

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Maria Krajačić

Individualna i kolektivna dinamika neurona

Zagreb, 2018.

Ovaj rad je izađen na Zavodu za teorijsku fiziku čestica i polja, Fizičkog odsjeka, Prirodoslovno-matematičkog fakulteta, Sveučilišta u Zagrebu pod vodstvom mentora izv.prof.dr.sc. Davora Horvatića i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2017./2018.

Popis kratica

DFA	analiza flutuacija kojima je uklonjen trend
ACF	autokorelacijska funkcija
ARFIMA	integrirani autoregresivni proces s pomičnim prosjekom i razlomačkom razlikom
RMT	teorija matrice slučajnih varijabli
SVD	rastav na singularne vrijednosti
PCA	analiza glavnih komponenti

Sadržaj

1	Uvod	1
2	Mjerenje neuronske aktivnosti	2
3	Analiza fluktuacija kojima je uklonjen trend	6
4	Fenomen prijelaza	13
5	Kroskorelacije među neuronima	18
6	Zaključak	24
7	Zahvale	25
	Dodaci	26
A	Implementacija DFA metode	26
B	Pregled zadatak/nagrada segemenata za cijeli vremenski niz	31
	Literatura	36

1 Uvod

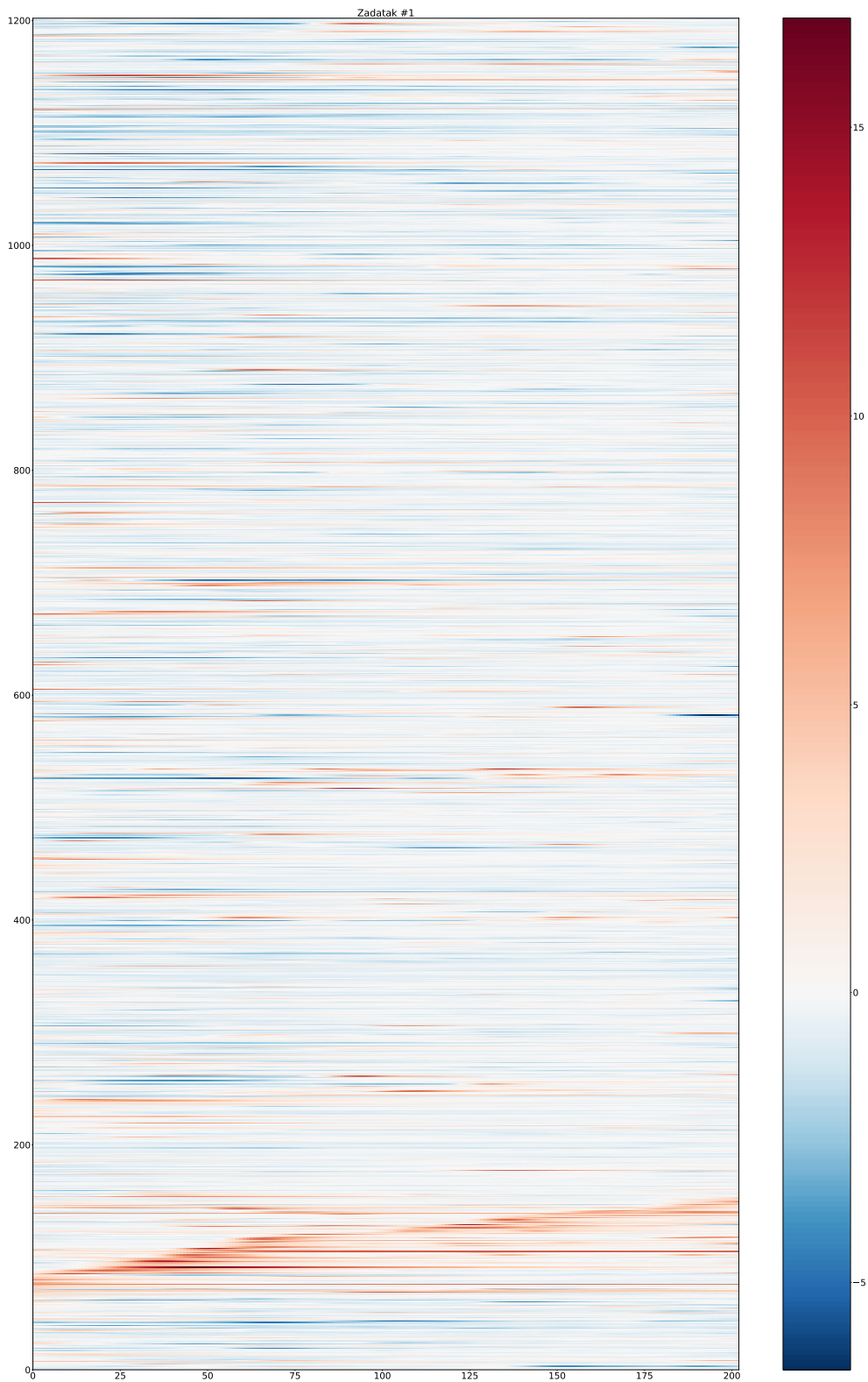
Hipokampus kao glavna komponenta u mozgu ljudi i ostalih kralježnjaka ima veliku ulogu u raznim procesima vezanim uz pamćenje i od velike je važnosti za prostornu memoriju koja omogućuje orijentaciju. Neki hipokampalni neuroni aktiviraju se kada životinja dođe na određeno mjesto u prostoru, te ih nazivamo stanicama za lociranje (eng. place cells). Aktivacija neurona ovisi o lokaciji na kojoj se životinja nalazi, zbog čega nisu sve stanice za lociranje aktivne uvijek i u isto vrijeme. Također, neki neuroni mogu biti aktivni i na različitim lokacijama, pa određenu točku u prostoru karakterizira jedinstvena populacija aktivnih neurona. Otprilike 30% hipokampalnih neurona su stanice za lociranje, a preostali dio nazivamo tihi neuroni, jer je njihova aktivnost u odnosu na stanice za lociranje puno manja. Snimanjem aktivnosti neurona dobivamo vremenske serije koje možemo analizirati raznim metodama statističke fizike. U ovom radu smo za obradu podataka koristili dvije metode. Prva je analiza fluktuacija kojima je uklonjen trend, na temelju koje smo određivali korelacijsku strukturu. S druge strane, teorija matrice slučajnih varijabli daje nam uvid u kroskorelacije između neurona.

U drugom poglavlju dajemo kratak opis i prikaz signala koji opisuju aktivnost pojedinih neurona. Treće poglavlje daje pregled analize fluktuacija kojima je uklonjen trend. U četvrtom poglavlju opisujemo fenomen prijelaza i proučavamo autokorelacijska svojstva na raznim skalama. Peto poglavlje daje pregled metoda baziranih na kroskorelacijskoj matrici povezanih s kolektivnom dinamikom neurona.

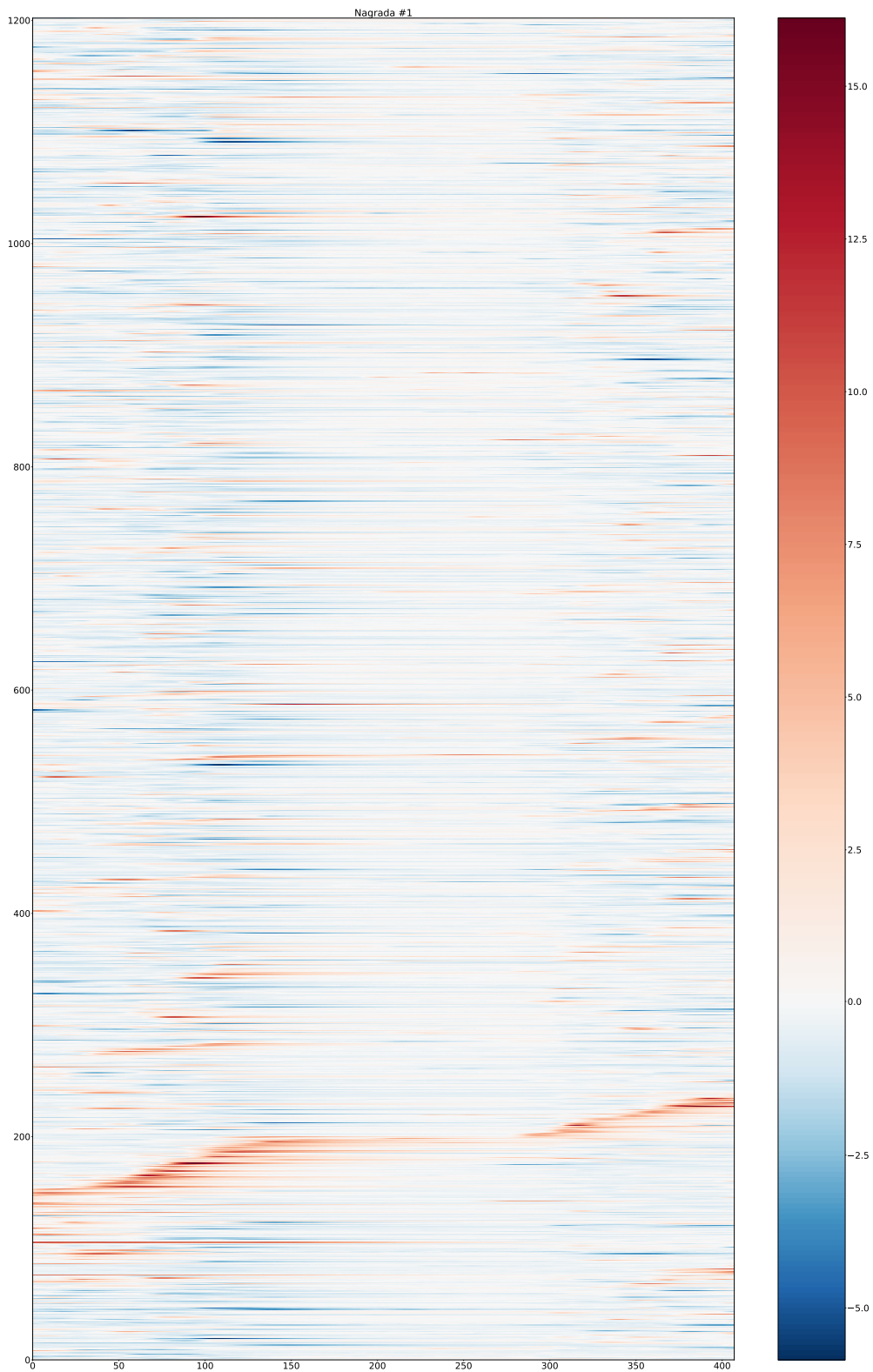
2 Mjerenje neuronske aktivnosti

U ovom radu analizirali smo podatke dobivene snimanjem aktivnosti neurona transgeničnog miša (ekspresija GCaMP3, genetski enkodiran kalcij indikator) tijekom trčanja po pokretnoj traci. Trajanje snimanja aktivnosti pojedinih neurona metodom fluorescencije je 25 minuta 42 sekunde, a frekvencija uzorkovanja iznosi 20 Hz. Miš ima zadatak trčati po pokretnoj traci, a trajanje zadatka je 10 sekundi. Na kraju svakog zadatka miš dobije vodu za nagradu. Podaci su snimljeni na Theoretical Cognitive Neuroscience Lab, Center for Memory and Brain, Boston University, a eksperiment je dobio odobrenje odbora za etičko postupanje sa životinjama. Navedena tehnika sada predstavlja frontu istraživanja, a signali u ovom radu dižu kompleksnost ukupnog broja neurona za sva reda veličine u odnosu na slične radove [1].

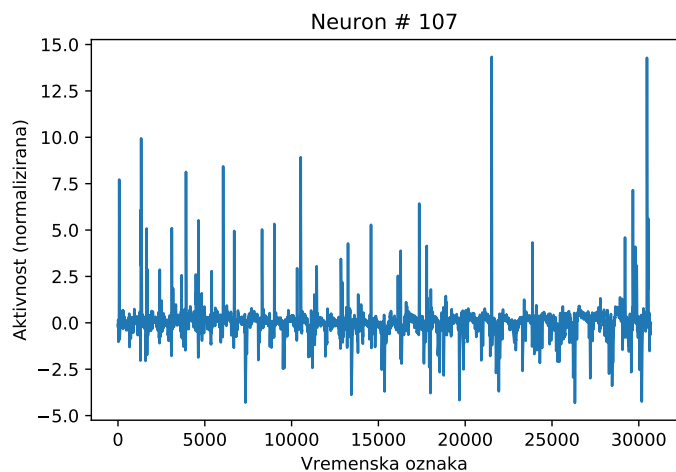
Kolektivna aktivnost neurona prikazana je na slikama 2.1 i 2.2. Na y-osi prikazani su svi neuroni, a na x-osi je vremenska skala (u analizi podataka gledali smo aktivnost neurona u vremenu, prostornu komponentu smo nismo uzeli u razmatranje). Ako ovaj prikaz kolektivne aktivnosti zamislimo kao matricu, tada jedan redak te matrice predstavlja aktivnost pojedinog neurona, a prema intenzitetu boje možemo vidjeti kada je određeni neuron aktivan. Na slikama 2.3, 2.4 i 2.5 prikazana je aktivnost tri karakteristična neurona različitih aktivnosti kao funkcija vremenskih koraka. Budući da zadatak traje 10 s, a frekvencija snimanja je 20 Hz, svaki zadatak obuhvaća 200 vremenskih koraka.



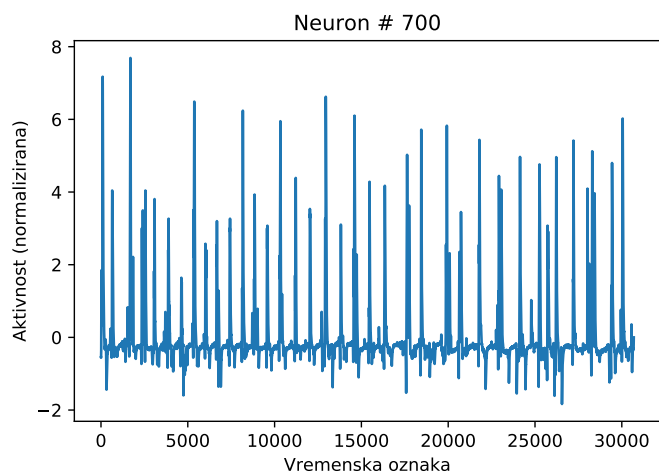
Slika 2.1: Kolektivna aktivnost neurona za prvi zadatak. Vremenska skala prikazana je na x -osi, a na y -osi su neuroni. Prema intenzitetu boje možemo vidjeti aktivnost neurona. Između otprilike 100. i 150. neurona vidimo veći intenzitet, odnosno ti neuroni su aktivniji od ostalih.



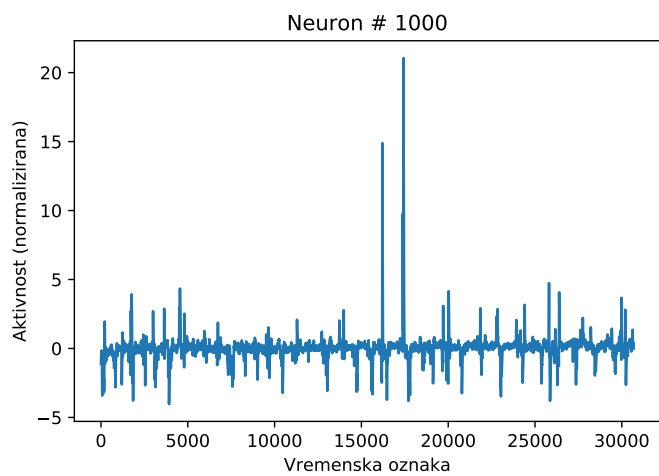
Slika 2.2: Kolektivna aktivnost neurona za prvu nagradu. Vremenska skala prikazana je na x -osi, a na y -osi su neuroni. Na ovom prikazu vidimo pojačanu aktivnost između otprilike 170 i 220 neurona.



Slika 2.3: Aktivnost 107. neurona kao funkcija vremenskih koraka.



Slika 2.4: Aktivnost 700. neurona kao funkcija vremenskih koraka.



Slika 2.5: Aktivnost 1000. neurona kao funkcija vremenskih koraka.

3 Analiza fluktuacija kojima je uklonjen trend

Razne vremenske serije dobivene snimanjem određenih fizioloških procesa mogu se analizirati metodama statističke fizike. To se prvenstveno odnosi na metode koje proučavaju kritične fenomene u kojima postoje fluktuacije u vremenskim serijama.

Svaki signal sastavljen je od fluktuacija koje su rezultat unutarnje dinamike samog sustava, ali i fluktuacija koje dolaze od vanjskih podražaja. U slučaju kada fluktuacije dobivene nekoreliranim podražajima iz okoliša možemo razdvojiti od unutarnjih fluktuacija dinamičkog sustava, pokazuje se da te dvije skupine fluktuacija imaju različita korelacijska svojstva. Fluktuacije koje proizlaze iz dinamike samog sustava mogu imati dugodosežne korelacije. S druge strane, nestacionarni tipovi fluktuacija povezani su s vanjskim podražajima, te taj dio signala nazivamo trend. Trend u praksi možemo razlikovati od finijih fluktuacija koje mogu otkriti intrinzična korelacijska svojstva dinamike tog sustava [2–5].

Za analizu fizioloških podataka uvedena je analiza modificiranog srednjeg kvadratnog korjena nasumičnog hoda, koju nazivamo analiza fluktuacija kojima je uklonjen trend (eng. *detrended fluctuation analysis*, dalje u tekstu DFA). Kod te metode vremensku seriju najprije podijelimo u prozore duljine n , a srednji kvadratni korjen (fluktuacijska funkcija) detrendirane vremenske serije u tim prozorima duljine n daje uvid u fluktuacije sustava.

DFA metoda detektira dugodosežne korelacije koje postoje u naizgled nestacionarnim vremenskim serijama, pritom izbjegavajući lažne dugodosežne korelacije koje su rezultat nestacionarih fluktuacija dobivenih od vanjskih podražaja. Ova metoda uspješno je primjenjena na detekciju dugodosežnih korelacija u visokoheterogenim DNA nizovima (sekvenca), te na ostale kompleksne fiziološke signale.

Autokorelacija je korelacija signala sa zakašnjelom kopijom istog tog signala i funkcija vremenskog kašnjenja. Što je vrijeme autokorelacije manje, sustav brže „zaboravlja“ ranije informacije i uvjete, te kažemo da postoji kratkodosežna korelacija. Vrijedi i obratno – veće vrijeme autokorelacije znači da sustav duže pamti prethodne informacije i postoji dugodosežna korelacija. Stacionarne procese koji dolaze od unutarnje dinamike sustava karakterizira divergentno vrijeme autokorelacije. Obzirom da izraz „divergentno“ predstavlja dugo vrijeme autokorelacije, takav sustav ima korelacije na većim vremenskim skalama. U tom slučaju, funkcija autokorelacije, $C(s)$, pri čemu je s vremensko kašnjenje, može biti opisana polinomom. Kada je $C(s)$ opisana eksponencijalnim padom, sustav je koreliran na kratkim vremenskim skalama i informacije koje su prethodile danom trenutku pamti vrlo kratko.

Osim toga, kao što je spomenuto ranije u tekstu, svaki proces nekog sustava je pod utjecajem vanjskih čimbenika. Ti čimbenici su nestacionarni i mogu utjecati na duljinu korelacije, obično u smjeru njenog smanjivanja.

Postoji jasna veza između funkcije fluktuacije i korelacijske strukture. Signali koji imaju kratkodosežne korelacije pokazuju jasan fenomen prijelaza. Taj prijelaz dijeli funkciju fluktuacije na dva dijela koji imaju različite eksponente skaliranja α , odnosno različite nagibe u log-log grafu. Samo u slučaju kada je duljina prozora n veća od skale na kojoj se postoji fenomen prijelaza, možemo vidjeti dobro skalirajuće ponašanje za kratkodosežne korelacije. Drugim riječima, vremenska serija koju promatramo mora biti dovoljno duga.

Funkcija autokorelacije (eng. autocorrelation function, dalje u tekstu ACF) za stacionarni proces drugog reda, x_t dana je s

$$C(s) = \frac{\langle (x_t - \mu)(x_{t+s} - \mu) \rangle}{\langle (x_t - \mu)^2 \rangle} \quad (3.1)$$

gdje je srednja vrijednost $\langle x_t \rangle = \mu$ za svaki t .

Razlika između procesa koji imaju dugo i kratko pamćenje definirana je sumom funkcije autokorelacije po svim pomacima u vremenu s $\sum_{s=0}^{\infty} C(s)$. Ako navedena suma konvergira, sustav pokazuje kratkodosežne korelacije, što znači da prilično brzo „zaboravlja“ početne uvjete, odnosno ono što se dogodilo prije određenog trenutka. Asimptotsko ponašanje koje u tom slučaju pokazuje funkcija autokorelacije opisujemo eksponencijalnim padom

$$C(s) \sim e^{-s/s_c} \quad (3.2)$$

gdje je s_c karakteristično vrijeme korelacije dano s $s_c = \int_0^{\infty} C(s) ds$.

Suprotno tome, kada suma $\sum_{s=0}^{\infty} C(s)$ divergira, sustav jako sporo „zaboravlja“ vrijednosti u prethodnim trenucima. Obzirom da postoji divergencija, ne možemo govoriti o karakterističnom vremenu kao kod kratkodosežnih korelacija. Sada funkciju autokorelacije opisujemo polinomom

$$C(s) \sim s^{-\gamma}. \quad (3.3)$$

Parametar γ je parametar korelacije i poprima vrijednosti između 0 i 1.

Određene metode procjene funkcije autokorelacije primjenjive su samo pod pretpostavkom stacionarnosti. Međutim, prirodne vremenske serije (koje nisu iz laboratorijskih uvjeta) su nestacionarne. Upravo iz tog razloga u analizi takvih vremenskih serija moramo koristiti

metode koje su primjenjive u takvim uvjetima. Da bi procjena ACF za danu seriju bila dobra, fluktuacije samog sustava moramo kvalitetno moći razdvojiti od fluktuacija koje su rezultat vanjskog utjecaja. Upravo zato je za takvu procjenu povoljna DFA metoda, jer je primjenjiva na podatke za koje ne znamo kakav je trend, odnosno ne znamo kakav je šum koji dolazi od vanjskih podražaja.

Ako želimo vremensku seriju x_t , gdje je $t = 1, 2, \dots, N$ analizirati DFA metodom moramo vremenski niz transformirati u niz y_t

$$y_t = \sum_{k=1}^t (x_k - \bar{x}), \quad (3.4)$$

gdje je \bar{x} srednja vrijednost $\bar{x} = \frac{1}{N} \sum_{k=1}^N x_k$, a y_t odgovara nasumičnom hodu s povećanjima iznosa $x_k - \bar{x}$. Vremensku os podijelimo u segmente duljine n . Za svaki segment v , gdje je $v = 1, 2, \dots, K$ računamo $y_t - p_t$, pri čemu je $t = (v-1)n+1, (v-1)n+2, \dots, vn$. p_t je polinom reda q i računamo ga metodom najmanjih kvadrata za točke (t, y_t) u v -tom segmentu. Za DFA metodu koja koristi uklanjanje trenda polinomom reda q kažemo da je DFA reda q , odnosno DFA- q . Srednja vrijednost varijance $y_t - p_t$ po svim segmentima je kvadrat funkcije fluktuacije:

$$F^2(n) = \frac{1}{K} \sum_{v=1}^K \left(\frac{1}{n} \sum_{t=(v-1)n+1}^{vn} (y_t - p_t)^2 \right). \quad (3.5)$$

Ovaj postupak ponovimo za sve duljine prozora n , $n = 2, 3, \dots, N/4$ kako bi dobili ovisnost F o duljini prozora n , $F(n)$. Širinu prozora n računamo kao $n = 2q + 2$. Za dovoljno veliki n , funkcija fluktuacije ima oblik $F(n) \sim n^\alpha$. Ovakvo ponašanje direktno je povezano sa strukturom korelacije. Za $\alpha = 0.5$ proces je koreliran na kraćim skalama, dok za vrijednosti α između 0.5 i 1, postoje dugodosežne korelacije i $\alpha = 1 - \gamma/2$, pri čemu parametar korelacije γ može poprimiti vrijednosti između 0 i 1.

Bijeli šum je naziv za signal kod kojeg su uzorci serijski nekorelirane slučajne varijable s nultom srednjom vrijednošću i konačnom varijancijom. Za ovaj tip podataka pripadni eksponent skaliranja α iznosi $\alpha = 0.5$. Ponekad su moguće korelacije na kraćim skalama, pa α može blago odstupati od spomenute vrijednosti, međutim, za veći broj podataka, α će iznositi 0.5. Vrijednost α za koju vrijedi $0.5 < \alpha \leq 1$ indicira dugodosežne korelacije za koje vrijedi da postoji veća vjerojatnost da će nakon dužeg vremenskog intervala slijediti još jedan duži vremenski interval i obrnuto. S druge strane, ako α poprima vrijednosti između 0 i 0.5, postoji drugi tip korelacije i to takav da će se duži i kraći vremenski intervali naizmjenice

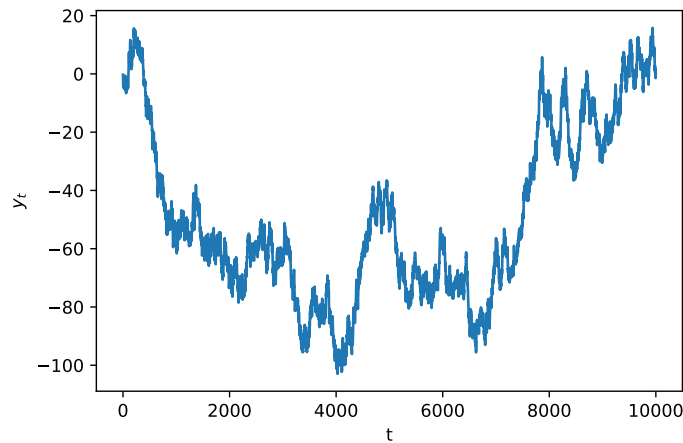
izmjenjivati. U posebnom slučaju kada α poprima vrijednost 1, signal odgovara $1/f$ šumu, a kada je $\alpha \geq 1$, korelacija postoji, međutim postoje i određena odstupanja. U posebnom slučaju kada je vrijednost $\alpha = 1.5$ imamo tzv. Brownov šum (Brown noise), koji dobijemo integracijom bijelog šuma.

Eksponent skaliranja α govori nam, s druge strane, o „hrapavosti“ vremenske serije i to tako da veći α odgovara glađoj seriji, a manja vrijednost opisuje više hrapavi signal. Sada $1/f$ signal možemo opisati kao kompromis između u potpunosti nepredvidivog bijelog šuma i glatkog Brownovog šuma.

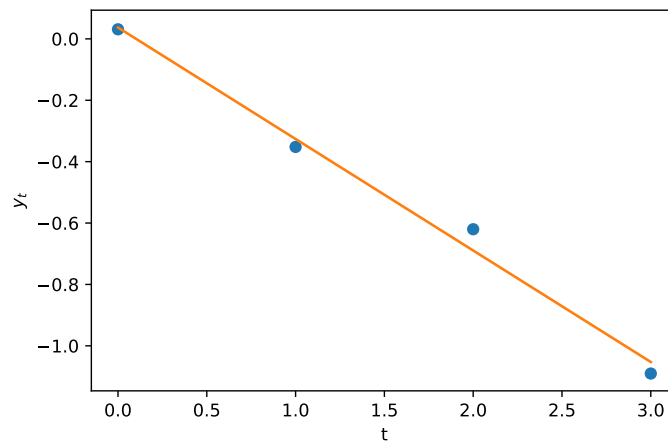
Kada ovisnost F o n prikažemo u log-log grafu, dobivamo linearnu ovisnost koja indicira prisutnost skaliranja. To znači da fluktuacije karakterizira eksponent skaliranja α koji predstavlja nagib pravca u $\log(F(n)) - \log(n)$ grafu. U ovom radu algoritam DFA metode implementirat ćemo u python programskom jeziku (danas standardni izbor u obradi podataka [6]).

```
def dfa(tmp, box, nfit=1):
    xtmp=np.arange(1,box+1);
    chisq=0.0
    for i in range(len(tmp)-box+1):
        temp=0.0
        ytmp=tmp[i:i+box]
        pc=np.polyfit(xtmp,ytmp,nfit)
        temp = sum((ytmp-np.polyval(pc,xtmp))**2)
        chisq += temp
    return chisq/((len(tmp)-box+1)*box)
```

Promorimo kod implementacije kroz jednadžbu (3.5). Krenimo od neke proizvoljne vremenske serije. Prvi primjer koji ćemo promatrati je vremenski niz sastavljen od nasumičnih vrijednosti iz standardne normalne distribucije. Dobivene varijable kumulativno sumiramo (jednadžba 3.4). Rezultat je prikazan na slici 3.1. Sada odaberemo veličinu prozora za koju ćemo računati kvadrat funkcije fluktuacije, jednadžba 3.5. Za primjer računa podsume u jednadžbi 3.5 možemo uzeti prozor od prve 4 točke niza y_t , što je ujedno i najmanja veličina prozora koju možemo koristiti. Trend za određeni prozor dobivamo tako da kroz odabrane točke provučemo pravac dobiven metodom najmanjih kvadrata (DFA-1), kao što prikazuje slika 3.2. Kada nađemo trend za dani prozor i zbrojimo kvadrate reziduala, pomičemo prozor



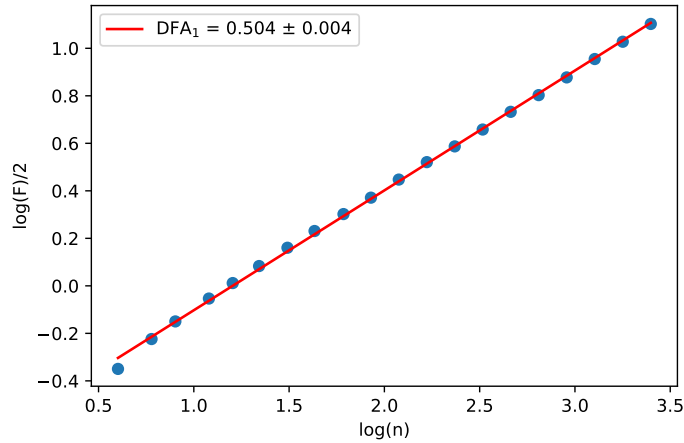
Slika 3.1: Vremenski niz kumulativno sumiranih vrijednosti, y_t , dobivenih prema jednadžbi 3.4 iz vremenskog niza, x_t , 10 000 slučajnih varijabli iz standardne normalne distribucije



Slika 3.2: Prozor od prve 4 točke niza y_t i pravac dobiven metodom najmanjih kvadrata koji u navedenom segmentu predstavlja trend

za jednu vremensku točku na novu poziciju i ponovo na isti način računamo trend. Postupak ponavljamo dok ne dođemo do kraja vremenske serije, a to nam u jednadžbi 3.5 pokazuje prva suma, $\sum_{v=1}^K$. Na ovaj način usrednjili smo podatke po svim prozorima. Ovu proceduru ponavljamo za sve širine prozora, pri čemu je najveća širina prozora odgovara 1/4 duljine vremenske serije koju analiziramo. Rezultat DFA metode za odabranu širinu prozora prikazan je u log-log grafu na slici 3.3. Nagib dobivenog pravca je $\alpha = 0.504 \pm 0.004$, što je u skladu s teorijskom vrijednošću, $\alpha = 0.5$.

Obzirom da sustavi koje promatramo mogu biti vrlo kompleksni, uvijek je dobro za testiranje koristiti više procesa koji generiraju vremenske serije, kako bi se provjerila ispravnost dobivenih rezultata. U tu svrhu možemo koristiti ARFIMA proces (eng. fractionally auto-



Slika 3.3: Grafički prikaz ovisnosti funkcije fluktuacije o širini prozora $F(n) - n$ u log-log grafu za vremensku seriju slučajnih brojeva iz standardne normalne distribucije.

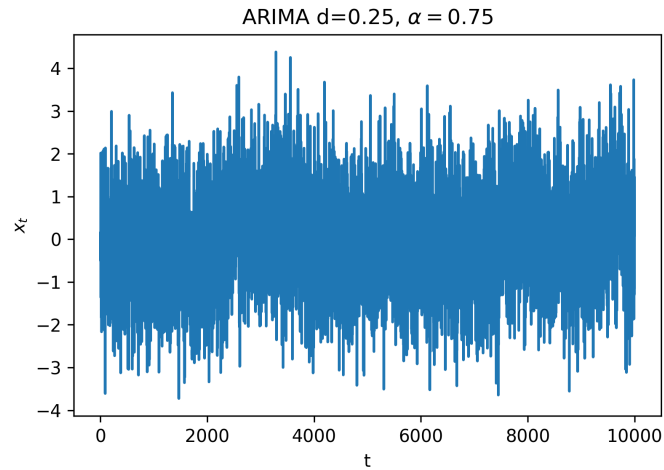
regressive integrated moving-average process), pomoću kojeg možemo modelirati procese s dugodosežnim autokorelacijama u stohastičkim varijablama. Vremensku seriju x_t možemo izračunati pomoću

$$x_t = \sum_{n=1}^{\infty} a_n(d)x_{t-n} + \varepsilon_t, \quad (3.6)$$

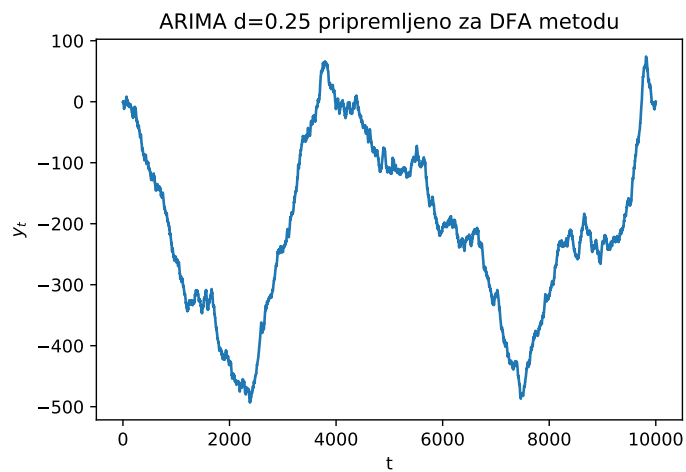
pri čemu vrijedi da parametar skaliranja d poprima vrijednosti između -0.5 i 0.5 . ε_t predstavlja nezavisno i jednoliko raspodjeljene slučajne varijable. Vrijedi da je srednja vrijednost $\langle \varepsilon_t \rangle = 0$ i varijanca $\langle \varepsilon_t^2 \rangle = 1$. Težinski faktor $a_n(d)$ definiran je na sljedeći način:

$$a_n(d) = \frac{d\Gamma(n-d)}{\Gamma(1-d)\Gamma(n+1)}, \quad (3.7)$$

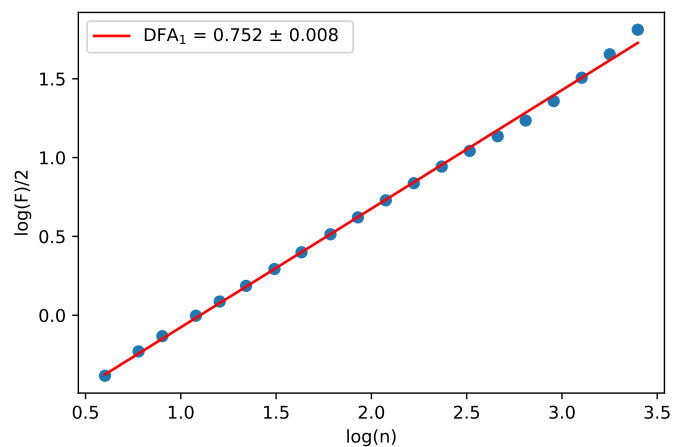
gdje Γ označava gama funkciju, a n je vremenska skala. Funkciju autokorelacije za vremensku seriju x_t označavamo s $A(n)$, te vrijedi da je $A(n) \equiv A(x_t, x_{t-n})$. Vremenska serija x_t postaje nasumična za $d = 0$. U sljedećem primjeru uz pomoć ARFIMA procesa generirat ćemo vremensku seriju i provjeriti ispravnost našeg prethodnog rezultata. Vrijednost parametra skaliranja je $d = 0.25$. Bitno je napomenuti da su nagib α i parametar skaliranja d povezani izrazom $\alpha = 0.5 + d$. Zbog ograničenja računalne preciznosti, u stvarnim primjerima nije potrebno raditi beskonačnu sumu, stoga ćemo u našem računu podatke sumirati do $n = 1000$. Na slici 3.4 prikazan je jedan takav niz. Rezultat primjene jednadžbe 3.4 prikazan je na slici 3.5, a rezultat DFA metode na slici 3.6. Vidimo da je vrijednost dobivenog nagiba $\alpha = 0.752 \pm 0.008$, što se dobro slaže s našom ranijom analizom, jer vrijedi da je $\alpha = d + 0.5$. Detaljni pregled implementacije bit će izložen u dodatku A.



Slika 3.4: Vremenski niz dobiven ARFIMA procesom. Parametar skaliranja je $d = 0.25$



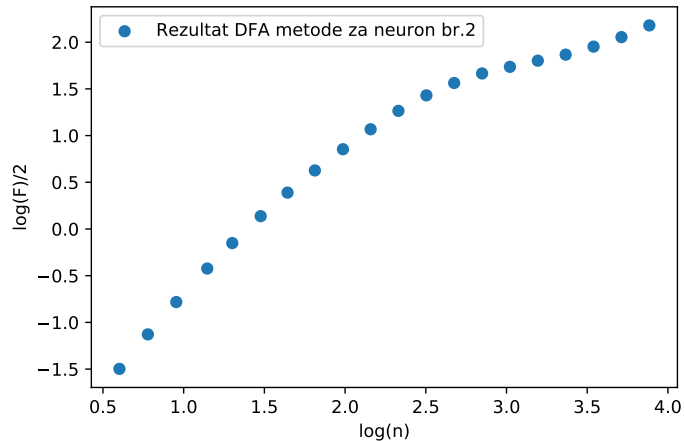
Slika 3.5: Vremenski niz kumulativno sumiranih vrijednosti, y_t , dobivenih iz vremenskog niza, x_t , generiranog ARFIMA procesom. Parametar skaliranja d iznosi $d = 0.25$



Slika 3.6: Rezultat DFA metode za vremenski niz x_t generiran ARFIMA procesom

4 Fenomen prijelaza

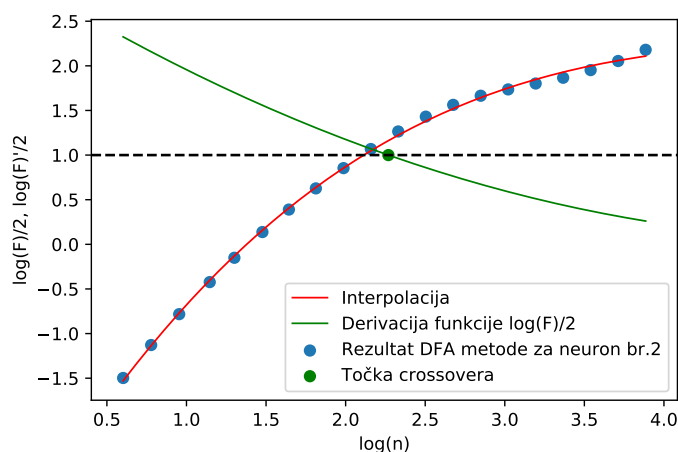
Eksponent skaliranja može nam poslužiti u dijagnostičke svrhe, no problem je što je za takvu analizu potreban velik broj podataka. Obično kada koristimo kraće vremenske serije na njima je vidljiv fenomen prijelaza [7]. Rezultat DFA metode u log-log grafu za neuron označen rednim brojem 2 prikazan je na slici 4.1. Jasno se vidi da krivulja mijenja nagib,



Slika 4.1: Rezultat DFA metode za neuron br. 2 prikazan u log-log grafu

odnosno, na neki način bismo dobiveni grafički prikaz mogli podijeliti u 2 dijela različitih nagiba koji su međusobno odvojeni područjem prijelaza. Ranije u tekstu spomenuto je da fluktuacije karakterizira eksponent skaliranja α . Dakle, na rezultat DFA metode možemo prilagoditi dva pravca i odrediti njihove nagibe, a ti nagibi su upravo eksponenti skaliranja α_1 i α_2 .

Prijelaz iz jednog u drugi režim nalazi se u točki u kojoj je derivacija funkcije po logaritmu širine prozora jednaka 1. Obzirom da analizu radimo s diskretnim podacima, koristimo tzv. spline interpolaciju. Kako za obradu podataka koristimo programski jezik python koristit ćemo implementaciju `UnivariateSpline` [8] iz `scipy` paketa. Ovaj paket također može automatski računati derivaciju, što je još jedan razlog odabira istog. Na slici 4.2 prikazana je metoda određivanja točke prijelaza.

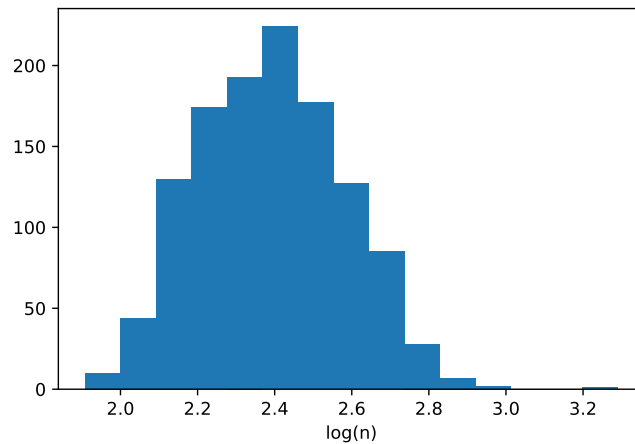


Slika 4.2: Rezultat DFA metode za neuron br. 2 i interpolacija. Koordinata točke prijelaza određena je točkom u kojoj derivacija funkcije $\log(F(n))/2$ po logaritmu širine prozora iznosi 1.

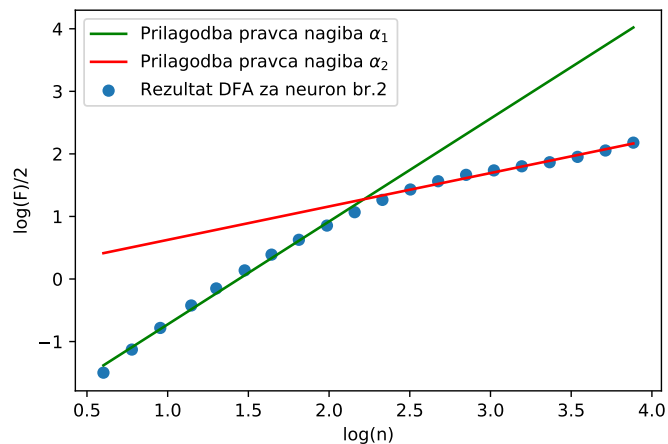
Sada lako možemo odrediti točku prijelaza tražeći rješenje jednadžbe u kojoj derivacija funkcije $\log(F(n))/2$ po logaritmu širine prozora iznosi 1. Kako bismo točku prijelaza brže i jednostavnije izračunali za sve neurone, napravili smo funkciju koja kao argument uzima redni broj neurona, a kao rezultat daje koordinatu prijelaza u logaritamskoj skali.

```
def crossover(nrn):
    ex=np.log10(dfaj[:, 2*nrn])
    ey=np.log10(dfaj[:, 2*nrn+1])/2
    spl=UnivariateSpline(ex, ey)
    xs=np.linspace(ex[0], ex[-1], 1000)
    dspl=spl.derivative()
    t=optimize.newton(lambda x: dspl(x)-1, 1.5)
    return t
```

Raspodjelu točaka prijelaza svih neurona prikazuje histogram na slici 4.3. Maksimum distribucije se nalazi na širini prozora $n = 233$ ($\log(233) = 2.4$) i predstavlja sredinu prozora u histogramu za koji opažamo navedeni maksimum. Kako znamo iz opisa podataka, trajanje pojedinog zadatka traje 200 vremenskih koraka (trajanje zadataka iznosi 10 sekundi uz frekvenciju uzorkovanja od 20 Hz), tako da analiza pokazuje da se točka prijelaza obično javlja 30 vremenskih koraka nakon kraja zadataka što odgovara vremenskom intervalu od 1.5 s. Budući da nas zanimaju eksponenti skaliranja, moramo odrediti nagibe za dva različita



Slika 4.3: Distribucija točke prijelaza za sve vremenske serije. Maksimum distribucije se nalazi na širini prozora $n = 233$.



Slika 4.4: Prilagodba pravca nagiba α_1 na prvi segment, te pravca nagiba α_2 na drugi segment rezultata DFA metode za neuron br. 2.

područja.

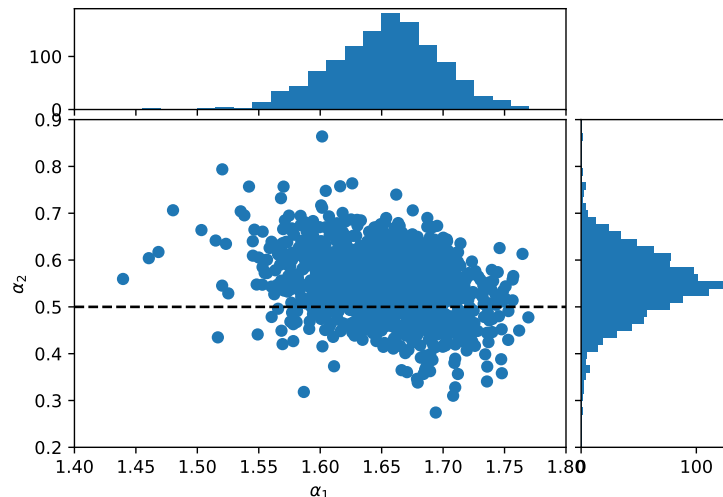
Metodom najmanjih kvadrata odredit ćemo nagibe pravaca u dva područja odvojena točkom prijelaza. Nagibi su automatski određivani uz pomoć funkcije `slopes`, čiji je algoritam prikazan niže u tekstu.

```
def slopes(nrn):
    ex=np.log10(dfaj[:, 2*nrn])
    ey=np.log10(dfaj[:, 2*nrn+1])/2
    c_nrn=crossover(nrn)
    ex1=ex[ex[:] < c_nrn]
```

```

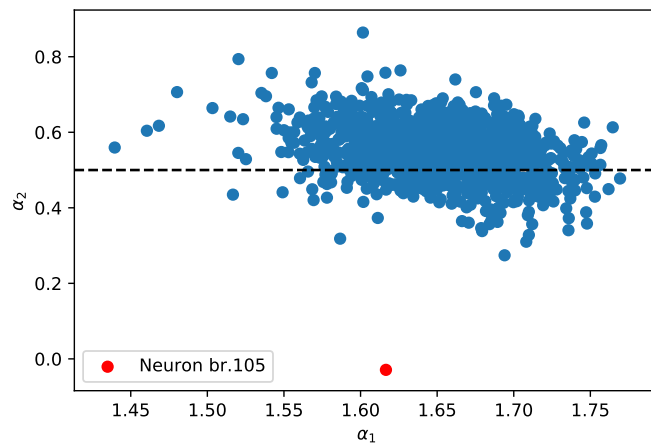
cross=len(ex1)
xs = np.linspace(ex[0], ex[-1], 1000)
pf=np.polyfit(ex[:cross], ey[:cross], 1)
pf1=np.polyfit(ex[cross:], ey[cross:], 1)
slope=pf[0],pf1[0]
return slope

```

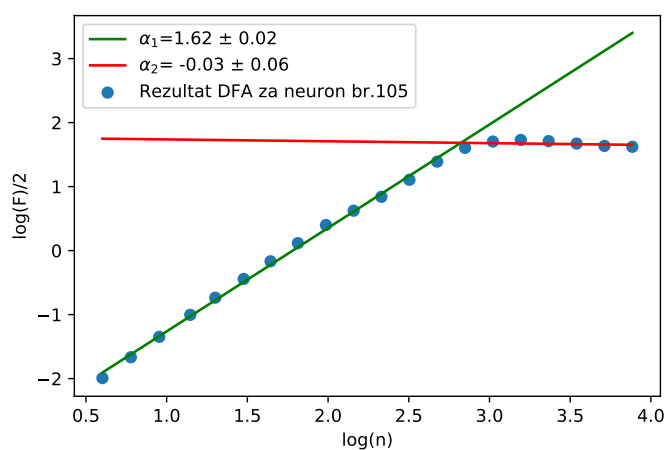


Slika 4.5: Grafički prikaz nagiba za 1201 neuron. Nagibi za pojedini neuron prikazani su kao točka s koordinatama (α_1, α_2) . Gornji histogram prikazuje distribuciju nagiba α_1 za sve neurone, a histogram na desnoj strani distribuciju nagiba α_2 .

Na slici 4.5 prikazani su nagibi za svaki pojedini neuron kao točka s koordinatama (α_1, α_2) . Pravac $\alpha_2 = 0.5$ prikazuje graničnu vrijednost. Neuroni za koje vrijedi da je $\alpha_2 > 0.5$ pokazuju dugodosežne korelacije, a oni za koje vrijedi $\alpha_2 < 0.5$ dugodosežne antikorelacije. Iz raspodjele je vidljivo da 22.3% neurona ne pokazuje dugodosežne korelacije nakon točke prijelaza. S druge strane, srednja vrijednost nagiba na kraćim vremenskim skalama iznosi $\bar{\alpha}_1 = 1.65$. To je nešto više od vrijednosti koja odgovara Brownovom šumu, pa možemo zaključiti da je signal nestacionaran. Standardna devijacija za α_1 iznosi 0.05, dok je njena vrijednost za podatke na većim vremenskim skalama 0.07. Na slici 4.5 nije prikazan neuron 105 čiji nagib α_2 značajnije odstupa od vrijednosti nagiba za druge neurone, kao što prikazuje slika 4.6. Ako pogledamo rezultat DFA metode za dani neuron (slika 4.7), vidimo da naša metoda radi ispravno, no fenomen prijelaza se kod ovog neurona javlja tek za širinu prozora $n = 510$ ($\log(510) = 2.7$). Dva nagiba i pripadne pogreške su $\alpha_1 = (1.62 \pm 0.02)$ i $\alpha_2 = (-0.03 \pm 0.06)$. Nakon točke prijelaza za neuron 105 nemamo nagib koji je statistički značajno različit od nule.



Slika 4.6: Grafički prikaz eksponenata skaliranja za sve neurone. Fenomen prijelaza za neuron 105 javlja se za širinu prozora $n = 510$, a pripadajući nagib pravca α_2 iznosi $\alpha_2 = -0.03 \pm 0.06$.



Slika 4.7: Prilagodba pravca nagiba α_1 na prvi segment, te pravca nagiba α_2 na drugi segment rezultata DFA metode za neuron br. 105.

5 Kroskorelacije među neuronima

Jedna od metoda određivanja korelacija u vremenskim serijama je i teorija matrice slučajnih varijabli (eng. random matrix theory, dalje u tekstu RMT), koju je u nuklearnu fiziku uveo E. Wigner [9] kako bi modelirao energetske nivoe jezgri teških atoma. Wigner je pretpostavio da su interakcije između konstituenata u jezgri toliko kompleksne da ih možemo modelirati kao nasumične. Na temelju te pretpostavke izveo je statistička svojstva velikih simetričnih matrica čiji su elementi slučajne vrijednosti, a rezultati su se jako dobro slagali s eksperimentalno dobivenim rezultatima. RMT metoda temelji se na usporedbi distribucija svojstvenih vrijednosti dviju matrica: prve, Wishartove matrice, koja je matrica korelacije neovisnih i jednoliko distribuiranih varijabli i druge, matrice kroskorelacije empirički dobivenih vremenskih serija.

Razlika u distribuciji svojstvenih vrijednosti tih dviju matrica daje nam informaciju o kroskorelacijama. Ako u empirički dobivenim vremenskim serijama postoje kroskorelacije, tada će postojati singularna vrijednost veća od λ_+ , pri čemu najveći iznos svojstvene vrijednosti indicira globalno ponašanje višestrukih vremenskih serija. Svojstvene vrijednosti iznosa manjeg od λ_+ i pripadajuće svojstvene vektore razmatramo kao šum [10]. Danas je ta metodologija postala standard u istraživanju neuralne dinamike [11].

Stupanj kroskorelacije između dvije vremenske serije određujemo koeficijentom kroskorelacije kojeg računamo kao

$$C_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\langle (X - \mu_X)(Y - \mu_Y) \rangle}{\sigma_X \sigma_Y}, \quad (5.1)$$

pri čemu su σ_X i σ_Y standardne devijacije od X i Y , a μ_X i μ_Y su pripadajuće očekivane vrijednosti. U empirički dobivenim vremenskim serijama postoje kroskorelacije između parova serija, međutim postoje i autokorelacije unutar pojedinačne vremenske serije. Prema tome, razlika u distribuciji svojstvenih vrijednosti matrice korelacije dobivene iz empiričkih podataka i distribuciji svojstvenih vrijednosti Wishartove matrice može biti rezultat i kroskorelacija i autokorelacija.

Elemente matrice kroskorelacije računamo pomoću izraza 5.2.

$$C_{ij} \equiv \frac{\langle X_{i,t} X_{j,t} \rangle - \langle X_{i,t} \rangle \langle X_{j,t} \rangle}{\sigma_X \sigma_Y}. \quad (5.2)$$

Vremenska serija X od N varijabli i duljine T dana je kao $X = \{X_{i,t} : i = 1, \dots, N; t = 1, \dots, T\}$,

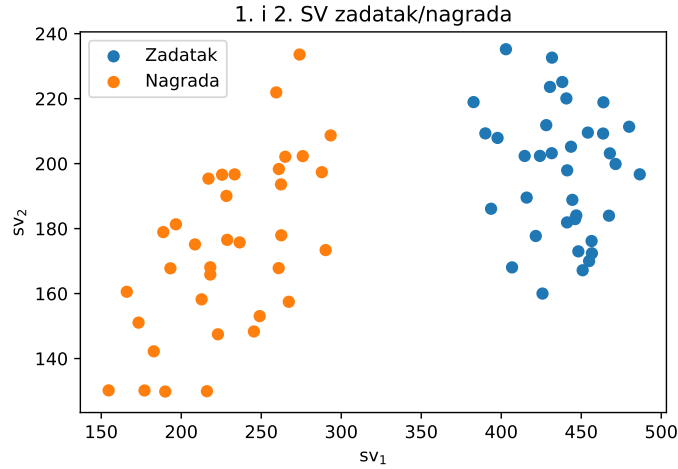
pri čemu N označava dimenziju broja neurona, a T dimenziju vremenskih koraka. Indeks i je indeks vremenske serije, a početna matrica je dimenzija $N \times T$ i vrijedi da je $T > N$ ($\frac{T}{N} > 1$). Tu matricu pomnožimo s transponiranom matricom i dobijemo konačnu matricu kroskorelacija dimenzija $N \times N$. Element C_{ij} je kroskorelacijski koeficijent između i -tog i j -tog neurona. Kasnije ćemo gledati kroskorelacije s pomakom u vremenu. Svojstvene vrijednosti u tom slučaju su kompleksni brojevi, pa ćemo, kako bi zadržali istu interpretabilnost, umjesto svojstvenih vrijednosti koristiti singularne.

Standardna metoda za redukciju dimenzionalnosti u analizi podataka je rastav na singularne vrijednosti (eng. singular value decomposition, dalje u tekstu SVD) [12]. SVD je metoda faktorizacije matrice u kojoj matricu M dimenzija $m \times n$ faktoriziramo u oblik $U\Sigma V^*$, pri čemu je U $m \times m$ jedinična matrica, Σ je pravokutna dijagonalna $m \times n$ matrica koja na dijagonali ima pozitivne realne brojeve, a V je jedinična matrica dimenzija $n \times n$ (V^* označava transponiranu matricu V). Dijagonalne vrijednosti matrice Σ su singularne vrijednosti matrice M . Matrice U i V su ortogonalne, odnosno matrice rotacije, a matrica Σ je matrica skaliranja. Dakle, izraz $U\Sigma V^*$ možemo interpretirati kao kompoziciju 3 geometrijske transformacije: rotacija, skaliranje, te ponovo rotacija.

Specifični slučaj SVD je analiza glavnih komponenti (eng. principal component analysis, dalje u tekstu PCA). PCA je procedura koja pomoću ortogonalnih transformacija kompleksne podatke prebacuje u nižu dimenziju, te na taj način otkriva jednostavnije strukture koje su često zasjenjene velikom količinom prikupljenih podataka [14]. Ova metoda omogućuje identifikaciju baze koja je najpovoljnija za izražavanje podataka, a ta nova baza je linearna kombinacija originalne. Drugim riječima, ako imamo set podataka u n dimenzija, PCA nam daje projekciju tih podataka u nižu dimenziju, ali tako da sačuvamo najviše podataka. To radimo na način da zadržimo prvih nekoliko glavnih komponenti, a ostale odbacujemo čime reduciramo dimenzionalnost naših podataka. Mjerenjem varijance duž određene komponente određujemo važnost te komponente, odnosno dimenzije. Na taj način, kvantificirali smo važnost svake pojedine dimenzije u opisivanju varijabilnosti podataka. Postupak obično ponavljamo dok ne obuhvatimo otprilike 95% varijance.

U ovom radu smo zbog brzine računanja koristili metodu TruncatedSVD [13] iz python biblioteke koja je optimizirana za metode strojnog učenja `scikit-learn`. Dodatna motivacija za korištenje matrice kroskorelacije je pitanje možemo li iz nje dobiti informaciju o dinamici neurona prema aktivnosti, odnosno saznati kada miš obavlja zadatak, a kada dobiva nagradu. SVD metodom izračunali smo singularne vrijednosti matrice kroskorelacije

i u razmatranje uzeli samo prve dvije vrijednosti. Na slici 5.1 prikazan je grafički prikaz



Slika 5.1: Prikaz prve dvije singularne vrijednosti za zadatke i nagrade. Svaki zadatak (nagrada) prikazan je kao točka s koordinatama (sv_1, sv_2) .

singularnih vrijednosti za određeni zadatak, odnosno nagradu. Vidimo da se podaci grupiraju, te je za razliku u ponašanju između zadatka i nagrade dovoljno znati iznos samo prve singularne vrijednosti.

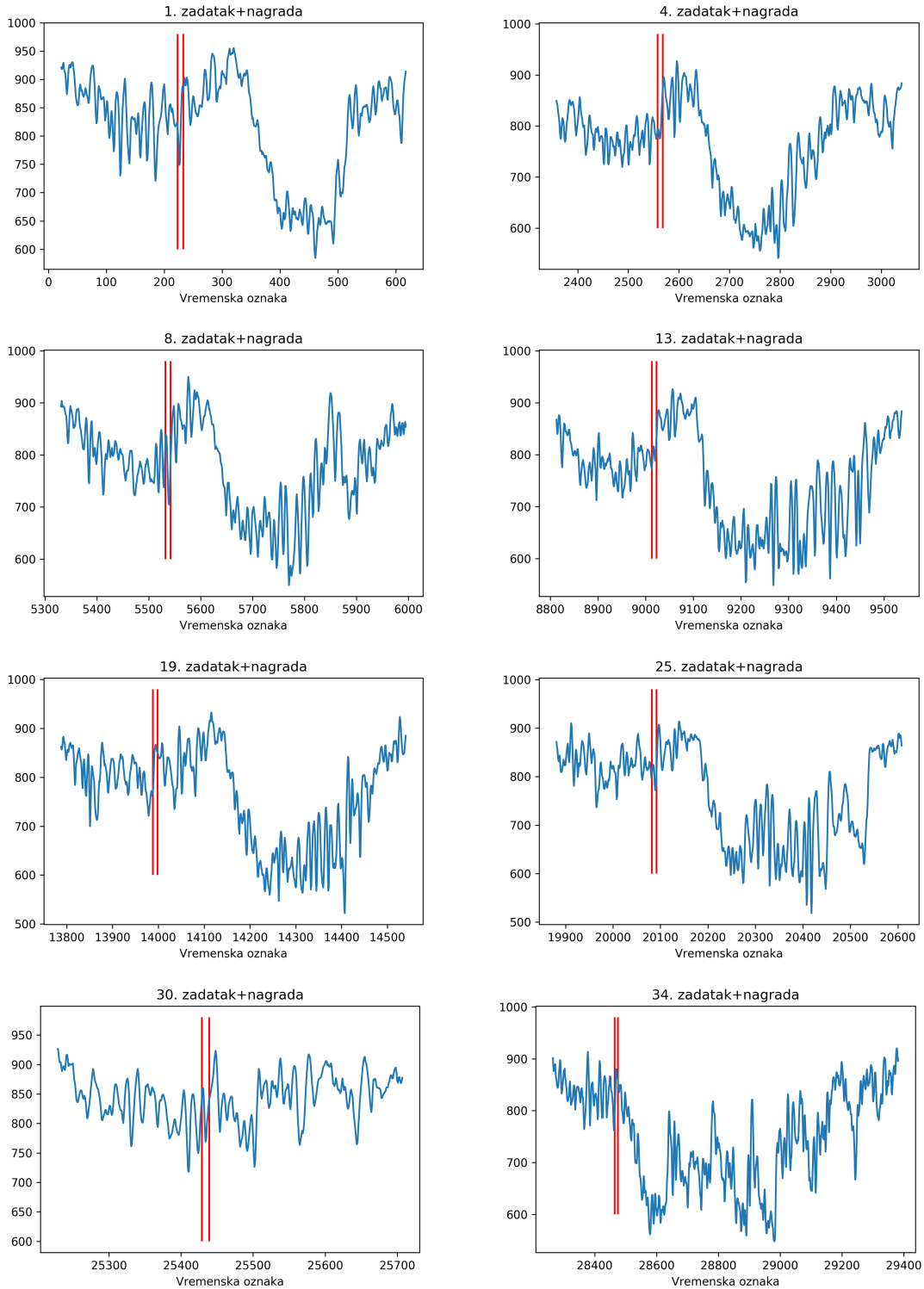
Takav rezultat nameće ideju da odaberemo pomični prozor i pomoću njega gledamo dinamiku neurona. Širina odabranog prozora je 10 vremenskih oznaka (0.5 s). Nagli prijelaz u međuinterakciji neurona predstavlja prijelaz iz zadatka u nagradu. Na slici 5.2 vidimo spomenuti prijelaz, no on se ne događa trenutno. Prikaz rezultata za sve intervale zadataka i nagrada bit će izložen u dodatku B. Izuzmemo li anomalne situacije (primjer 30. i 34. zadatka) koje dolaze iz činjenice da se radi o živom biću, koje je podložno umoru, distrakcijama i sl. vidimo jasan uzorak. Jasna separacija između zadatka i nagrade dolazi od samo jednog dijela kolektivne dinamike. Kada je riješen zadatak, potrebno je još doći do nagrade, te se vratiti na obavljanje zadatka. Indikacija vremena kada se preuzima nagrada jasno je izražena padom u iznosu maksimalne svojstvene vrijednosti.

Kao što je spomenuto ranije u tekstu, u ovom radu računamo kroskorelacije s pomakom u vremenu [15, 16]. Matricu kroskorelacija računamo pomoću

$$C_{ij}(\Delta t) \equiv \frac{\langle X_{i,t} X_{j,t+\Delta t} \rangle - \langle X_{i,t} \rangle \langle X_{j,t+\Delta t} \rangle}{\sigma_i \sigma_j}, \quad (5.3)$$

pri čemu su σ_i i σ_j standardne devijacije od X_i i X_j . Kovarijancu u brojniku računamo prema

$$\langle X_{i,t} X_{j,t+\Delta t} \rangle = \frac{1}{T} \sum_{t=1}^{T-\Delta t} X_{i,t} \cdot X_{j,t+\Delta t}, \quad (5.4)$$



Slika 5.2: Grafički prikaz prve singularne vrijednosti za pojedine zadatke i nagrade. Prozor širine 10 koraka označen je crvenim vertikalnim linijama i nalazi se na prijelazu iz zadatka u nagradu. Na slikama vidimo da se prijelaz ne događa trenutno. Prikaz 30. i 34. zadatka i nagrade su nešto drugačiji: 30. zadatak i nagrada trajali su kraće od ostalih, a na prikazu 34. zadatka i nagrade vidimo netipičan potpis u području nagrade.

pri čemu t može ići do $T - \Delta t$. Radi lakše implementacije u programski jezik python elemente matrice kroskorelacije C_{ij} definiramo kao $C_{ij} \equiv \langle r_i r_j \rangle$, a elemente r_i dobijemo tako da

od niza oduzimamo srednju vrijednost i zatim dijelimo sa standardnom devijacijom:

$$r_i(t) \equiv \frac{R_i - \langle R_i \rangle}{\sigma_i} \quad (5.5)$$

Konačno, matricu kroskorelacija možemo zapisati u obliku

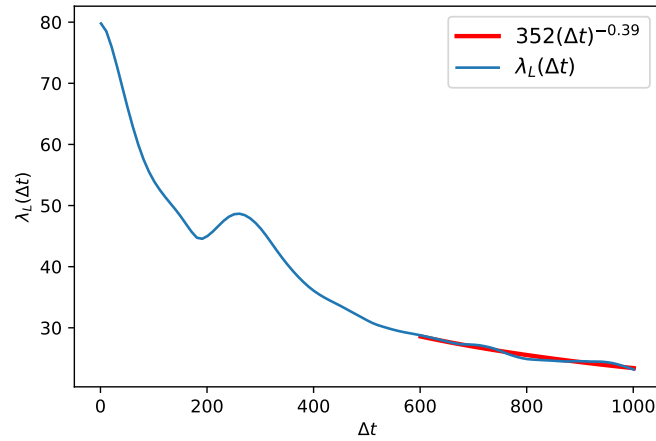
$$\mathbf{C} = \frac{1}{T} \mathbf{R} \mathbf{R}^T, \quad (5.6)$$

gdje je \mathbf{R} $N \times T$ matrica, a \mathbf{R}^T transponirana matrica \mathbf{R} . Kroskorelacije s pomaknom u vremenu radili smo pomoću funkcije `lag_CC` čiji je algoritam dan s:

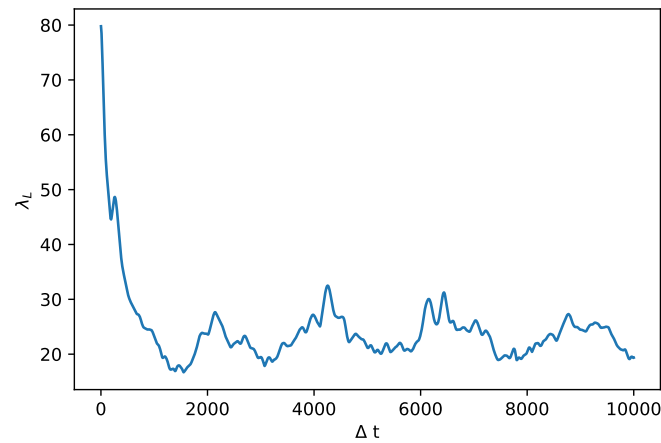
```
def lag_CC(niz,delta):
    tmp1=np.copy(niz[:, :-delta])
    tmp1 -= np.mean(tmp1, axis=1,keepdims=True)
    tmp1 /= np.std(tmp1, axis=1,keepdims=True)
    tmp2=np.copy(niz[:, delta:])
    tmp2 -= np.mean(tmp2, axis=1,keepdims=True)
    tmp2 /= np.std(tmp2, axis=1,keepdims=True)
    return np.dot(tmp1,tmp2.T)/tmp1.shape[1]
```

Kako bi izračunali kroskorelacije za različite vremenske pomake Δt , tražimo najveću singularnu vrijednost $\lambda_L(\Delta t)$ matrice kroskorelacije $C(\Delta t)$. Rezultati su prikazani na slikama 5.3 i 5.4. Na slici 5.3 primjećujemo da s vremenskim pomakom imamo pad vrijednosti maksimalne singularne vrijednosti, no u vremenskom pomaku koji odgovara duljini trajanja zadatka vidimo pojavu lokalnog maksimuma. Pad iznosa maksimalne singularne vrijednosti možemo kvalitativno tumačiti kao proces "zaboravljanja" prijašnjih vrijednosti u kolektivnom ponašanju neurona. Pojavu lokalnog maksimuma tumačimo suprotno, povećavanjem "pamćenja" prijašnjih vrijednosti. Kako smo opazili, i u individualnoj dinamici imamo promjenu na vremenskim oznakama nakon izvršenog zadatka. Na pomaku dvostruko dužem od trajnja zadatka imamo povratak u režim smanjivanja maksimalne svojstvene vrijednosti, dok na pomaku većem od trostrukog trajanja zadatka možemo aproksimirati prilagodnom na potencijsku ovisnost varijable. Rezultat takve prilagodbe daje $\lambda_L(\Delta t) = 352(\Delta t)^{-0.39}$. Slika 5.4 pokazuje iznos najveće singularne vrijednosti u ovisnosti o vremenskom pomaku za prvih 10 000 vremenskih oznaka. Jasno je izraženo zaustavljanje pada svojstvene vrijednosti, te

cikličko održavanje "memorije" vremenskog niza u kolektivnoj dinamici. Razmak između uzastopnih lokalnih maksimu raste, te nakon 10 000 vremenskih oznaka zadržava konstantnu vrijednost, što je indikacija dugodosežnih korelacija među neuronima.



Slika 5.3: Najveća singularna vrijednost u ovisnosti o vremenskom pomaku za vrijednosti pomaka do 1000 vremenskih oznaka. Nakon 600 vremenskih oznaka iznos maksimalne svojstvene vrijednosti možemo aproksimirati izrazom $\lambda_L(\Delta t) = 352(\Delta t)^{-0.39}$.



Slika 5.4: Najveća singularna vrijednost u ovisnosti o vremenskom pomaku za prvih 10 000 vremenskih oznaka. Jasno je izraženo zaustavljanje pada svojstvene vrijednosti, te cikličko održavanje "memorije" vremenskog niza u kolektivnoj dinamici.

6 Zaključak

Primjena DFA metode na vremenske serije dobivene snimanjem neurona transgeničnog miša pokazala je da 22.3% neurona ne pokazuje dugodosežne korelacije nakon točke prijelaza, odnosno njihov eksponent skaliranja α_2 je ispod granične vrijednosti koja indicira korelacije na većim vremenskim skalama. S druge strane, gotovo 80% neurona pokazuje dugodosežne korelacije. Ako pogledamo statistiku za naše podatke, ali izuzmemo neuron br.105, srednja vrijednost nagiba na kraćim vremenskim skalama α_1 iznosi $\alpha_1 = 1.65$ iz čega zaključujemo da je signal nestacionaran. Minimalna vrijednost spomenutog nagiba je 1.44, a maksimalna 1.77. Standardna devijacija iznosi 0.05. Srednja vrijednost nagiba α_2 je $\alpha_2 = 0.55$, što se slaže s činjenicom da većina neurona pokazuje dugodosežne korelacije. Minimalna vrijednost spomenutog eksponenta je 0.27, a maksimalna 0.86. Standardna devijacija iznosi 0.07. Također, maksimum distribucije točki prijelaza nalazi se na širini prozora $n = 233$, a ona se obično javlja oko 1.5 s nakon kraja zadatka.

Razmatranje prve dvije singularne vrijednosti dobivene SVD metodom pokazalo je da zadaci i nagrade pokazuju zanimljiv potpis. Grafički prikaz pokazuje grupiranje zadataka i nagrada, iz čega vidimo da je dovoljno znati samo prvu singularnu vrijednost za razlikovanje ta dva područja.

Razmatranje dinamike neurona pomoću pomičnog prozora pokazuje jasnu separaciju između zadatka i nagrade. Nagli pad u iznosu najveće singularne vrijednosti je prijelaz iz zadatka u nagradu, međutim, taj prijelaz se ne događa trenutno. To je posljedica činjenice da je nakon zadatka još potrebno doći do nagrade, te se kasnije vratiti na obavljanje zadatka.

Prikaz maksimalne singularne vrijednosti u ovisnosti o vremenskom pomaku za vrijednosti Δt do 1000 koraka, pokazuje zaboravljanje ranijih vrijednosti u kolektivnom ponašanju. No, na vremenskom pomaku koji odgovara trajanju zadatka vidimo povećanje dinamike te povezano povećanje vremenske dugodosežnosti. Nakon 600 vremenskih oznaka iznos maksimalne svojstvene vrijednosti možemo aproksimirati prilagodbom na potencijalnu ovisnost. Prikaz maksimalne singularne vrijednosti u ovisnosti o vremenskom pomaku za prvih 10 000 vremenskih oznaka pokazuje cikličko održavanje memorije vremenskog niza u kolektivnoj dinamici. Dobiveni rezultati su tek preliminarna analiza, te su u ovom području potrebna daljnja dodatna istraživanja.

7 Zahvale

Želim reći veliko hvala svom mentoru izv. prof. dr. sc. Davoru Horvatiću najprije na ohrabrenju i ukazanom povjerenju da se upustim u pisanje ovog rada, na prilici koju mi je pružio, predanom mentorstvu i inzistiranju da provedem veliki dio vremena u istraživanju. Zahvaljujem mu na brojnim savjetima i strpljenju, sudjelovanju u svim etapama nastanka ovog rada i ugodnoj radnoj atmosferi na Zavodu za teorijsku fiziku čestica i polja. Svojim nesebičnim i susretljivim pristupom predstavlja mi uzor kako u akademskom, tako i u osobnom razvoju. Od srca mu želim zahvaliti na entuzijazmu, stručnim savjetima i pristupu koji su mi bili izuzetno važni tijekom istraživanja i pisanja ovog rada.

Zahvaljujem se i svojoj obitelji i Kristianu na velikoj podršci i razumijevanju.

Dodaci

Dodatak A Implementacija DFA metode

Jedan način modeliranja vremenskih serija s dugodosežnim autokorelacijama je ARFIMA proces. Vremensku seriju x_t možemo izračunati pomoću

$$x_t = \sum_{n=1}^{\infty} a_n(d)x_{t-n} + \varepsilon_t. \quad (\text{A.1})$$

Parametar skaliranja d ima vrijednosti između -0.5 i 0.5, a ε_t predstavlja nezavisno i jednoliko raspodjeljene slučajne varijable. Srednja vrijednost $\langle \varepsilon_t \rangle = 0$ i varijanca $\langle \varepsilon_t^2 \rangle = 1$. Težinski faktor $a_n(d)$ definiramo kao:

$$a_n(d) = \frac{d\Gamma(n-d)}{\Gamma(1-d)\Gamma(n+1)}. \quad (\text{A.2})$$

Γ označava gama funkciju, a n je vremenska skala. U python programskom jeziku težinski faktor $a_n(d)$ implementirali smo na sljedeći način:

```
def a_coef(n,d):  
    if n<50:  
        tmp=(d*special.gamma(n-d))/(special.gamma(1-d)*special.gamma(n+1))  
    else:  
        tmp=(d/special.gamma(1-d))*np.exp((1+d)+  
            (-0.5-n)*np.log(n)+(-0.5-d+n)*np.log((-1-d+n)))  
    return tmp
```

Zatim smo pomoću naredbe `np.random.normal` generirali 1000 neovisno i jednoliko distribuiranih varijabli, sa srednjom vrijednošću 0 i varijancom 0.333, te rezultat pohranili u varijablu `s`:

```
s = np.random.normal(0, 0.333, 1000)
```

Sada računamo težinski faktor za svaku varijablu. Vrijednost parametra d je $d = 0.25$:

```
ad=np.flip([a_coef(i,0.25) for i in range(1,1001)],0)
```

Obzirom da ćemo u sljedećem koraku uzeti zadnjih 1000 vrijednosti, moramo okrenuti redoslijed brojeva kako bi se oni ispravno množili. Zatim pomnožimo svaki težinski koeficijent s pripadajućim članom niza, te dodajemo jednu slučajnu varijablu iz niza ε_t . Dobivene vrijednosti x_{tmp} dodajemo na s i sve spremamo u novo polje s .

```
for j in range(15000):  
    xtmp=np.dot(ad,s[-1000:])+np.random.normal(0,1)  
    s=np.append(s,xtmp)
```

Vidjeli smo da numerički ima smisla računati samo prvih nekoliko tisuća težinskih koeficijenata. Iz tog razloga moramo dodati onoliko inicijalnih točaka koliko imamo težinskih koeficijenata. Na taj način izračunali smo više točaka nego što je potrebno za naše računanje i stvorili smo "memoriju", međutim, taj višak u konačnom računu odbacujemo:

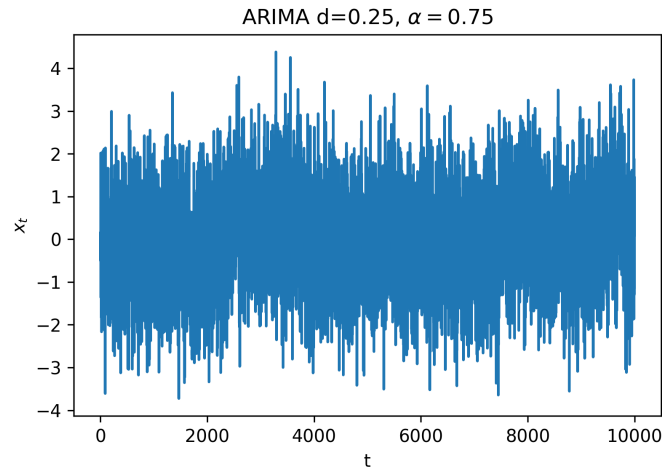
```
s = s[6000:]
```

Dobiveni vremenski niz pohranili smo u varijablu `dat1`:

```
dat1=s
```

Zatim smo napravili niz cijelih brojeva od 1 do 10 000, jer je veličina naših podataka 10 000 (dodajemo 1, jer zadnji broj nije obuhvaćen), te nacrtali vremenski niz koji smo dobili ranije objašnjenim procesom ARFIMA (slika A.1).

```
x = range(1,size(dat1)+1)  
plt.plot(x, dat1)  
plt.title('ARIMA d=0.25,  $\alpha=0.75$ ')  
plt.xlabel('t')  
plt.ylabel('$x_t$')
```



Slika A.1: Vremenski niz dobiven ARFIMA procesom. Parametar skaliranja je $d = 0.25$

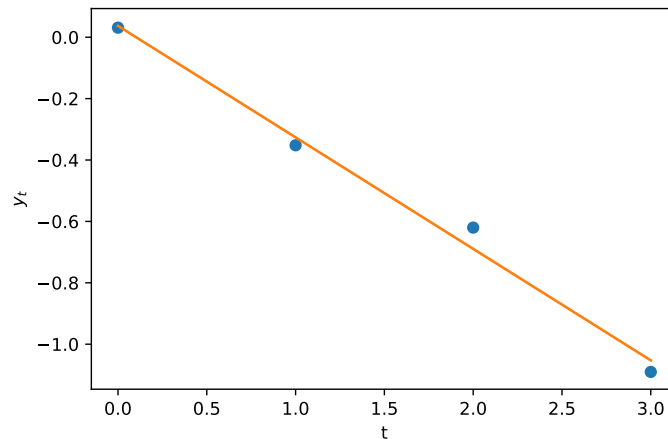
Sada od svake pojedine vrijednosti oduzmemo srednju vrijednost i radimo kumulativnu sumu:

```
dat1m=np.cumsum(dat1-mean(dat1)).
```

Grafički prikaz rezultata dan je na slici 3.5 u glavnom dijelu teksta.

```
x = range(1,size(dat1)+1)
plt.plot(x, dat1m)
plt.title('ARIMA d=0.25 pripremljeno za DFA metodu')
plt.xlabel('t')
plt.ylabel('$y_t$')
```

Kvadrat funkcije fluktuacije za primjer smo računali za veličinu prozora $n = 4$. Trend za određeni prozor dobivamo tako da kroz odabrane točke provučemo pravac dobiven metodom najmanjih kvadrata, slika A.2. No znamo da postupak radimo za sve širine prozora, pri čemu je najmanja širina prozora dana s $n = 2q + 2$, gdje je q red DFA (DFA- q), a najveća širina prozora odgovara $1/4$ duljine niza. U sljedećem dijelu koda odredili smo širine prozora za koje smo računali kvadrat funkcije fluktuacije, no treba napomenuti da gledamo logaritamsku skalu. Funkcijom `np.logspace` smo u 20 točaka odredili jednako udaljene vrijednosti između početne i konačne točke (širine prozora). Obzirom da dobivene vrijednosti nisu cijeli brojevi, u sljedećem retku smo vrijednosti zaokružili na najmanji cijeli broj



Slika A.2: Prozor od prve 4 točke niza y_t i pravac dobiven metodom najmanjih kvadrata koji u navedenom segmentu predstavlja trend

koji je veći od dane vrijednosti (npr. ako dobijemo broj 5.3, zaokružimo ga na 6). Konačno, rezultate pohranimo u `scale` i napravimo polje (`np.array(scale)`).

```
rs=np.logspace(log10(2*1+2),log10(len(dat1)/4),20)
scale=[int(ceil(rs[i])) for i in range(len(rs))]
np.array(scale)
```

Sada DFA funkciju pozivamo na naše podatke za sve širine prozora. Rezultate pohranimo u varijablu `dfao1`.

```
dfao1=[dfa(dat1m,scale[i]) for i in range(len(scale))]
```

Informaciju o korelacijama daje nam eksponent skaliranja α , koji predstavlja nagib pravca u log-log grafu. Prema tome na naše podatke moramo prilagoditi pravac. Funkcija `np.polyfit` daje nam koeficijente polinoma, odnosno nagib a i odsječak na y -osi, b , a pomoću funkcije `np.polyval` izračunali smo vrijednosti polinoma u danim točkama.

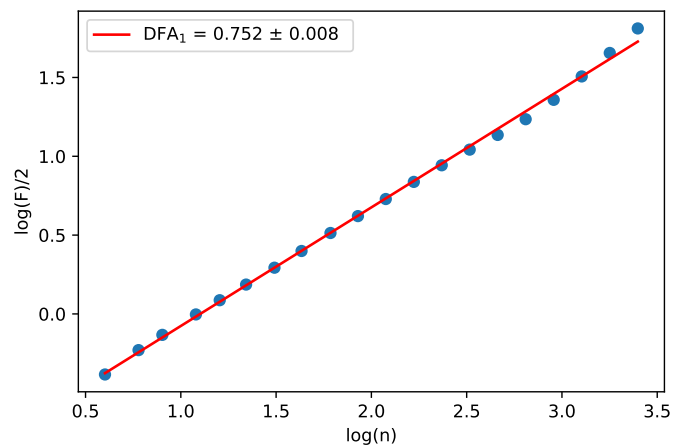
```
pf1=np.polyfit(log10(scale), log10(dfao1)/2.,1)
yp1=np.polyval(pf1,log10(scale))
```

Pogrešku možemo izračunati na sljedeći način:

```
sl1=stats.linregress(log10(scale), log10(dfao1)/2)
```

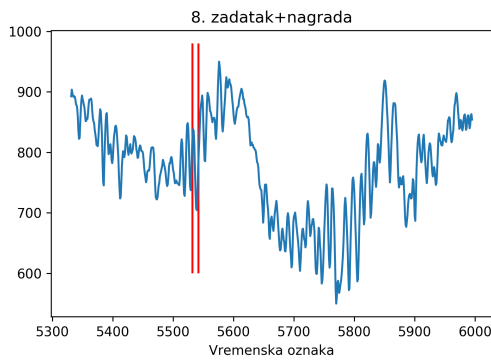
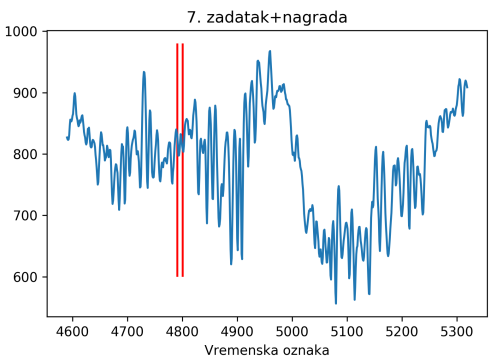
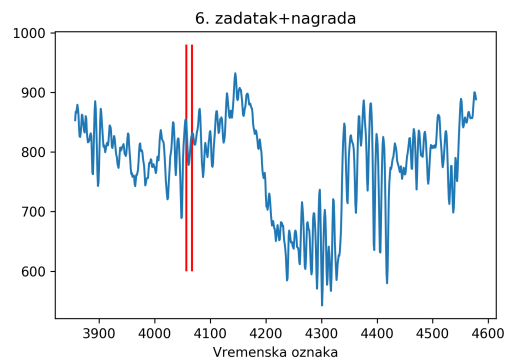
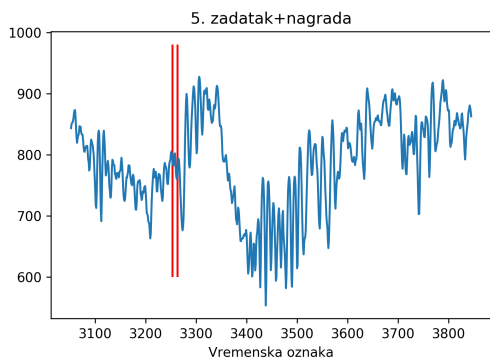
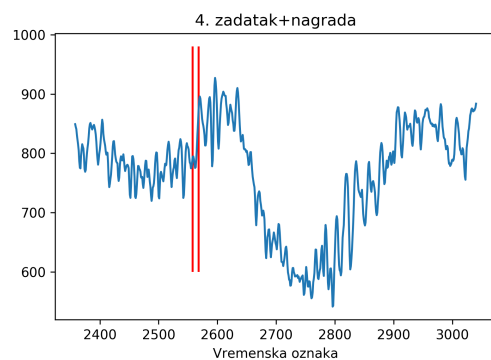
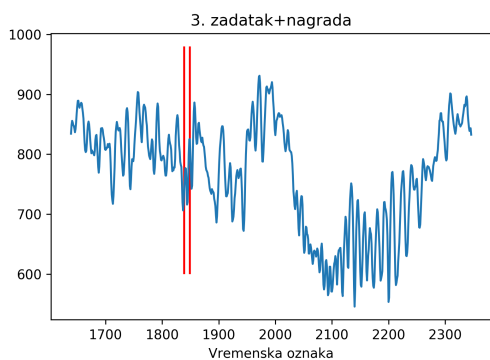
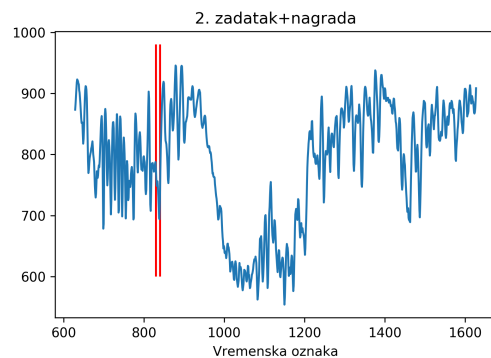
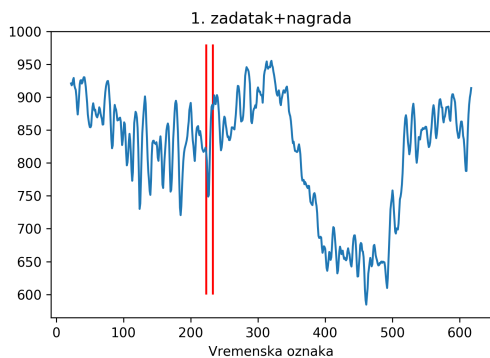
Konačno, grafički prikažemo naše podatke i pravac prilagodbe. Rezultat je prikazan na slici A.3.

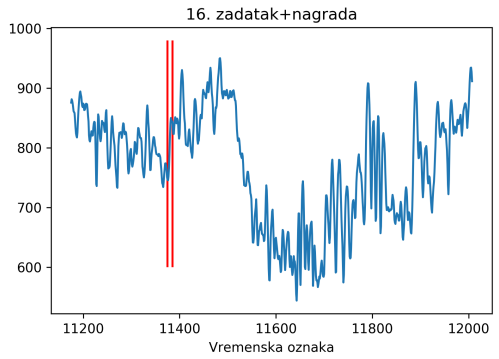
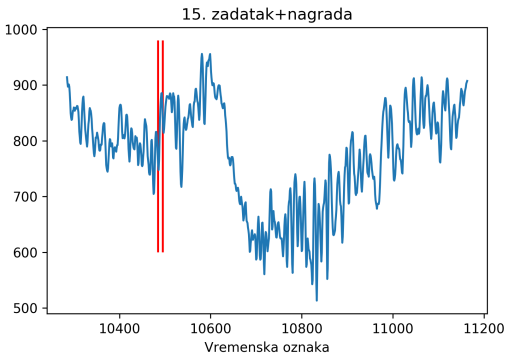
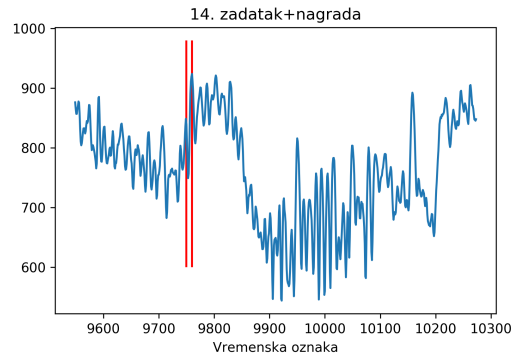
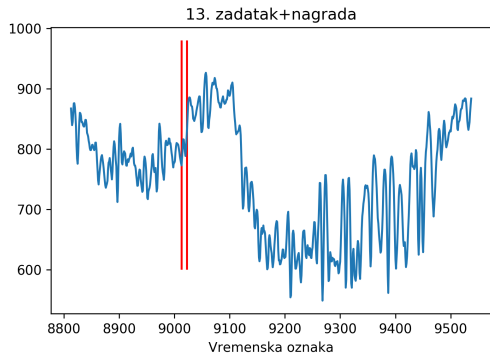
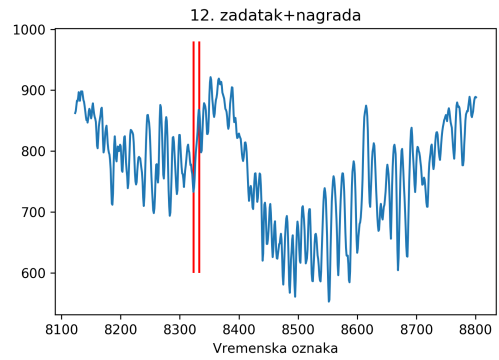
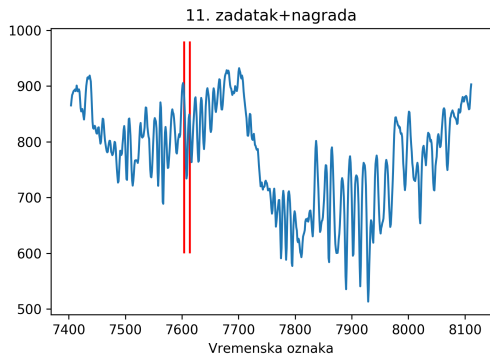
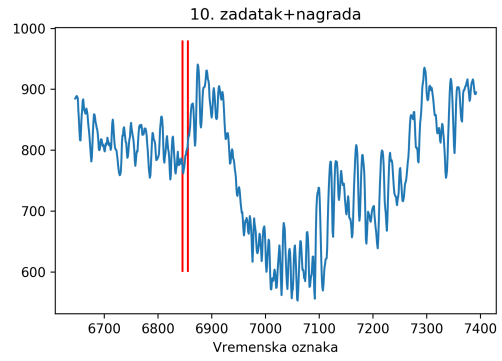
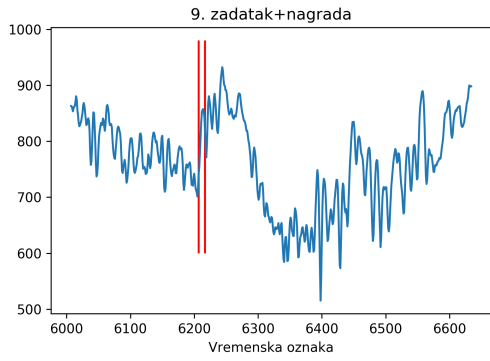
```
plt.plot(log10(scale), log10(dfao1)/2.,marker='o',linestyle='None')
plt.plot(log10(scale), yp1,linestyle='-',color='r',
        label='DFA $\{ \}_1$  = %.3f  $\pm$  %.3f ' % (pf1[0], s11[4]))
plt.xlabel('log(n)')
plt.ylabel('log(F)/2')
plt.legend(loc=2)
```

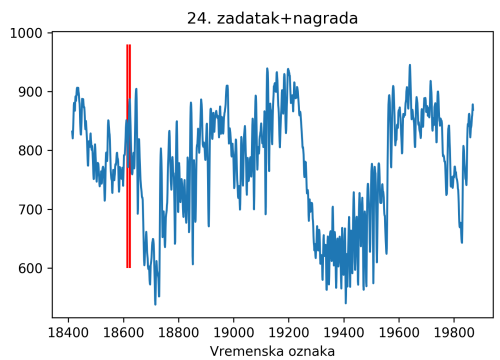
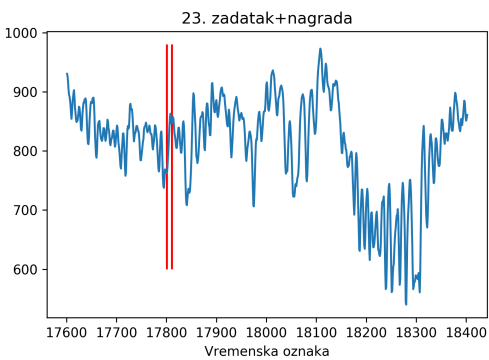
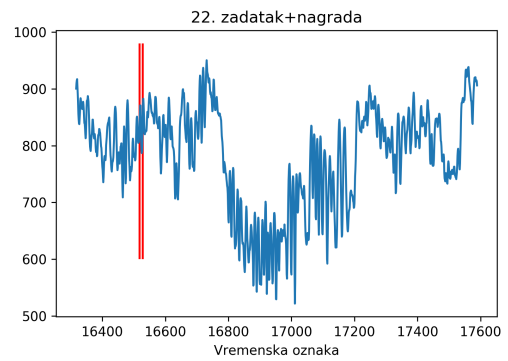
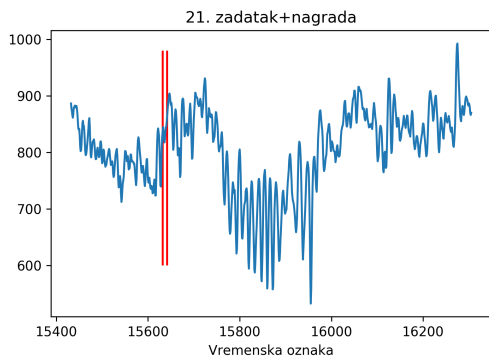
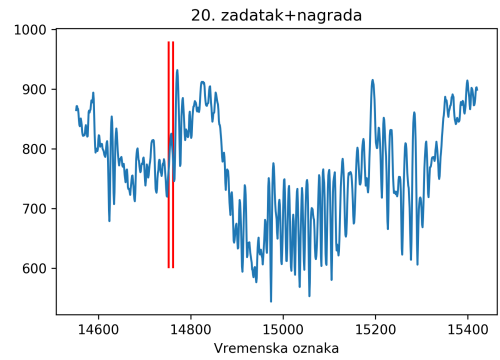
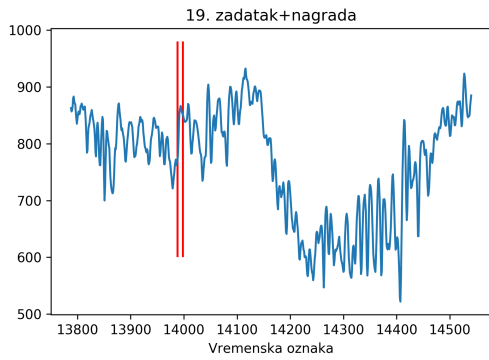
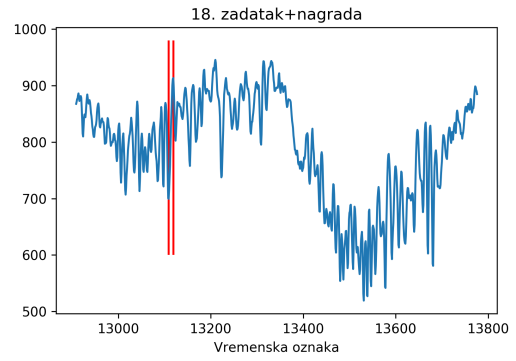
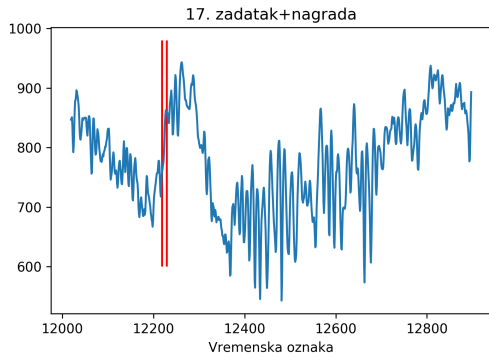


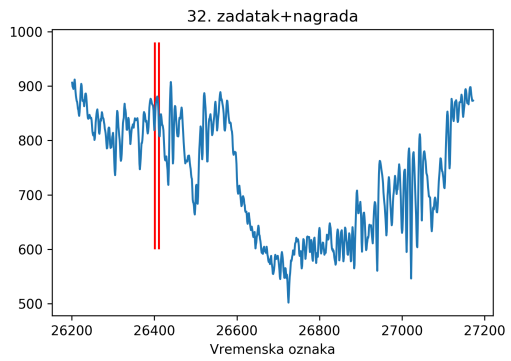
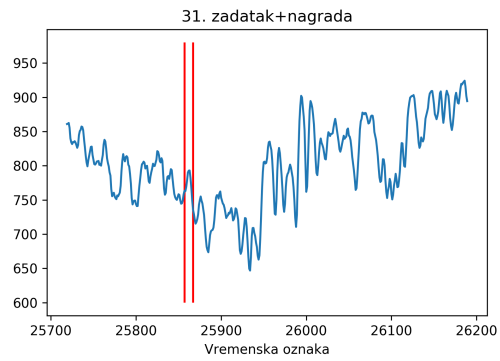
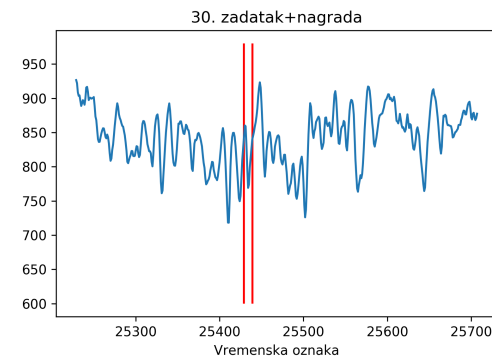
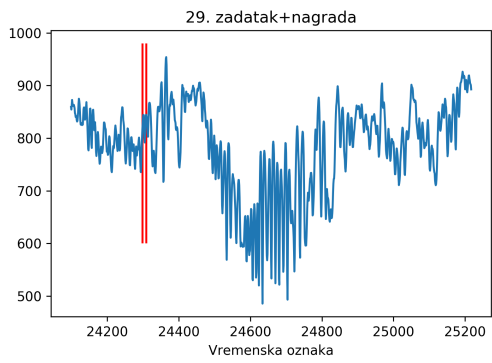
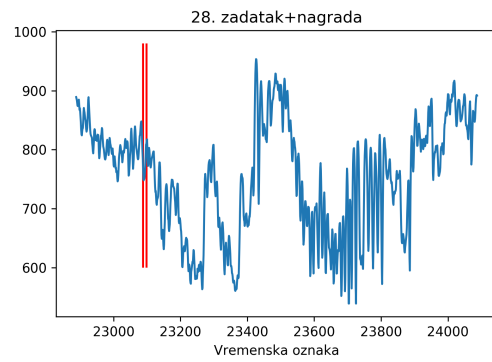
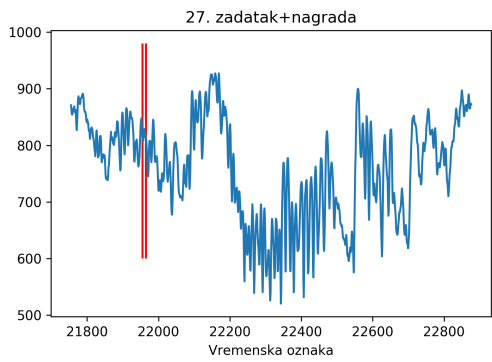
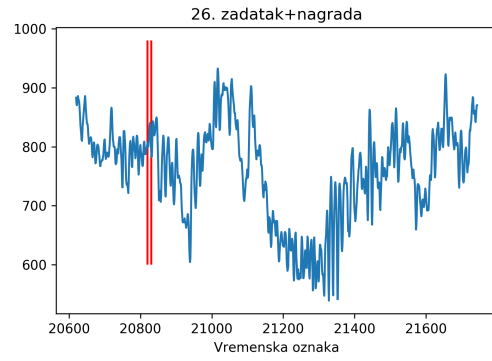
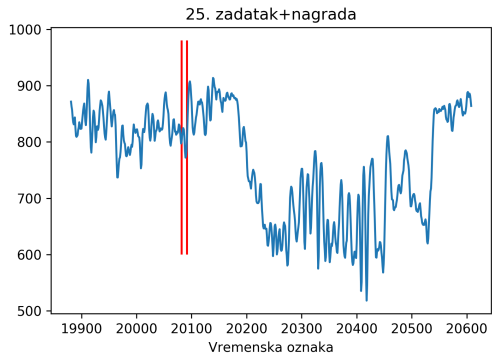
Slika A.3: Rezultat DFA metode za vremenski niz x_t generiran ARFIMA procesom

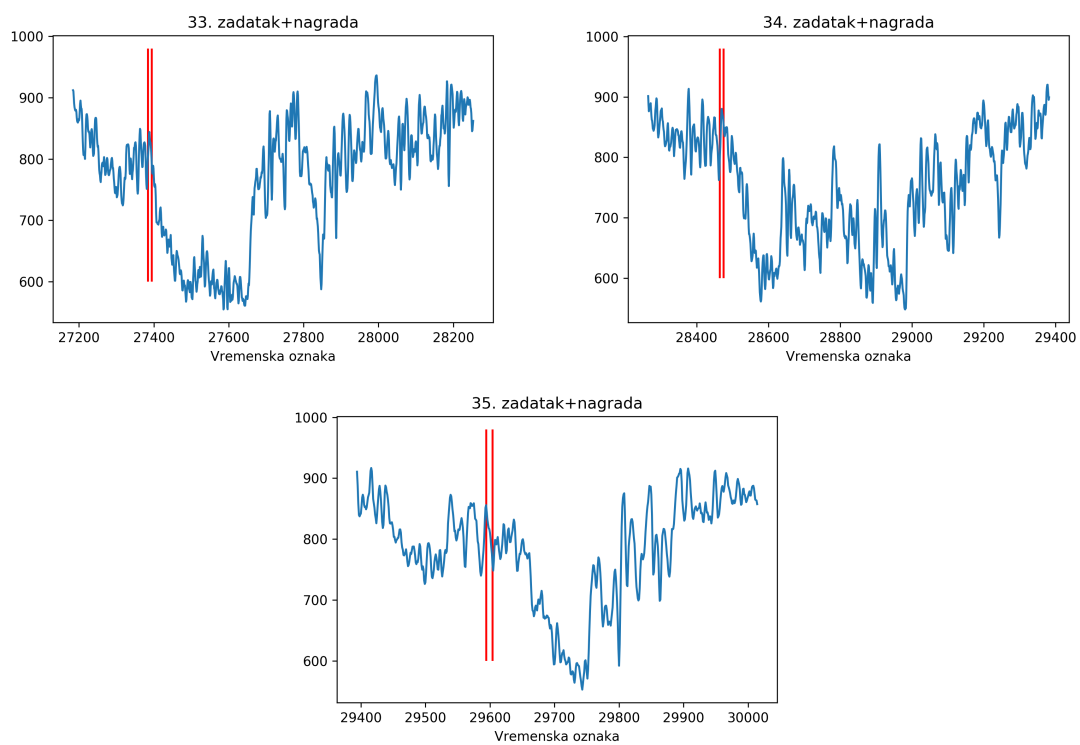
Dodatak B Pregled zadatak/nagrada segemenata za cijeli vremenski niz











Slika B.1: Grafički prikaz prve singularne vrijednosti za sve zadatke i nagrade. Prozor širine 10 koraka označen je crvenim vertikalnim linijama i nalazi se na prijelazu iz zadatka u nagradu.

Literatura

- [1] Leenoy, M. et al.; Collective Behavior of Place and Non-place Neurons in the Hippocampal Network // *Neuron*, Vol. 96, 5, (2017) str. 1178.
- [2] Horvatic, D.; Stanley, H. E.; Podobnik, B.; Detrended Cross-Correlation Analysis for Non-Stationary Time Series with Periodic Trends. // *EPL*, Vol. 94, (2011), str. 18007.
- [3] Podobnik, B.; Horvatic, D.; Lam Ng, A.; Stanley, H. E.; Ivanov, P.Ch. Modeling long-range corss-correlations in two component ARFIMA and FIARCH processes. // *Physica A*, Vol. 387, (2008), str. 3954.
- [4] Kantelhardt, J.W. et al. Detecting long-range correlations with detrended fluctuation analysis. // *Physica A*, Vol. 295, (2001), str. 441.
- [5] Bryce, R.M.; Sprague, K.B. Revisiting detrended fluctuation analysis. // *Sci. Rep.* Vol. 2, (2012), str. 315.
- [6] Nylén, E. L.; Wallisch, P. *Neural data science : A Primer with MATLAB and Python.* Academic Press, 2017.
- [7] Peng, C.-K.; Havlin, S.; Stanley, H. E.; Goldberg, Ary L. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. // *Chaos*, Vol. 5, (1995) str. 82.
- [8] One-dimensional smoothing spline <https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.UnivariateSpline.html>, 1.4.2018.
- [9] Wigner, E.; Characteristic vectors of bordered matrices with infinite dimensions. // *Annals of Mathematics*. Vol. 62, 3, (1955), str. 548.
- [10] Sensoy, A.; Yuksel, S.; Erturk, M. Analysis of cross-correlations between financial markets after the 2008 crisis. // *Physica A*, Vol. 392, (2013), str. 5027.
- [11] Sompolinsky, H.; Crisanti, A.; Sommers, H.; Chaos in Random Neural Networks. // *Physical Review Letters*, Vol. 61, 3, (1988), str. 259.
- [12] Géron, A. *Hands-On Machine Learning with Scikit-Learn & TensorFlow: Concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, 2017.

- [13] Dimensionality reduction using truncated SVD <http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html>, 1.4.2018.
- [14] Shlens, J. A Tutorial on Principal Component Analysis. // *arXiv:1404.1100*
- [15] Wang, D.; Zhang, X.; Horvatic, D.; Podobnik, B.; Stanley, H. E. A generalization of random matrix theory and its application to statistical physics. // *Chaos* Vol. 27, (2017), str. 023104.
- [16] Podobnik, B.; Wang, D.; Horvatic, D.; Grosse, I.; Stanley, H. E. Time-lag cross-correlations in collective phenomena. // *Europhysics Letters* 90, (2010), str. 68001.

Sažetak

Individualna i kolektivna dinamika neurona

Maria Krajačić

Glavna komponenta vezana uz procese pamćenja i prostornu memoriju u mozgu ljudi i kralježnjaka je hipokampus. Neki hipokampalni neuroni aktiviraju se kada se životinja nađe na određenom mjestu u prostoru. Mjerenjem te aktivnosti dobivamo vremenske serije koje se mogu analizirati metodama statističke fizike. U ovom radu podatke smo najprije analizirali metodom fluktuacija kojima je uklonjen trend, jer navedena metoda može detektirati dugodosežne korelacije u nestacionarnim uvjetima. Skoro 80% hipokampalnih neurona pokazuje dugodosežne korelacije. Izračun kroskorelacija svih neurona te rastav na singularne komponente pokazuje da nam je dovoljna samo prva singularna vrijednost za razlikovanje dinamike neurona povezanom s obavljanjem zadatka i nagrade. Prikazom najveće singularne vrijednosti u ovisnosti o vremenskom pomaku pokazujemo da postoji cikličko održavanje memorije vremenskog niza u kolektivnoj dinamici. Fenomen prijelaza koji je opažen u individualnoj dinamici dešava se u vremenu koje odgovara završetku obavljanja zadatka, a jednako tako u kolektivnoj dinamici nakon tog vremena opažamo povećanu kolektivnu dinamiku neurona.

Ključne riječi: analiza fluktuacija kojima je uklonjen trend, fenomen prijelaza, kroskorelacije s vremenskim pomakom, neuralna dinamika

Summary

Individual and collective neural dynamics

Maria Krajačić

The main component associated with memory processes and spatial memory in the brain of humans and vertebrates is a hippocampus. Some hippocampal neurons are activated when the subject is found in a certain location. By measuring this activity, we get time series of individual neural cell activity that can be analyzed by methods of statistical physics. In this paper, we first analyzed the data using detrended fluctuation analysis, because this method can detect long-range correlations in nonstationary time series. Almost 80% hippocampal neurons show long-range correlations. Using crosscorrelation matrix of the neural time series and calculating singular components shows that only the first singular value for the neuronal dynamics can be used to separate task from reward. Time-lag dependence of the largest singular value exhibits cyclical persistence the time series memory in collective neural dynamics. The transition phenomena observed in the individual neurons occurs at the time that corresponds to the completion of the task, and equally in the collective dynamics after that time, we observe the increased collective dynamics of the neurons.

Keywords: detrended fluctuation analysis, crossover phenomena, time-lag crosscorrelations, neural dynamics

Životopis

Maria Krajačić rođena je 29.9.1993. u Zagrebu. Živi u Karlovcu, gdje je pohađala osnovnu školu, Opću gimnaziju Karlovac, te Glazbenu školu Karlovac. 2011. godine maturirala je flautu s odličnim uspjehom, a 2012. godine završava opću gimnaziju s vrlo dobrim uspjehom. Iste godine upisuje integrirani preddiplomski i diplomski sveučilišni studij fizike na Prirodoslovno-matematičkom fakultetu u Zagrebu, gdje je studentica 5. godine. Stipendist je grada Karlovca i demonstrator na Zavodu za eksperimentalnu fiziku.