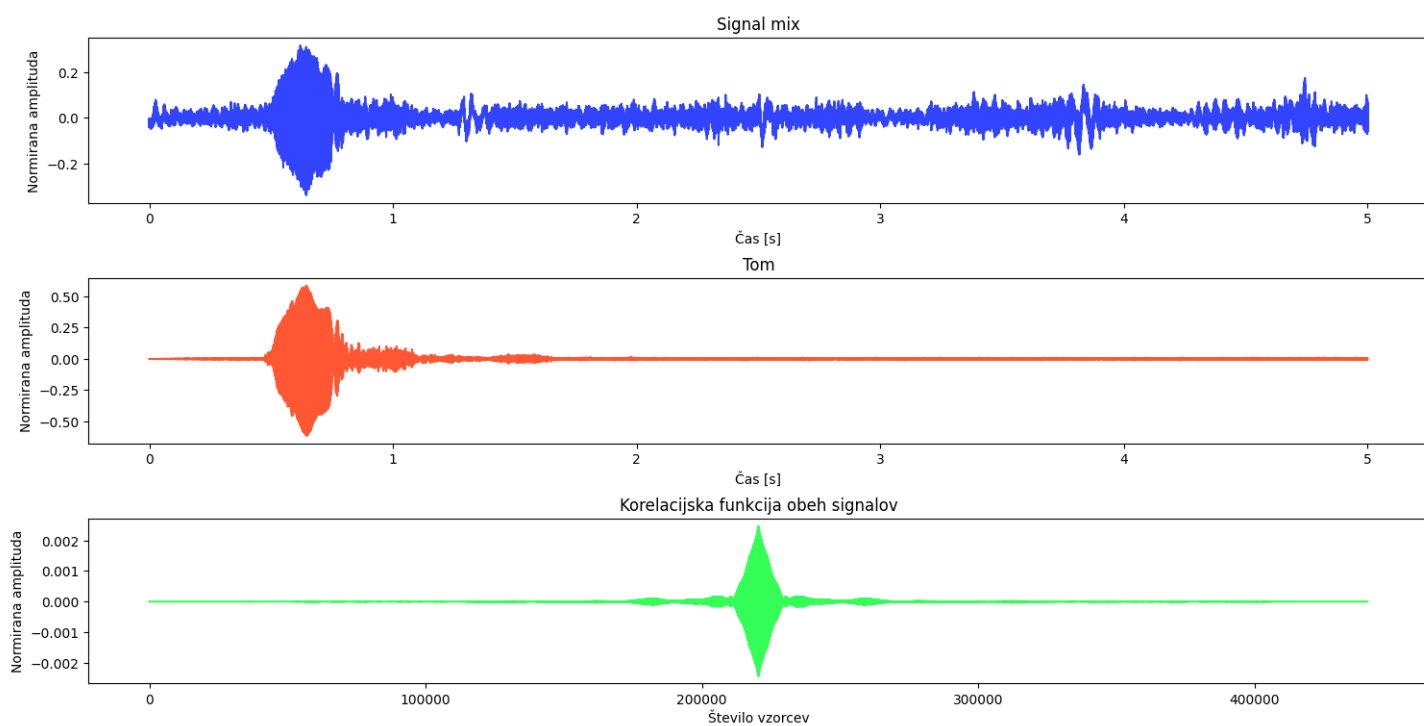
### 4.2 Jerry

Tako kot za Toma sem za Jerryja naredil korelacijo z vsemi signali, kar je prikazano na grafih 11, 12, 13 in 14.

Kot v seriji, Jerry je tiho in le poredko spusti kakšen zvok iz sebe, zato ga zaznamo le na zadnjem posnetku 14. Seveda se tako kot pri Tomu nahajajo tudi takšni 'ostanki' korelacije, vendar to ni zares avtokorelacija kot smo prikazali pri korelaciji signalov Toma in Jerryja skupaj. Tu se pokaže prava moč korelacijskih funkcij za prepoznavanje in klasifikacijo raznih neznanih, zelo zašumljenih signalov in s tem izluščevanje informacij o le teh.

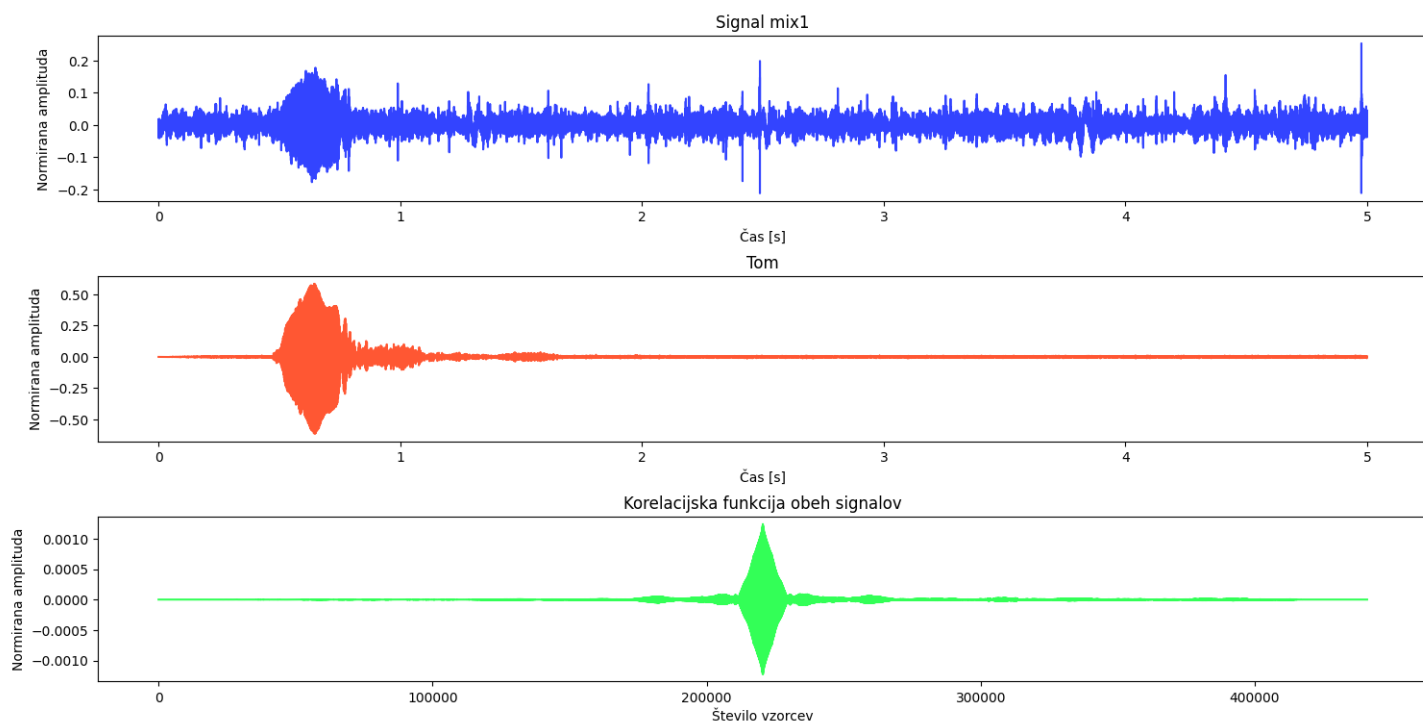


Slika 6: Korelacija med signalom Toma in Jerryja.

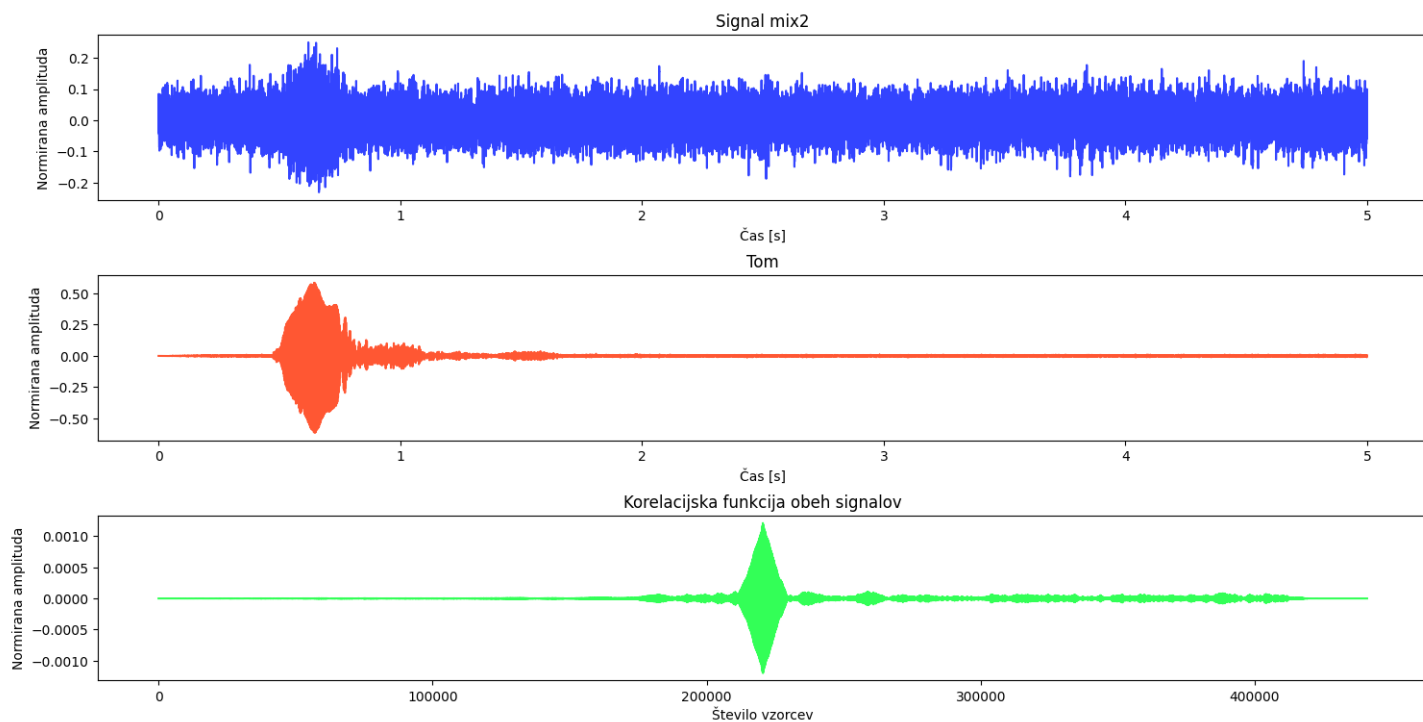


Slika 7: Korelacija med signalom Toma in zašumljenega signala narave mix.wav.

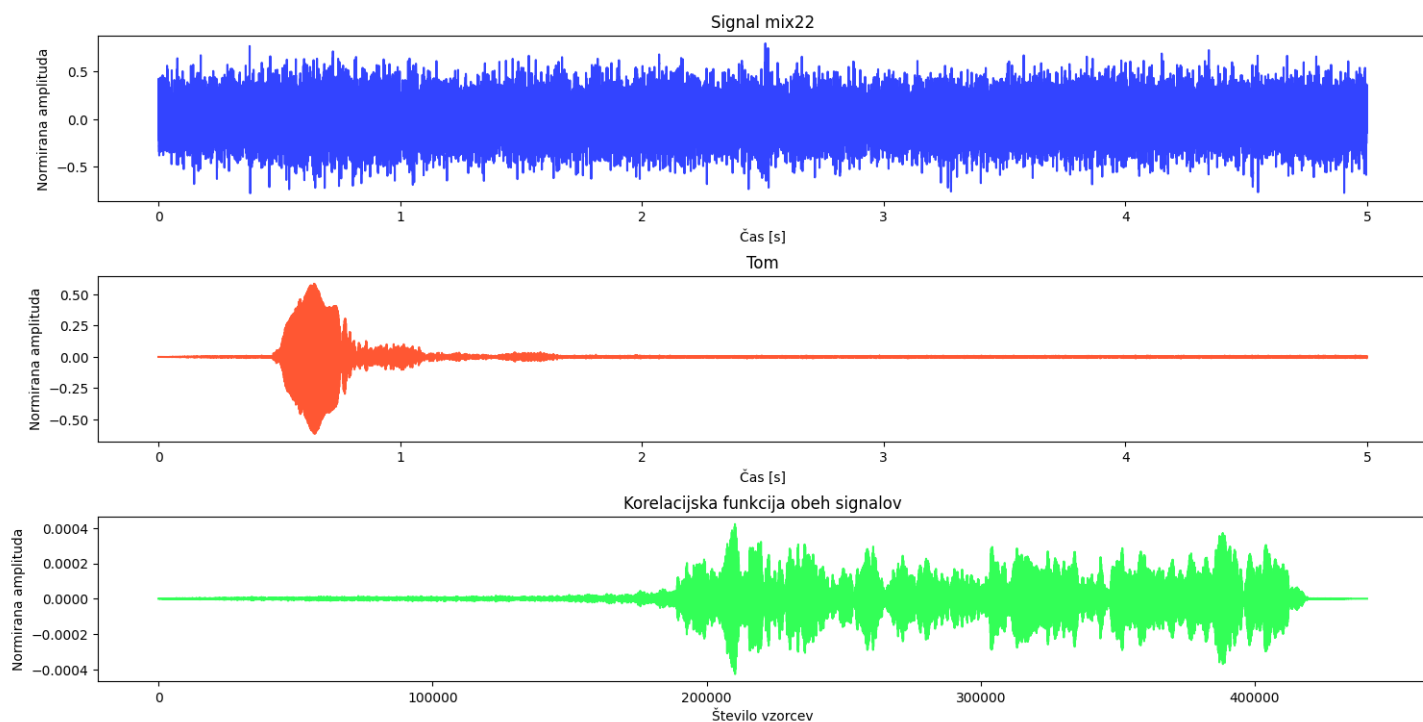




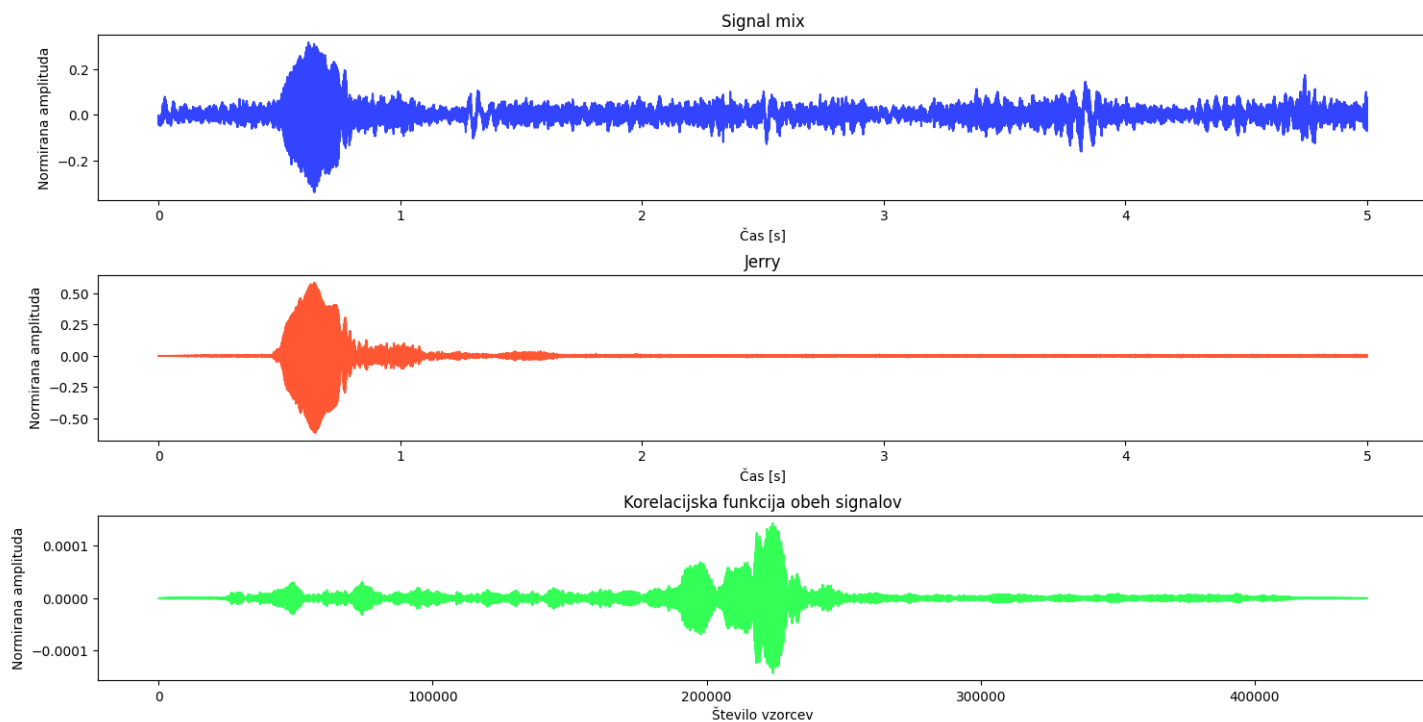
Slika 8: Korelacija med signalom Toma in zašumljenega signala narave mix1.wav.



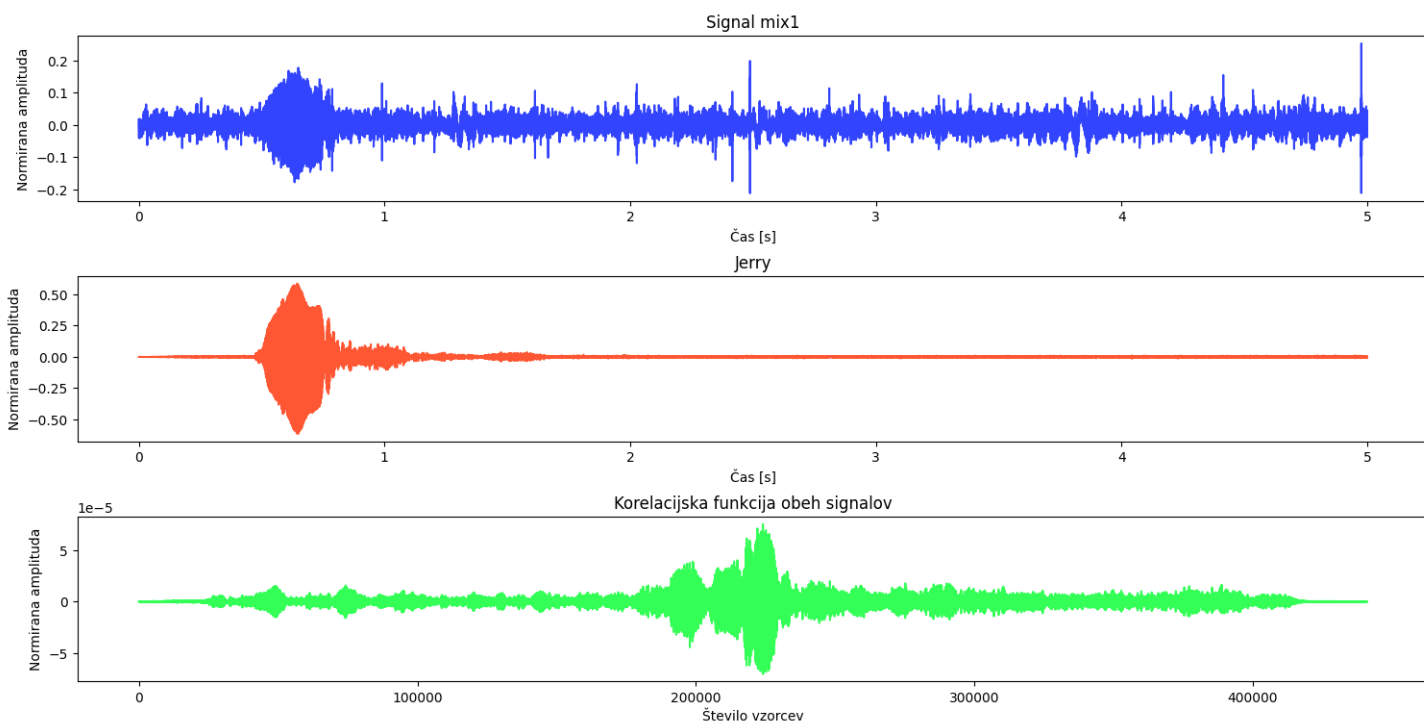
Slika 9: Korelacija med signalom Toma in zašumljenega signala narave mix2.wav.



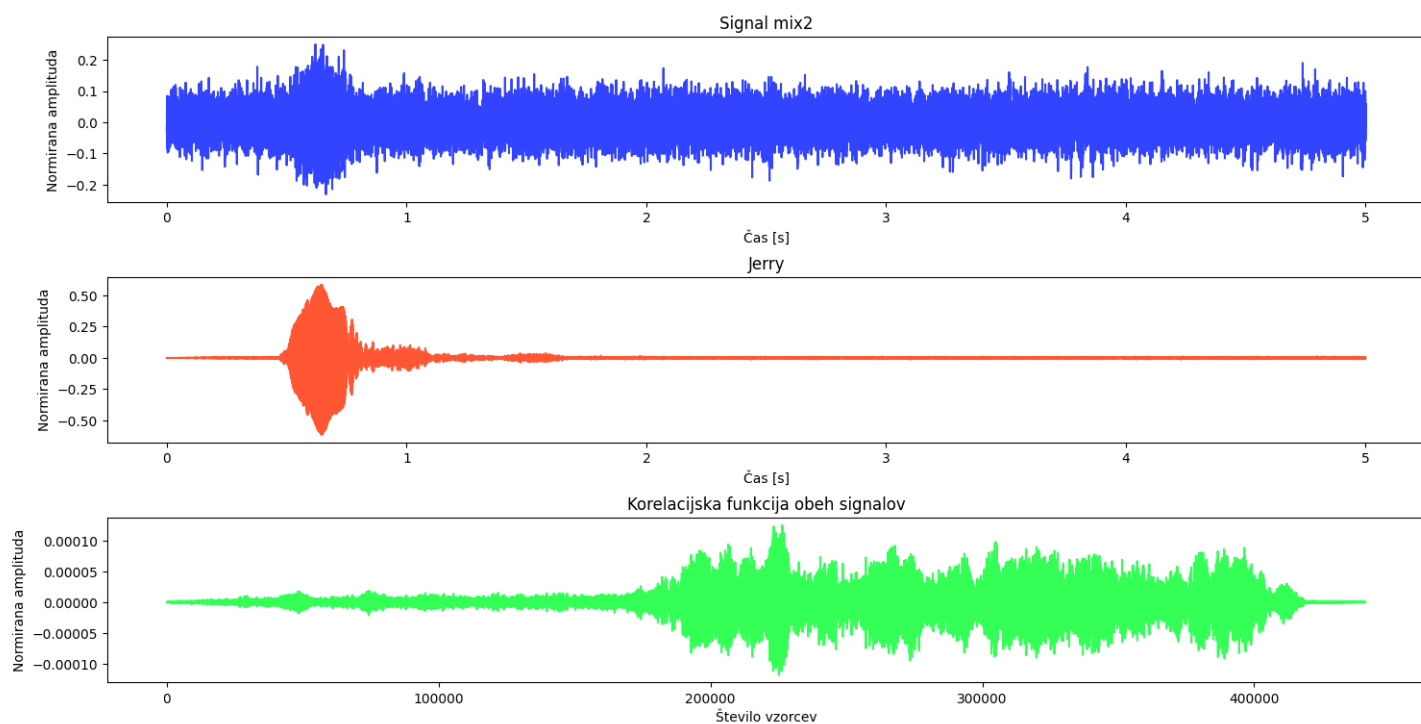
Slika 10: Korelacija med signalom Toma in zašumljenega signala narave mix22.wav.



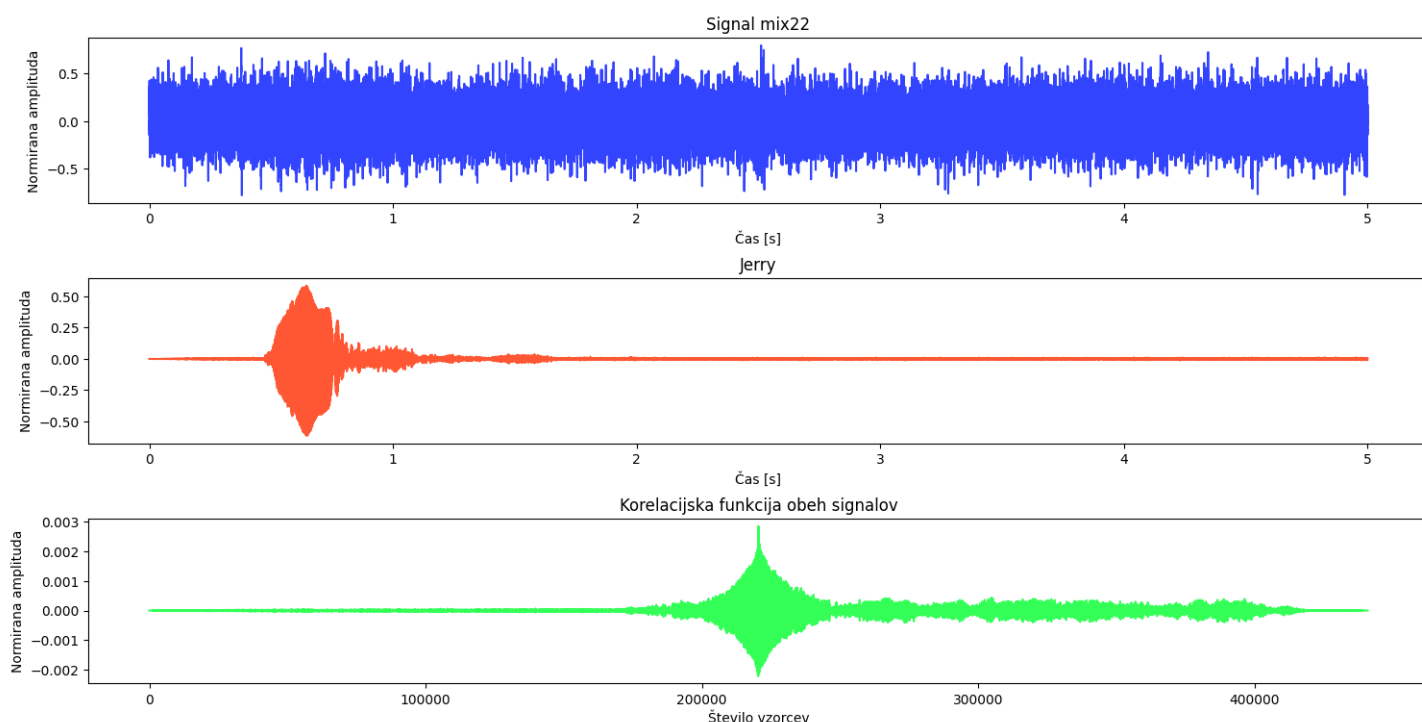
Slika 11: Korelacija med signalom Jerryja in zašumljenega signala narave mix.wav.



Slika 12: Korelacija med signalom Jerryja in zašumljenega signala narave mix1.wav.



Slika 13: Korelacija med signalom Jerryja in zašumljenega signala narave mix2.wav.



Slika 14: Korelacija med signalom Jerryja in zašumljenega signala narave mix22.wav.

## 5 Dodatna naloga

Kot sem namignil pri prejšnji nalogi, je uporaba Fourierove analize razširjena v razna področja. Odločil sem se prikazati uporabo Fourierove analize in avtokorelacijskih funkcij v financah, saj me tudi to področje zanima (žal smo tudi fiziki občasno denarno motivirani).

Avtokorelacijske funkcije omogočajo vpogled v finančne podatke. So orodje za oceno zaporedne korelacije, pomagajo nam določiti, ali pretekla uspešnost sredstva vpliva na njegove prihodnje donose. Pozitivna avtokorelacija kaže na vztrajnost trendov, medtem ko negativna avtokorelacija nakazuje na povprečje.

### 5.1 Implementacija

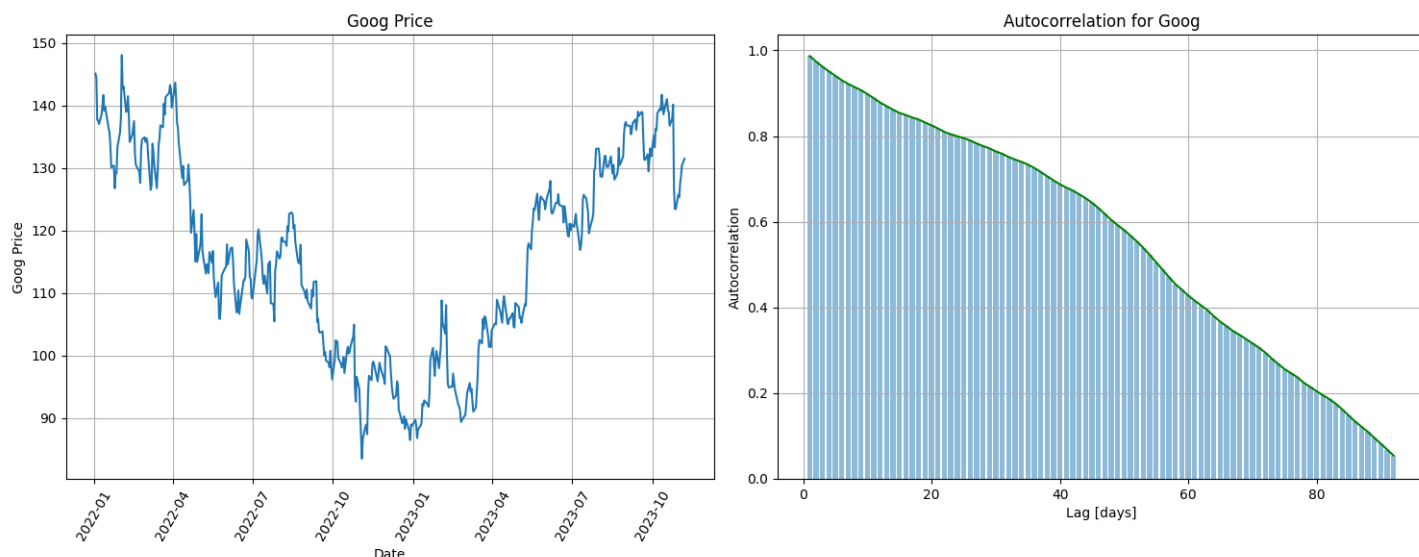
Za risanje grafov in podatke sem uporabil python knjižnici *pandas* za delovanje z takšnim tipom podatkov ter yahoo finance *yfinance*, ki nam omogoča brezplačen (a veendar zato malo zamujen) vpogled v zgodovino finančnih instrumentov. Nato pa sem za implementacijo avtokorelacije kar uporabil vgrajeno funkcijo *pandas.autocorr*.

Za zabavo sem tudi dodal graf, ki pa ne predstavlja resnične slike, vendar naključno generirane podatke, saj so cene delnic le stohastičen proces.

### 5.2 Analiza krivulj avtokorelacij

Moje znanje na področju tehnične analize je omejeno, močno sem se zanašal na razne članke, zato morda interpretacija krivulj ni na željenih standardih. Vse napisano je izključno v izobraževalne namene in se ne sme jemati kot finančni nasvet in ne prevzeman odgovornosti za neželjene posledice.

Kot prej omenjeno. Avtokorelacijske krivulje razkrivajo stopnjo korelacije med cenovnimi premiki sredstva in njihovimi preteklimi premiki. Pozitivna avtokorelacija kaže na vztrajnost trendov, negativna avtokorelacija pa na povprečje.



Slika 15: Vrednost Google delnic v letu.

To prikazemo kot na grafu 15, 16 in 17.

Borzni znak 'Goog' se nanaša na podjetje Google, 'BTC' na decentralizirano valuto Bitcoin, ki jo ne smemo zamešati z BTC - Blagovno Trgovinskim Centrom Ljubljana, ter 'MSFT' na podjetje Microsoft.

Implementiral sem dve uporabni orodji za tehnično analizo:

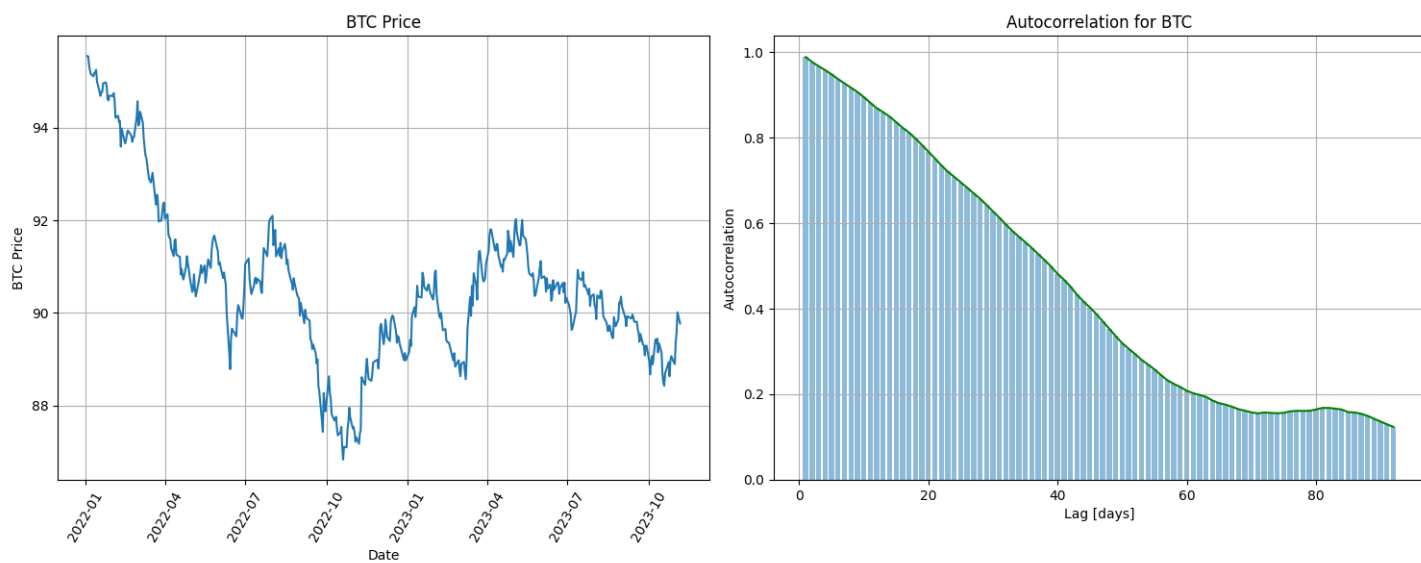
### Gibanje povprečij

Gibanje povprečij, kot so 10-dnevna in 50-dnevna gibanja povprečij, gladijo cenovne podatke in razkrivajo trende. Preklopi/preseki med krajšimi in daljšimi gibanji povprečij lahko služijo kot nakupni ali prodajni signali. Na primer, prehod, kjer 10-dnevno gibanje povprečja prečka nad 50-dnevnim gibanjem povprečij, lahko nakazuje bikovski '*bull*' trend.

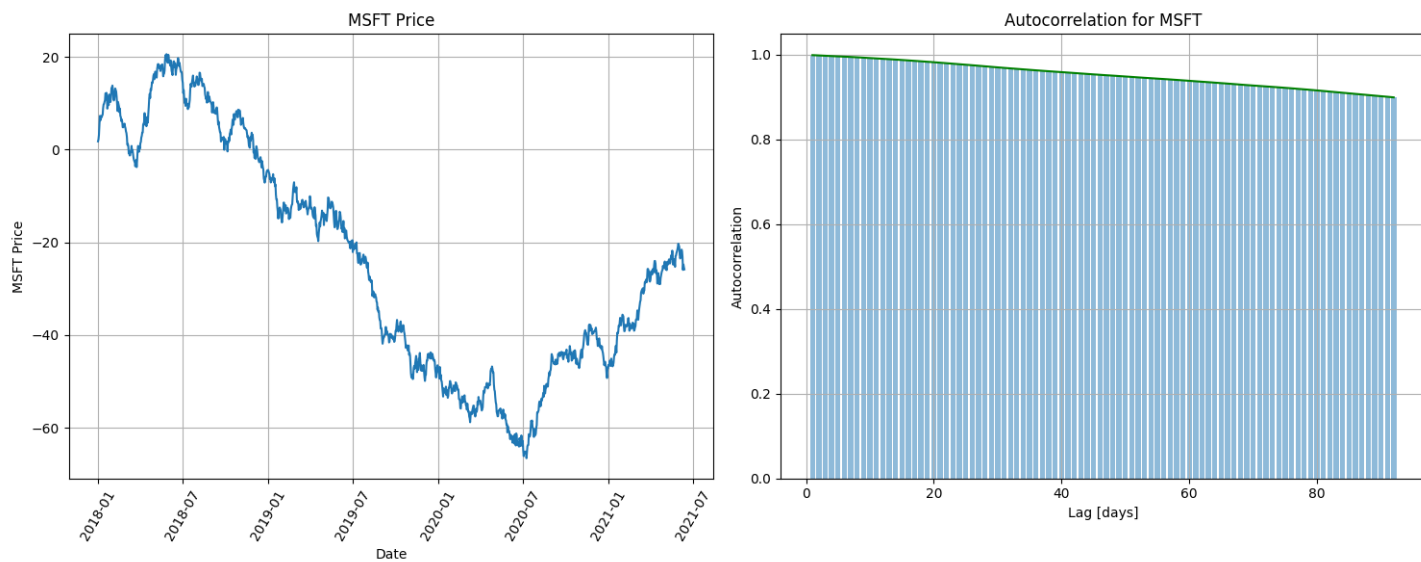
### Oscilatorji

Oscilatorji, kot tisti, ki so tukaj predstavljeni, pomagajo trgovcem prepoznavati prekupljene ali preprodane razmere. Ko oscilator prečka nad ničelno črto, lahko nakazuje na potencialen nakupni signal (prekupljene razmere). Nasprotno, ko prečka pod ničelno črto, lahko nakazuje na prodajni signal (preprodane razmere).

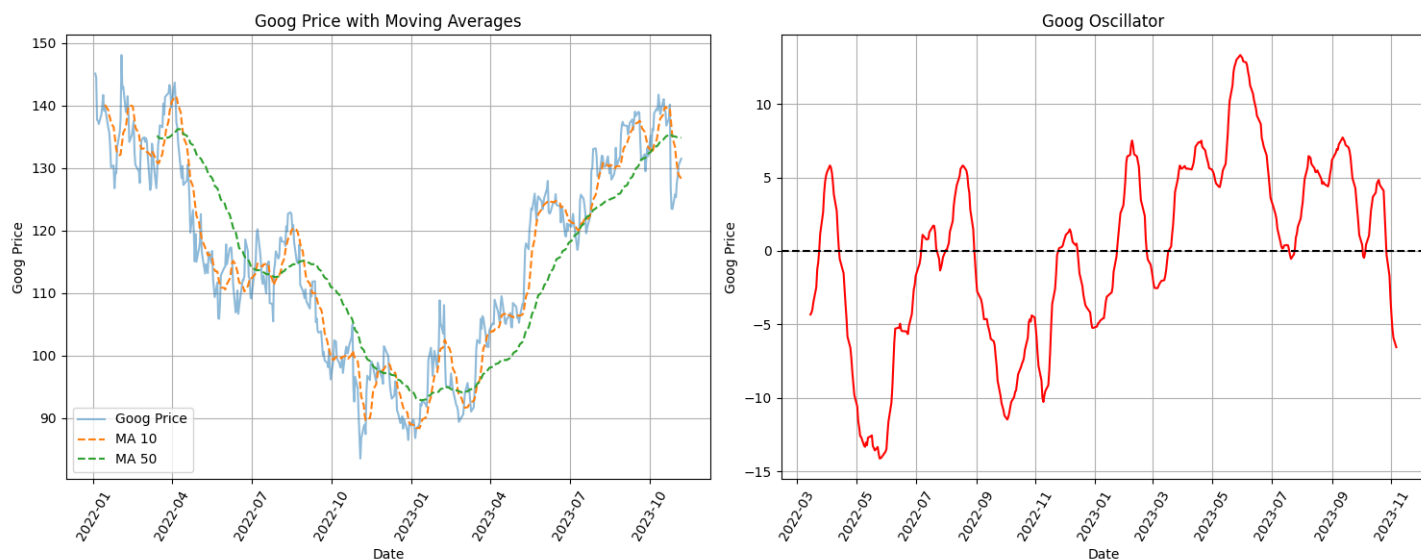
Uporabo tehnične analize prikazujejo grafi 18, 19 in 20.



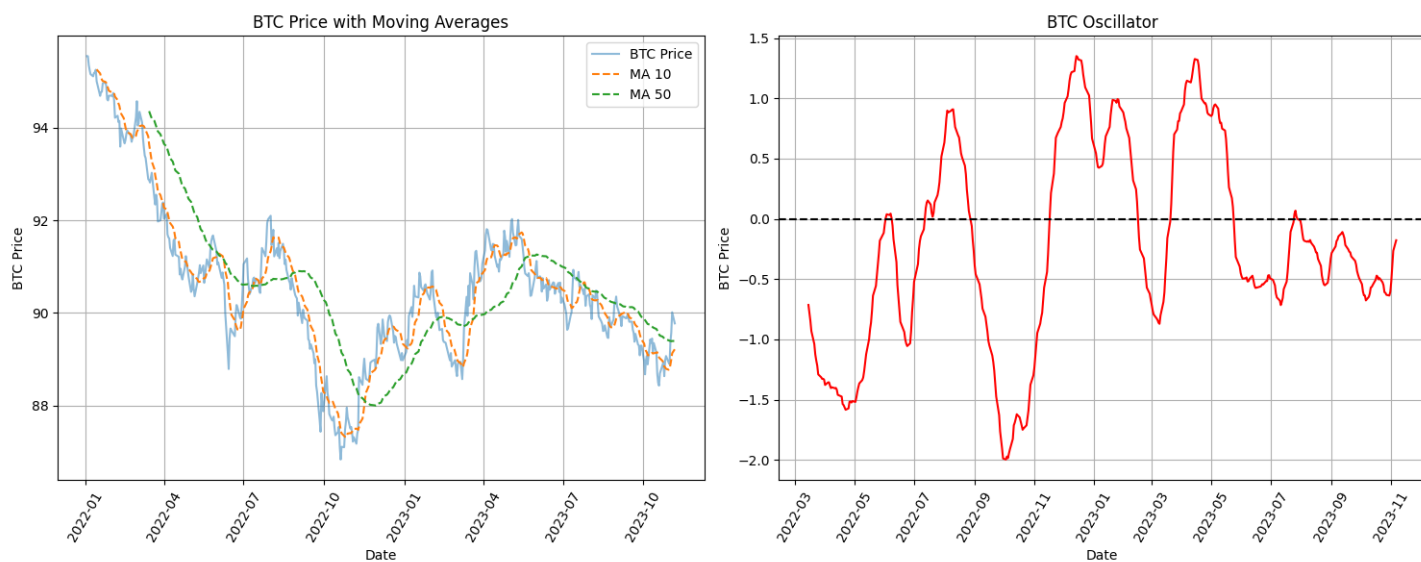
Slika 16: Vrednost Bitcoina.



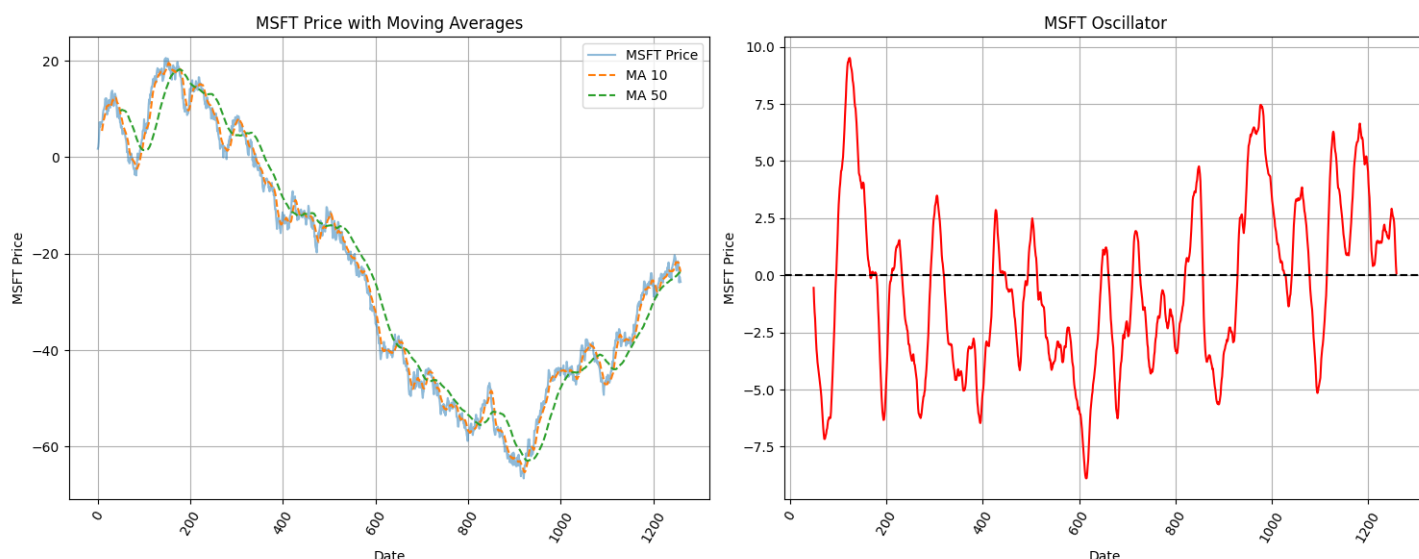
Slika 17: Vrednost Microsoft delnic.



Slika 18: Analiza grafa Googlovih delnic.



Slika 19: Analiza grafa Bitcoinove vrednosti.



Slika 20: Analiza grafa Microsoftovih delnic.

## 6 Zaključek

Cilj naloge je bil, da se spoznamo z uporabo in delovanjem hitre Fourierove transformacije in avtokorelacijskih funkcij za analizo signalov. Glavni del naloge mi je bil zelo všeč, saj mi je bil koncept Fourierove transformacije bil predstavljen ravno na takšnem primeru identificiranja zvoka v zelo zašumljenem signalu. Med raziskovanjem sem odkril, da so avtokorelacije osnovni element kvantitativnih financ in ekonometrije za analizo finančnih instrumentov. Pred dvema leti sem se z borzo ukvarjal samo na ravni psihološke analize trga, tehnična analiza mi je pa bila tuj in težaven koncept, vse se je zdelo kot neka črna magija risanja črt po grafih. Zdaj vem da je še zmeraj le črna magija ugibanja, vendar ima lahko določeno matematično ozadje. Želel sem si tudi pogledati vojaške aplikacije analize signalov kot pri zaznavi zvokov reaktivnih letal in klasifikacije le teh [4, vendar mi je zato, kot vedno, zmanjkalo časa.

## Literatura

- [1] Simon Bukovšek, *Skripta predavanj Matematika 4, Ljubljana 2023*.
- [2] Nyquist–Shannon sampling theorem
- [3] ARIMA - Autoregressive Integrated Moving Average
- [4] Autocorrelation analysis of military jet aircraft noise
- [5] Autocorrelation Wikipedia