



Matematički fakultet

Univerzitet u Beogradu

**Iterativni postupak za pronalaženje nizova sa dobrim autokorelacionim  
osobinama**

master rad

**Autor:**

Jasna Kovačević

**Članovi komisije:**

prof. dr Miodrag Živković, mentor

dr Zorica Stanimirović

dr Saša Malkov

Beograd

2013. godina

## Sadržaj

Spisak slika .....	IV
Spisak tabela .....	VI
Predgovor .....	VII
1 Uvod .....	1
1.1 Autokorelacije .....	2
1.2 Metod naizmeničnih projekcija.....	4
2 Algoritmi ITROX- $\mathcal{P}$ i ITROX- $\mathcal{AP}$ .....	6
2.1 Uvijeni proizvod vektora .....	6
2.2 Algoritam ITROX- $\mathcal{P}$ .....	7
2.3 Algoritam ITROX- $\mathcal{AP}$ .....	10
2.4 Pronalaženje samo jednog niza .....	12
2.5 ITROX algoritam za nizove nad konačnim ili diskretnim azbukama.....	13
3 Programski jezik Python.....	15
3.1 Sintaksa .....	15
3.2 Strukture podataka .....	15
3.3 Kontrola toka.....	16
3.4 Funkcije .....	17
3.5 Klase .....	17
3.6 Rad sa bibliotekama.....	18
3.7 Ulaz i izlaz .....	18
4 Implementacija algoritama ITROX-P i ITROX-AP .....	19
4.1 Grafičko okruženje .....	19
4.2 Klasa ITROXClass .....	20
4.2.1 Implementacija funkcija $\mathcal{F}$ i $\mathcal{F}'$ .....	20
4.2.2 Implementacija naizmeničnih projekcija .....	22
4.2.3 Generisanje početne matrice.....	27
4.2.4 Procedure „ITROXP“ i „ITROXAP“ .....	27
5 Rezultati rada programa ITROX.....	29
5.1 Brzina izvršavanja različitih implementacija programa ITROX.....	29
5.2 Testiranje programa ITROX.....	30
Dodatak A.....	32
Dodatak B.....	40

Zaključak .....	41
Literatura .....	42

## Spisak slika

Slika 1.1 Naizmenečne projekcije kada je presek skupova A i B neprazan	5
Slika 1.2 Naizmenečne projekcije kada je presek skupova A i B prazan	5
Slika 4.1 Šema preslikavanja $\mathcal{F}$	21
Slika 4.2 Šema preslikavanja $\mathcal{F}$ sa šiftovanjem kolona	21
Slika 4.3 Šema preslikavanja $\mathcal{F}^1$	22
Slika 4.4 Šema preslikavanja $\mathcal{F}^1$ sa šiftovanjem redova	22
Slika A.1 Raspodela PSL i ISL za 100 realnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	32
Slika A.2 Realni niz dužine 64 sa dobrim periodičnim autokorelacionim osobinama	32
Slika A.3 PSL i ISL za realni niz sa slike A.2	32
Slika A.4 Raspodela PSL i MF za 100 binarnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	33
Slika A.5 Binarni niz dužine 64 sa dobrim periodičnim autokorelacionim osobinama	33
Slika A.6 PSL i MF za binarni niz sa slike A.5	33
Slika A.7 Raspodela PSL i ISL za 100 realnih parova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	34
Slika A.8 Par realnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	34
Slika A.9 PSL i ISL za par sa slike A.8	34
Slika A.10 Raspodela PSL i MF za 100 binarnih parova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	35
Slika A.11 Par binarnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama	35
Slika A.12 PSL i MF za par sa slike A.11	35
Slika A.13 Raspodela PSL i ISL za 100 realnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	36
Slika A.14 Realni niz dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	36
Slika A.15 PSL i ISL za realni niz sa slike A.14	36
Slika A.16 Raspodela PSL i MF za 100 binarnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	37

Slika A.17 Binarni niz dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	37
Slika A.18 PSL i MF za niz sa slike A.17	37
Slika A.19 Raspodela PSL i ISL za 100 realnih parova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	38
Slika A.20 Par realnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	38
Slika A.21 PSL i ISL za par sa slike A.20	38
Slika A.22 Raspodela PSL i MF za 100 binarnih parova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	39
Slika A.23 Par binarnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama	39
Slika A.24 PSL i MF za par sa slike A.23	39
Slika B.1 Najbolji PSL za binarne nizove sa dobrim periodičnim autokorelacionim osobinama	40
Slika B.2 Najbolji PSL za binarne parove sa dobrim periodičnim autokorelacionim osobinama	40
Slika B.3 Najbolji PSL za binarne nizove sa dobrim aperiodičnim autokorelacionim osobinama	40
Slika B.4 Najbolji PSL za binarne parove sa dobrim aperiodičnim autokorelacionim osobinama	40

## **Spisak tabela**

Tabela 1.1 Barkerovi nizovi	4
Tabela 4.1 Oznake primera korišćenja programa ITROX	29
Tabela 4.2 Brzine različitih implementacija (u sekundama)	30
Tabela 4.3 Pregled najboljih rezultata testa A	31

## Predgovor

U master radu realizuju se i analiziraju algoritmi za pronalazenje nizova sa dobrim autokorelacionim osobinama opisani u radu [1].

Prvo poglavlje sadrži kratak pregled pojmova autokorelacije i dobrog autokorelacionog svojstva i pojam metode naizmeničnih projekcija.

Drugo poglavlje daje opis algoritama ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  predložene u radu [1].

Treće poglavlje opisuje programski jezik Python koji je korišćen za realizaciju algoritama ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$ .

Četvrto poglavlje se bavi opisom implementacije algoritama ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  korišćenjem programskog jezika Python.

Peto poglavlje prikazuje rezultate korišćenja programa ITROX dobijene kroz dva različita testa.

Zahvaljujem se profesoru Miodragu Živkoviću i članovima komisije dr Zorici Stanimirović i dr Saši Malkovu na pomoći prilikom formulisanja teme kao i na sugestijama i primedbama prilikom izrade ovog rada.

## 1 Uvod

Proračun autokorelacija nalazi primenu u raznim oblastima komunikacionih i informacionih tehnologija kao na primer: kod radarskih i sonarnih signala, sinhronizacija, u oblasti kriptografije i bezbedonosnih sistema [2]. U oblasti mobilnih komunikacija sve je veća potreba za pronalaženjem signala sa dobrim autokorelacionim osobinama koji se koriste u CDMA (Code Division Multiple Access) sistemima gde više korisnika može istovremeno da prenosi podatke u istom frekvencijskom opsegu. CDMA koristi različite PN kodove ("Pseudo Random Noise Code") koji imaju dobre autokorelacione osobine. Najčešće se koriste m-sekvence, Goldovi kodovi, Kasami kodovi i Barkerovi kodovi ([2] str. 402-421).

Pronalaženje niza sa optimalnim autokorelacionim osobinama je jako teško već za relativno kratke nizove. Zbog toga su razvijeni algoritmi koji daju približno optimalne rezultate kao što su: Brenner 1998; Schotten & Luke 2005; Borwein & Ferguson 2005. ([5] str. 21). Međutim, ovi algoritmi postaju veoma neefikasni za nizove velike dužine i mogu se koristiti za nizove dužine  $N \sim 10^2$  ([5] str. 21). Razvijeni su i neki ciklični algoritmi kao što su algoritmi za minimiziranje ISL metrike (ili ISL srodne metrike) nad unimodularnim nizovima kao što su CA i CAP (bazirani na dekompoziciji po singularnim vrednostima (mogu se koristiti za nizove dužine  $N \sim 10^3$ )); CAN (CA-New) (mogu se koristiti za nizove dužine  $N \sim 10^6$ ) i WeCAN (mogu se koristiti za nizove dužine  $N \sim 10^4$ ) (bazirani na brzom Furijeovoj transformaciji) ([6] str. 1416), CAD (CA direct) [7], PeCAN [8].

Pored toga, potrebno je pronaći i skupove međusobno ortogonalnih nizova sa optimalnim autokorelacionim osobinama kao što su m-sekvence i Walsh-Hadamard kodovi koji se najčešće koriste u „Synchronous CDMA“ kada su svi korisnici sinhronizovani, a kod „Asynchronous CDMA“ koriste se PN kodovi kao što su Goldovi kodovi.[14] Kada se signal šalje samo jednom potrebni su kodovi sa dobrim aperiodičnim autokorelacionim osobinama, a njih je teže generisati od nizova sa dobrim periodičnim autokorelacionim osobinama.[1] Za pronalaženje takvih skupova nizova razvijeni su ciklični algoritmi kao što su Multi-CAN, Multi-WeCAN i Multi CAO kao proširenja CAN, WeCAN i CA algoritama respektivno (videti [5] str. 48-79).

Pored generisanja ortogonalnih skupova razvijeni su algoritmi za pronalaženje skupova komplementarnih nizova. Komplementarne parove binarnih nizova prvi put pominje Marcel Golay 1949.[3] Od značaja su i međusobno ortogonalni skupovi komplementarnih nizova. Primenjuju se u oblastima fizike, kombinatorike, telekomunikacija (sinhronizacija, sistemi proširenog spektra i OFDM (Orthogonal Frequency Division Multiplexing ) sistemima). [1]

Algoritmi nazvani ITROX koji se obrađuju u ovom radu mogu se koristiti za pronalaženje niza ili skupa (periodično ili aperiodično) komplementarnih nizova sa dobrim autokorelacionim osobinama. Koriste metod naizmeničnih projekcija i bazirani su na aproksimaciji matrica korišćenjem uvijenog proizvoda. Kao pomenuti ciklični algoritmi i ITROX algoritmi minimiziraju ISL metriku i ne postavljaju ograničenja u vezi sa dužinom nizova koje je potrebno pronaći. [1]

U ovom poglavlju će biti ukratko pojašnjeni pojmovi autokorelacija (periodičnih i aperiodičnih) i metode naizmeničnih projekcija.



## 1.1 Autokorelacije

U ovom delu biće navedene definicije korelacije, međukorelacije, autokorelacije i skupa komplementarnih nizova (videti [2] str. 1-9, [1] i [3]).

**Definicija 1.1.1.** Neka je  $a = \begin{bmatrix} a_1 \\ \vdots \\ a_n \end{bmatrix}$  i  $b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$  vektori iz  $\mathbb{C}^{n \times 1}$ . Korelacija dva vektora dužine  $n$  definisana je sledećim izrazom:

$$C(a, b) = \frac{(a, b)}{|a||b|} = \frac{\sum_{i=1}^n a_i b_i^*}{\sqrt{\sum_{i=1}^n |a_i|^2} \sqrt{\sum_{i=1}^n |b_i|^2}}$$

gde je  $b_i^*$  konjugat kompleksnog broja  $b_i$ .

Odavde,  $C(a, b) = \frac{a}{|a|} \cdot \frac{b}{|b|}$ , tj. korelacija vektora  $a$  i  $b$  jednaka je njihovom normalizovanom skalarnom proizvodu. Ako je  $(a, b) = |a||b| \cos \theta$ , tada je  $C(a, b) = \cos \theta$  što znači da je korelacija ortogonalnih vektora jednaka nuli.

U oblasti obrade signala korelacija dva niza  $x$  i  $y$  dužine  $n$  se računa kao skalarni proizvod vektora koji imaju vrednosti članova nizova  $x$  i  $y$  (bez normalizacije) pa se korelacija može definisati i na sledeći način:

**Definicija 1.1.2.** Neka su  $X = \{x_i\}$  i  $Y = \{y_i\}$  za  $1 \leq i \leq n$  nizovi dužine  $n$  iz  $\mathbb{C}$ . Korelacija dva niza dužine  $n$  definisana je sledećim izrazom:

$$C(X, Y) = (X, Y) = \sum_{i=1}^n x_i y_i^*$$

**Definicija 1.1.3.** Ako posmatramo niz  $x$  dužine  $n$  tada:

a) Periodična autokorelacija niza  $X = \{x_i\}$  za  $1 \leq i \leq n$  je niz korelacija niza  $X$  i nizova  $X_k = \{x_{(i+k) \bmod n}\}$ ,  $1 \leq i \leq n$  za  $0 \leq k \leq n-1$ , tj.

$$c_k = \sum_{l=1}^n x_l x_{(l+k) \bmod n}^* = c_{-k}, 0 \leq k \leq (n-1)$$

b) Aperiodična autokorelacija niza  $X = \{x_i\}$  za  $1 \leq i \leq n$  je niz korelacija niza  $X$  i nizova  $X_k = \{x_{i+k}\}$ ,  $1 \leq i \leq n$  za  $0 \leq k \leq n-1$ , gde je  $x_l = 0, l > n$ , tj.

$$r_k = C(X, X_k) = \sum_{l=1}^{n-k} x_l x_{(l+k)}^* = r_{-k}, 0 \leq k \leq (n-1)$$

Vrednosti  $c_0$  i  $r_0$  se nazivaju energetska komponentom niza  $x$  i obeležavaju se sa  $E$  što označava energiju diskretnog signala  $X$  ( $E = \sum_{l=1}^n x_l x_l^*$ )

**Definicija 1.1.4.** Neka je  $S = \{x_1, x_1, \dots, x_m\}$  konačan skup nizova dužine  $n$ . Neka su  $c_{1,k}, c_{2,k}, \dots, c_{m,k}$  periodične autokorelacije  $k$ -tog reda redom nizova  $x_1, x_2, \dots, x_m$ . Skup  $S$  se naziva skupom periodično komplementarnih nizova kada:

$$\sum_{l=1}^m c_{l,k} = 0, 0 \leq k \leq (n-1)$$

Slično se definiše skup aperiodično komplementarnih nizova.

**Definicija 1.1.5.** Skup dva komplementarna niza dužine  $n$  se naziva (periodično ili aperiodično) komplementarnim parom.

Kada niz ima sve necentralne autorelacije (periodične ili aperiodične) jednake nuli, tada se za taj niz kaže da je savršen. Pored savršenih nizova od značaja su i nizovi sa dobrim periodičnim, odnosno aperiodičnim autokorelacionim osobinama gde su vrednosti  $c_1, c_2, \dots, c_n$ , odnosno  $r_1, r_2, \dots, r_n$  male u odnosu na neku izabranu metriku. U tu svrhu obično se koriste sledeće metrike:

- maksimalna necentralna korelacija (peak sidelobe level):  
 $PSL = \max\{|c_k|\}_{k=1}^{n-1}$  (periodične) ili  $PSL = \max\{|r_k|\}_{k=1}^{n-1}$  (aperiodične)
- integrisana necentralna koralacija (integrated sidelobe level)  
 $ISL = \sum_{k=1}^{n-1} |c_k|^2$  (periodične) ili  $ISL = \sum_{k=1}^{n-1} |r_k|^2$  (aperiodične)
- faktor kvaliteta (merit factor)  
 $MF = \frac{|c_0|^2}{2 \sum_{k=1}^{n-1} |c_k|^2}$  (periodične) ili  $MF = \frac{|r_0|^2}{2 \sum_{k=1}^{n-1} |r_k|^2}$  (aperiodične)  
 Odnosno,  $MF = \frac{E^2}{2ISL}$ .

PSL, ISL i MF se za skup komplementarnih nizova definišu sledećim formulama:

- $PSL = \max\{|\sum_{l=1}^m c_{l,k}|\}_{k=1}^{n-1}$  (periodične) ili  $PSL = \max\{|\sum_{l=1}^m r_{l,k}|\}_{k=1}^{n-1}$  (aperiodične)
- $ISL = \sum_{k=1}^{n-1} |\sum_{l=1}^m c_{l,k}|^2$  (periodične) ili  $ISL = \sum_{k=1}^{n-1} |\sum_{l=1}^m r_{l,k}|^2$  (aperiodične)
- $MF = \frac{E^2}{2ISL}$  (periodične i aperiodične)

Pretpostavlja se da ne postoje savršeni periodični binarni nizovi veće dužine od pet. Nivoi periodičnih autokorelacija binarnih nizova kongruentni su sa dužinom po modulu 4 dok binarni nizovi za aperiodične autokorelacije imaju optimalni PSL jednak jedan. Poznati su Barkerovi nizovi dužine do trinaest (videti [2] str. 405):

Dužina niza	Niz
1	+1
2	+1,+1
3	+1,+1,-1
4	+1,+1,-1,+1
5	+1,+1,+1,-1,+1

7	+1,+1,+1,-1,-1,+1,-1
11	+1,+1,+1,-1,-1,-1,+1,-1,-1,+1,-1
13	+1,+1,+1,+1,+1,-1,-1,+1,+1,-1,+1,-1,+1

Tabela 1.1 Barkerovi nizovi

Golay je uveo i pojam faktora kvaliteta kao meru za određivanje dobrih osobina aperiodičnih autokorelacija. Slučajne sekvence imaju aperiodični faktor kvaliteta oko 1 dok se faktor većim od 3 smatra veoma dobrim. Barkerova sekvenca dužine 13 ima faktor kvaliteta 14.08. ([3] str. 7-8)

## 1.2 Metod naizmeničnih projekcija

U ovom delu biće opisan metod naizmeničnih projekcija i dokaz konvergencije (videti [4] str. 1-4).

Metod naizmeničnih projekcija se koristi da bi se pronašla tačka koja pripada preseku dva konveksna skupa. Koriste se projekcije jednog skupa u drugi koje se naizmenično primenjuju.

Neka su  $A$  i  $B$  zatvoreni konveksni skupovi u  $R^n$  i neka je  $f_A$  projekcija skupa  $B$  na skup  $A$  i  $f_B$  projekcija skupa  $A$  na skup  $B$ . Algoritam počinje sa izborom jednog elementa  $a_0$  iz skupa  $A$ . Zatim se pronalazi element  $b_1 = f_B(a_0)$ . U trećem koraku pronalazi se element  $a_1 = f_A(b_1)$ . Ponavljajući postupak projekcija iz  $A$  u  $B$  i nazad iz  $B$  u  $A$  dobijaju se nizovi  $a_1, a_2, \dots, a_n \in A$  i  $b_1, b_2, \dots, b_n \in B$ .

**Stav 1.2.1.** Ako je presek skupova  $A$  i  $B$  neprazan, nizovi  $(a_n)_{n \in \mathbb{N}}$  i  $(b_n)_{n \in \mathbb{N}}$  će konvergirati elementu  $c \in A \cap B$  kada  $n \rightarrow \infty$ .

Dokaz: Neka je  $c \in A \cap B$ . Za skup  $B$  i  $k \in \mathbb{N}$  važi:

$$B \subseteq \{x | (a_k - b_k)^T(x - b_k) \leq 0\} \quad (1.1)$$

Sa druge strane važi:

$$\|a_k - c\|^2 = \|a_k - b_k + b_k - c\|^2 = \|a_k - b_k\|^2 + \|b_k - c\|^2 + 2(a_k - b_k)^T(b_k - c)$$

pa iz (1.1) sledi:

$$\|a_k - c\|^2 \geq \|a_k - b_k\|^2 + \|b_k - c\|^2,$$

odnosno

$$\|b_k - c\|^2 \leq \|a_k - c\|^2 - \|a_k - b_k\|^2 \quad (1.2)$$

Slično se pokazuje da je

$$\|a_{k+1} - c\|^2 \leq \|b_k - c\|^2 - \|a_{k+1} - b_k\|^2 \quad (1.3)$$

što znači da je  $b_k$  bliži  $c$  nego što je  $a_k$  i da je  $a_{k+1}$  bliži  $c$  nego što je  $b_k$ . Još važi sledeće:

$$\|a_k - c\|^2 \leq \|a_0 - c\|^2 \text{ i } \|b_k - c\|^2 \leq \|a_0 - c\|^2, k \in \mathbb{N}$$

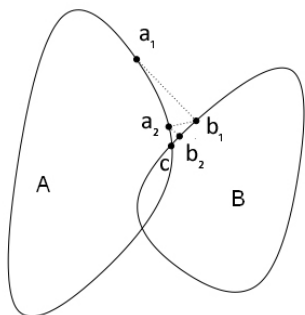
što znači da su nizovi  $(a_n)_{n \in \mathbb{N}}$  i  $(b_n)_{n \in \mathbb{N}}$  ograničeni pa imaju graničnu vrednost. Za  $(a_n)_{n \in \mathbb{N}}$  obeležimo graninu vrednost sa  $\bar{a}$ . Kako je niz

$$\|a_0 - c\|, \|b_0 - c\|, \|a_1 - c\|, \|b_1 - c\|, \dots \quad (1.4)$$

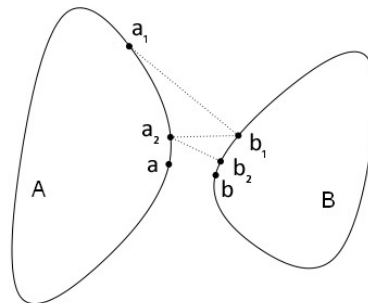
opadajući, on konvergira, pa iz (1.2) i (1.3) sledi da  $\|a_k - b_k\|$  i  $\|a_{k+1} - b_k\|$  teže nuli kada  $k \rightarrow \infty$ . Dalje važi da  $\text{dist}(a_k, B) = \text{dist}(a_k, b_k) \rightarrow 0, k \rightarrow \infty$  i postoji podniz od  $(a_n)_{n \in \mathbb{N}}$  koji konvergira ka  $\bar{a}$ . Kako je  $B$  zatvoren važi da je  $\bar{a} \in B$ , odnosno  $\bar{a} \in A \cap B$ . Kako je  $c$  proizvoljna tačka preseka  $A$  i  $B$  možemo uzeti da je  $c = \bar{a}$ . Kako podniz niza iz (1.4)  $\|a_n - c\| \rightarrow 0, n \rightarrow \infty$  tada i  $\|b_n - c\| \rightarrow 0, n \rightarrow \infty$  ■ [4]

Ako je presek skupova  $A$  i  $B$  prazan i ako sa  $\text{dist}(A, B)$  obeležimo udaljenost skupova (npr. najmanju euklidsku normu od svih parova  $(a, b), a \in A$  i  $b \in B$ ), tada će  $a_n \rightarrow a \in A$  i  $b_n \rightarrow b \in B$  kada  $n \rightarrow \infty$  tako da  $\|a - b\| = \text{dist}(A, B)$ . Dokaz konvergencije rešenja u ovom slučaju je sličan dokazu prethodnog stava.

Slike 1.1 i i.2 ilustruju prvih nekoliko iteracija metode naizmernih projekcija kada je presek konveksnih skupova  $A$  i  $B$  neprazan, odnosno prazan.



Slika 1.1 Naizmernične projekcije kada je presek skupova  $A$  i  $B$  neprazan



Slika 1.2 Naizmernične projekcije kada je presek skupova  $A$  i  $B$  prazan

Metod naizmernih projekcija se može proširiti na veći broj skupova i tada se naziva metod uzastopnih projekcija. Tada se projekcije skupova  $C_1, C_2, \dots, C_k$  primenjuju na sledeći način: element  $x_1^{(1)} \in C_1$  se projektuje u  $x_2^{(1)} \in C_2$ ,  $x_2^{(1)} \in C_2$  se projektuje u  $x_3^{(1)} \in C_3$ , ...,  $x_{k-1}^{(1)} \in C_{k-1}$  se projektuje u  $x_k^{(1)} \in C_k$  i  $x_k^{(1)} \in C_k$  se projektuje u  $x_1^{(2)} \in C_1$ , a zatim se ponavlja ciklus od  $k$  projekcija.

## 2 Algoritmi ITROX- $\mathcal{P}$ i ITROX- $\mathcal{AP}$

U radu [1] su predloženi sledeći algoritmi koji će se dalje razmatrati:

- ITROX- $\mathcal{P}$ , za pronalaženje skupa komplementarnih (realnih ili kompleksnih) nizova sa dobrim periodičnim autokorelacionim osobinama,
- ITROX- $\mathcal{AP}$ , za pronalaženje skupa komplementarnih (realnih ili kompleksnih) nizova sa dobrim aperiodičnim autokorelacionim osobinama.
- Za konačne ili diskretne azbuke predložene su izmene prethodnih algoritama.

Poglavlje 2.1 daje definicije pojma uvijenog proizvoda dva vektora  $x, y \in \mathbb{C}^{n \times 1}$  koji se označava sa  $x \cup y^H$  i funkcije  $\mathcal{F}$  kojom se matricni proizvod  $xy^H$  preslikava u uvijeni proizvod (videti [1] str. 2181). Poglavlja 2.2 i 2.3 opisuju algoritme ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  (videti [1] str. 2182-2186). Poglavlje 2.4 opisuje izmene algoritama kada je potrebno pronaći samo jedan niz, a poglavlje 2.5 kada se radi sa konačnim ili diskretnim azbukama (videti [1] str. 2186-2188).

### 2.1 Uvijeni proizvod vektora

**Definicija 2.1.1.** Neka su  $x, y \in \mathbb{C}^{n \times 1}$  ( $x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ ) i  $y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ . Uvijeni proizvod vektora  $x$  i  $y$  definiše se sledećom formulom:

$$x \cup y^H = \begin{bmatrix} x_1 y_1^* & x_2 y_2^* & \cdots & x_n y_n^* \\ x_1 y_2^* & x_2 y_3^* & \cdots & x_n y_1^* \\ \vdots & \vdots & \ddots & \vdots \\ x_1 y_n^* & x_2 y_1^* & \cdots & x_n y_{n-1}^* \end{bmatrix}$$

gde je  $y^H = (y_1^* \ \cdots \ y_n^*)$  hermitski transponat, a  $y_i^*$  je konjugat  $i$ -tog elementa vektora  $y$ .

Kako je  $xy^H = \begin{bmatrix} x_1 y_1^* & x_1 y_2^* & \cdots & x_1 y_n^* \\ x_2 y_1^* & x_2 y_2^* & \cdots & x_2 y_n^* \\ \vdots & \vdots & \ddots & \vdots \\ x_n y_1^* & x_n y_2^* & \cdots & x_n y_n^* \end{bmatrix}$  to se permutacijom elemenata uvijenog proizvoda  $x \cup y^H$  može dobiti proizvod  $xy^H$ .

Ako sa  $\text{vec}(A)$  označimo slaganje elemenata matrice  $A \in \mathbb{C}^{n \times n}$  na sledeći način:

$$\text{vec}(A) = \text{vec} \left( \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \right) = \begin{bmatrix} a_{11} \\ \vdots \\ a_{1n} \\ a_{21} \\ \vdots \\ a_{2n} \\ \vdots \\ a_{n1} \\ \vdots \\ a_{nn} \end{bmatrix}$$

tada postoji permutacijska matrica  $P \in \mathbb{C}^{n^2 \times n^2}$  takva da je  $xy^H = P \text{vec}(x \cup y^H)$ , odnosno  $x \cup y^H = P^T \text{vec}(xy^H)$ .

**Definicija 2.1.2.** Neka je  $Z \in \mathbb{C}^{n \times n}$ . Funkcija  $\mathcal{F}$  se definiše nad  $\mathbb{C}^{n \times n}$  sa  $\mathcal{F}(Z) = P^T \text{vec}(Z)$  gde je  $P \in \mathbb{C}^{n^2 \times n^2}$  permutacijska matrica definisana sa:

$$P = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots & \dots & \vdots & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots & \dots & \vdots & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & & & \vdots & \vdots & & & \vdots & & & & \vdots & & & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots & \dots & \vdots & \ddots & \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

ili

$P = (e_1 \ e_{n+1} \ \dots \ e_{n(n-1)+1} \ e_{n+2} \ e_{2n+2} \ \dots \ e_2 \ \dots \ e_{n^2} \ e_{n-1} \ \dots \ e_{n(n-1)-1})^T$  gde je  $e_k$  ( $k = [1..n]$ )  $k$ -ti standardni bazni vektor prostora  $\mathbb{R}^n$  (Senčenjem su odvojeni blokovi dimenzija  $n \times n$  delova matrice koja je dimenzije  $n^2 \times n^2$ ).

Primitimo da je  $P^{-1} = P^T$  pa se inverz funkcije  $\mathcal{F}$  može napisati kao  $\mathcal{F}^{-1}(Z) = P \text{vec}(Z)$ .

## 2.2 Algoritam ITROX- $\mathcal{P}$

Komplementarni skupovi binarnih nizova će uvek imati istu raspodelu energije za nizove koja je jednaka njihovoj dužini, a ukupna energija će biti jednaka dužini nizova pomnoženoj sa brojem nizova. Kada je potrebno pronaći skup realnih ili kompleksnih komplementarnih nizova sa dobrim periodičnim autokorelacionim osobinama, važno je unapred znati energetske komponente tih nizova. Najčešće se koristi jednaka raspodela energije, ali ponekad je potrebno nekim signalima dodeliti veću, a nekim manju energiju.

Zato se za ulazne podatke algoritma ITROX- $\mathcal{P}$  uzimaju

- dužina nizova ( $\mathbf{n}$ ),
- broj traženih komplementarnih nizova ( $\mathbf{k}$ ) i

- njihova energetska raspodela koja se sadrži u vektoru  $\mathbf{p} = \begin{bmatrix} \Gamma_1 \\ \vdots \\ \Gamma_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in (\mathbb{R}^+ \cup \{0\})^{n \times 1}$ .

Sa  $\mathbf{E}$  obeležimo ukupnu energiju traženog skupa komplementarnih nizova (sumu elemenata vektora  $\boldsymbol{\rho}$ ), tj.  $\mathbf{E} = \sum_{i=1}^k r_i$ .

Sa  $\mathbf{1}$  označimo vektor iz  $\mathbb{C}^{n \times 1}$  čiji su svi elementi jednaki jedinici, odnosno  $\mathbf{1} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ .

Primitimo da za uvejeni proizvod vektora  $x \in \mathbb{C}^{n \times 1}$  ( $x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ ) sa samim sobom važi da je

$$(x \cup x^H)\mathbf{1} = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix},$$

gde su sa  $c_k$  označene periodične autokorelacije  $k$ -tog reda niza  $x_1, x_2, \dots, x_n$ .

Ako posmatramo skup  $S = \{x_1, x_1, \dots, x_k\}$  periodično komplementarnih nizova dužine  $n$  tada važi:

$$\sum_{i=1}^k (x_i \cup x_i^H)\mathbf{1} = \begin{bmatrix} c_{1,0} + \dots + c_{k,0} \\ c_{1,1} + \dots + c_{k,1} \\ \vdots \\ c_{1,n-1} + \dots + c_{k,n-1} \end{bmatrix} = \mathbf{E}e_1$$

Neka je sada  $A \in \mathbb{C}^{n \times n}$  matrica za koju važi  $A\mathbf{1} = \mathbf{E}e_1$  i pretpostavimo da matrica  $\mathcal{F}(A)$  ima  $k$  ( $k \leq n$ ) pozitivnih sopstvenih vrednosti  $\lambda_1, \lambda_2, \dots, \lambda_k$ . Tada se  $\mathcal{F}(A)$  može zapisati kao dekompozicija po sopstvenim vrednostima:

$$\mathcal{F}(A) = UDU^H$$

gde je  $D$  dijagonalna matrica iz  $\mathbb{R}^{n \times n}$  čiji su dijagonalni elementi sopstvene vrednosti matrice  $\mathcal{F}(A)$ , a  $U \in \mathbb{C}^{n \times n}$  matrica čije kolone predstavljaju sopstvene vektore  $u_1, u_2, \dots, u_k$  matrice  $\mathcal{F}(A)$ .

Ako sa  $x_j$  označimo vektor  $\sqrt{\lambda_j}u_j$  (za  $1 \leq j \leq k$ ), tada važi:

$$\mathcal{F}(A) = \sum_{j=1}^k (x_j x_j^H) \text{ pa je } A = \sum_{j=1}^k (x_j \cup x_j^H).$$

što znači da je

$$A\mathbf{1} = \sum_{j=1}^k (x_j \cup x_j^H)\mathbf{1} = \mathbf{E}e_1.$$

Odnosno, vektori  $x_j$  (za  $1 \leq j \leq k$ ) predstavljaju  $k$  komplementarnih nizova dužine  $n$  čije su energetske vrednosti određene sopstvenim vrednostima matrice  $\mathcal{F}(A)$ .

Dakle, rešenje problema pronalaženja skupa nizova sa dobrim periodičnim autokorelacionim osobinama se može ostvariti pronalaženjem matrice koja pripada preseku sledeća dva skupa matrica:

**Definicija 2.2.1.** Skup  $\Gamma_{\mathcal{P}}(\mathbf{E})$  je skup svih matrica  $\mathcal{F}(A)$  takvih da važi  $A\mathbf{1} = Ee_1$  gde je  $A \in \mathbb{C}^{n \times n}$ .

**Definicija 2.2.2.** Skup  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  je skup svih hermitskih matrica čije su sopstvene vrednosti zadate vektorom  $\boldsymbol{\rho} \in (\mathbb{R}^+ \cup \{0\})^{n \times 1}$ .

Algoritam ITROX- $\mathcal{P}$  koristi metod naizmeničnih projekcija skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  i  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$ . Skup  $\Gamma_{\mathcal{P}}(\mathbf{E})$  je definisan ukupnom energijom  $\mathbf{E}$  i predstavlja sliku funkcije  $\mathcal{F}$  skupa matrica kojima je suma elemenata prvog reda jednaka  $\mathbf{E}$  dok su sume elemenata drugih redova jednake nuli. Skup  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  je određen energetskim komponentama nizova koje je potrebno generisati. Sa jedne strane elementi skupa  $\Gamma_{\mathcal{P}}(\mathbf{E})$  čuvaju osobinu komplementarnosti nizova, a sa druge elementi skupa  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  čuvaju njihovu energetsku raspodelu.

Projekcije skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  u  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  i  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  u  $\Gamma_{\mathcal{P}}(\mathbf{E})$  su određene korišćenjem sledećih teorema čiji dokazi se mogu pronaći u [1] (str. 2183).

**Teorema 2.2.1.** Neka je  $X \in \mathbb{C}^{n \times n}$  hermitska matrica čija je dekompozicija po sopstvenim vrednostima  $VD'V^H$ . Tada se ortogonalna projekcija  $X$  u  $\Lambda_{\mathcal{P}}(\boldsymbol{\rho})$  može dobiti zamenom  $D'$  sa

$$D = \begin{bmatrix} r_1 & \cdots & 0 & & \\ \vdots & \ddots & \vdots & & \\ 0 & \cdots & r_k & & \\ \mathbf{0}_{k \times (n-k)} & & & \mathbf{0}_{(n-k) \times (n-k)} & \end{bmatrix}, \text{ odnosno } Y = X^\perp = VDV^H.$$

**Teorema 2.2.2.** Neka je  $X = Y^\perp$  ortogonalna projekcija matrice  $Y \in \mathbb{C}^{n \times n}$  u  $\Gamma_{\mathcal{P}}(\mathbf{E})$ . Tada se  $\mathcal{F}^{-1}(X)$  može dobiti dodavanjem fiksnih vrednosti svakom redu matrice  $\mathcal{F}^{-1}(Y)$  tako da je  $\mathcal{F}^{-1}(Y)\mathbf{1} = Ee_1$ , tj.  $[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y)]_{k,l} + \frac{1}{n}(E\delta_{k-1} - \sum_{l'=1}^n [\mathcal{F}^{-1}(Y)]_{k,l'})$ , gde je  $\delta_k$  Kronekerova delta funkcija  $(\delta_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases})$ .

### Algoritam ITROX- $\mathcal{P}$ :

**Ulaz:**  $n$  (dužina nizova),  $k$  (broj traženih nizova;  $1 \leq k \leq n$ ),  $\boldsymbol{\rho}$  (niz dužine  $n$  nenegativnih realnih brojeva)

**Izlaz:**  $S$  (matricadimenzije  $k \times n$  u čijim redovima se nalaze komplementarni nizovi dužine  $n$ )

**begin**

$E := \boldsymbol{\rho}.\text{sum}()$  {računa sumu elemenata niza  $\boldsymbol{\rho}$ }

$D := \text{diagonalna\_matrica}(\boldsymbol{\rho})$  {formira se dijagonalna matrica  $D$  dimenzije  $n \times n$  sa elementima vektora  $\boldsymbol{\rho}$  postavljenim na glavnoj dijagonali}

$X_0 = \text{generisati\_slučajnu\_matricu}()$  {generiše slučajnu matricu za koju važi  $X_0\mathbf{1} = Ee_1$ }

$X := \mathcal{F}(X_0)$

**do**

$(V, D') := \text{dekompozicija}(X)$  {dekompozicija po sopstvenim vrednostima  $X = VD'V^H$ }

$Y := VDV^H$  {primenjuje se teorema 2.2.1.}



{primenjuje se teorema 2.2.2.}

**for**  $k := 1$  to  $n$  **do**

**for**  $l := 1$  to  $n$  **do**

$$[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y)]_{k,l} + \frac{1}{n} \left( E\delta_{k-1} - \sum_{l'=1}^n [\mathcal{F}^{-1}(Y)]_{k,l'} \right)$$

    razlika:  $= \|X - Y\|_F$       {računa se euklidska norma razlike matrica  $X$  i  $Y$ }

**until** razlika  $< \varepsilon$  {algoritam se prekida ako je razlika dovoljno mala za neko  $\varepsilon > 0$ }

$(V, D')$ : = dekompozicija( $X$ )

$S = (\sqrt{D}V^T)(0, \dots, k-1 | 0, \dots, n-1)$  {matrica  $S$  dimenzije  $k \times n$  sadrži  $k$  pronađenih nizova dužine  $n$ }

**end**

### 2.3 Algoritam ITROX- $\mathcal{AP}$

Ako posmatramo niz  $x$  dužine  $n$ , tada se aperiodična korelacija niza  $x$  može posmatrati kao periodična autokorelacija proširenog niza  $x' = \begin{pmatrix} x \\ 0_{(n-1) \times 1} \end{pmatrix}$ .

Ulazni podaci su slični prethodnom algoritmu pri čemu se  $\rho$  zamenjuje sa  $\rho' = \begin{pmatrix} \rho \\ 0_{(n-1) \times 1} \end{pmatrix} \in (\mathbb{R}^+ \cup \{0\})^{(2n-1) \times 1}$ .

- dužina nizova ( $n$ ),
- broj traženih komplementarnih nizova ( $k$ ) i

- njihova energetska raspodela koja se sadrži u vektoru  $\rho' = \begin{bmatrix} r_1 \\ \vdots \\ r_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in (\mathbb{R}^+ \cup \{0\})^{(2n-1) \times 1}$

i  $E = \sum_{i=1}^k r_i$ .

Slično se skupovi  $\Gamma_{\mathcal{P}}(E)$  i  $\Lambda_{\mathcal{P}}(\rho)$  zamenjuju sa  $\Gamma_{\mathcal{AP}}(E)$  i  $\Lambda_{\mathcal{AP}}(\rho')$  datim sledećim definicijama:

**Definicija 2.3.1.** Skup  $\Gamma_{\mathcal{AP}}(E)$  je skup svih matrica  $\mathcal{F}(X_0)$  takvih da važi  $X_0 \mathbf{1} = E e_1$  i  $\mathcal{F}(X_0) \odot M = \mathcal{F}(X_0)$  gde je  $X_0 \in \mathbb{C}^{(2n+1) \times (2n+1)}$  i  $M = \begin{bmatrix} \mathbf{1}_{n \times n} & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}$  (operacija  $\odot$  predstavlja množenje matrica član po član).

**Definicija 2.3.2.** Skup  $\Lambda_{\mathcal{AP}}(\rho')$  je skup svih hermitskih matrica čije su sopstvene vrednosti zadate vektorom  $\rho' = \begin{pmatrix} \rho \\ 0_{(n-1) \times 1} \end{pmatrix}$  gde je  $\rho \in (\mathbb{R}^+ \cup \{0\})^{n \times 1}$ .

Projekcije su određene korišćenjem sledećih teorema čiji dokazi se mogu pronaći u [1] (str.2184).

**Teorema 2.3.1.** Neka je  $X \in \mathbb{C}^{(2n-1) \times (2n-1)}$  hermitska matrica čija je dekompozicija po sopstvenim vrednostima

$$X = \begin{bmatrix} V_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix} \begin{bmatrix} D'_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & 0_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} V_{n \times n}^H & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix}. \text{ Tada se ortogonalna}$$

projekcija  $X$  na  $\Lambda_{\mathcal{AP}}(\rho')$  može dobiti zamenom  $D'$  sa  $D = \begin{bmatrix} r_1 & \cdots & 0 & & \\ \vdots & \ddots & \vdots & & \mathbf{0}_{(2n-1-k) \times k} \\ 0 & \cdots & r_k & & \\ \mathbf{0}_{k \times (2n-1-k)} & & & \mathbf{0}_{(2n-1-k) \times (2n-1-k)} & \end{bmatrix}$ ,

odnosno  $Y = X^\perp = \begin{bmatrix} V_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix} \begin{bmatrix} D_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & 0_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} V_{n \times n}^H & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix}$ .

**Teorema 2.3.2.** Neka je  $X = Y^\perp$  ortogonalna projekcija matrice  $Y \in \mathbb{C}^{(2n-1) \times (2n-1)}$  na  $\Gamma_{\mathcal{AP}}(\mathbf{E})$ .

Obeležimo sa  $\omega_k$  broj jedinica u  $k$ -tom redu matrice  $\mathcal{F}^{-1}(M)$ . Tada se  $\mathcal{F}^{-1}(X)$  može dobiti dodavanjem fiksnih vrednosti svakom redu matrice  $\mathcal{F}^{-1}(Y)$  tako da je  $\mathcal{F}^{-1}(Y)\mathbf{1} = \mathbf{E}e_1$ , t.j.:

$$[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y)]_{k,l} + \frac{1}{\omega_k} (E\delta_{k-1} - \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M \odot Y)]_{k,l'}), \text{ gde je}$$

$$M = \begin{bmatrix} \mathbf{1}_{n \times n} & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \text{ i } \omega_k = \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M)]_{k,l'} = \begin{cases} n - k + 1, & k \leq n \\ k - n, & k > n \end{cases}$$

**Algoritam ITROX- $\mathcal{AP}$ :**

**Ulaz:**  $n$  (dužina nizova),  $k$  (broj traženih nizova;  $1 \leq k \leq n$ ),  $\rho'$  (niz dužine  $2n - 1$  nenegativnih realnih brojeva)

**Izlaz:**  $S$  (matrica dimenzije  $k \times n$  u čijim redovima se nalaze komplementarni nizovi dužine  $n$ )

**begin**

$E := \rho'.\text{sum}()$  {računa sumu elemenata niza  $\rho'$ }

$D := \text{diagonalna\_matrica}(\rho')$  {formira se dijagonalna matrica  $D$  dimenzije  $(2n - 1) \times (2n - 1)$  sa elementima vektora  $\rho'$  postavljenim na glavnoj dijagonali}

$X_0 = \text{generisati\_slučajnu\_matricu}()$  {generiše slučajnu matricu  $X_0$  dimenzije  $(2n - 1) \times (2n - 1)$  za koju važi  $X_0\mathbf{1} = Ee_1$ }

$X := \mathcal{F}(X_0)$

**do**

$(V, D') := \text{dekompozicija}(X)$  {dekompozicija po sopstvenim vrednostima matrice  $X$ }

$$Y := \begin{bmatrix} V_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix} \begin{bmatrix} D_{n \times n} & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & 0_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} V_{n \times n}^H & 0_{n \times (n-1)} \\ 0_{(n-1) \times n} & I_{n-1} \end{bmatrix} \text{ {primenjuje se teorema 2.3.1.}}$$

{primenjuje se teorema 2.2.2.}

**for**  $k := 1$  to  $2n - 1$  **do**

**for**  $l := 1$  to  $2n - 1$  **do**

$$[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y)]_{k,l} + \frac{1}{\omega_k} \left( E\delta_{k-1} - \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M \odot Y)]_{k,l'} \right)$$

razlika:  $= \|X - Y\|_F$  {računa se euklidska norma razlike matrica  $X$  i  $Y$ }

**until** razlika  $< \varepsilon$  {algoritam se prekida ako je razlika dovoljno mala za neko  $\varepsilon > 0$ }

$(V, D') := \text{dekompozicija}(X)$

$S := (\sqrt{D}V^T)(0, \dots, k-1 | 0, \dots, n-1)$  {matrica S dimenzije  $k \times n$  sadrži k pronađenih nizova dužine  $n$ }

end

## 2.4 Pronalaženje samo jednog niza

Ukoliko je potrebno pronaći samo jedan niz sa dobrim autokorelacionim osobinama tada je potrebno pronaći jedan sopstveni vektor i to onaj koji odgovara najvećoj sopstvenoj vrednosti matrice. U ovom slučaju se algoritmi ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  mogu izmeniti tako da se umesto izračunavanja dekompozicija po sopstvenim vrednostima matrice X koristi metod stepenovanja.

U ovom delu biće ukratko opisan metod stepenovanja (videti [13]).

Metod stepenovanja se koristi za pronalaženje sopstvene vrednosti koja odgovara najvećoj (po apsolutnoj vrednosti) sopstvenoj vrednosti matrice. Ako je  $X \in \mathbb{C}^{n \times n}$  algoritam počinje sa nekim vektorom  $x_0 \in \mathbb{C}^{n \times 1}$  (npr.  $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$ ). Zatim u svakoj k-toj iteraciji ( $k > 0$ ) računa normiran proizvod matrice X i vektora  $x_{k-1}$ , odnosno:

$$x_k = \frac{Xx_{k-1}}{\|Xx_{k-1}\|}, k > 0$$

Neka je  $X = V\Lambda V^{-1}$  dekompozicija matrice X gde je V matrica u čijim kolonama se nalaze sopstveni vektori  $v_i, 1 \leq i \leq n$ , a  $\Lambda$  dijagonalna matrica sa sopstvenim vrednostima  $\lambda_i, 1 \leq i \leq n$  na dijagonali takvih da je  $\lambda_1$  najveća po apsolutnoj vrednosti. Ako je  $x_0 = \sum_{i=1}^n c_i v_i$  i  $c_1 \neq 0$ , tada za  $k > 0$  važi

$$\begin{aligned} x_k &= \frac{Xx_{k-1}}{\|Xx_{k-1}\|} = \frac{X^k x_0}{\|X^k x_0\|} = \frac{(V\Lambda V^{-1})^k x_0}{\|(V\Lambda V^{-1})^k x_0\|} = \frac{V\Lambda^k V^{-1} x_0}{\|V\Lambda^k V^{-1} x_0\|} = \frac{V\Lambda^k V^{-1} (\sum_{i=1}^n c_i v_i)}{\|V\Lambda^k V^{-1} (\sum_{i=1}^n c_i v_i)\|} \\ &= \frac{V\Lambda^k (\sum_{i=1}^n c_i e_i)}{\|V\Lambda^k (\sum_{i=1}^n c_i e_i)\|} = \left(\frac{\lambda_1}{|\lambda_1|}\right)^k \frac{c_1}{|c_1|} \frac{v_1 + \frac{1}{c_1} V(\frac{1}{\lambda_1} \Lambda)^k (\sum_{i=2}^n c_i e_i)}{\|v_1 + \frac{1}{c_1} V(\frac{1}{\lambda_1} \Lambda)^k (\sum_{i=2}^n c_i e_i)\|} \end{aligned}$$

Kako je  $\lim_{k \rightarrow \infty} (\frac{1}{\lambda_1} \Lambda)^k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix}$  to je  $\lim_{k \rightarrow \infty} (\frac{1}{c_1} V(\frac{1}{\lambda_1} \Lambda)^k (\sum_{i=2}^n c_i e_i)) = 0$

što znači da je  $\lim_{k \rightarrow \infty} x_k = v_1$ .

### Algoritam Metod Stepenovanja:

**Ulaz:** n (pozitivan ceo broj), X(matrica iz  $\mathbb{C}^{n \times n}$ ), l (maksimalni broj iteracija)

**Izlaz:** y (sopstveni vektor matrice X koji odgovara najvećoj (po apsolutnoj vrednosti) sopstvenoj vrednosti matrice X)

begin

```

y :=  $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$       {početna vrednost vektora  $y \in \mathbb{C}^{n \times 1}$ }
d := 1
i := 0 {korak iteracije}
while d > 1E - 3 and i < l
    p := y      {prethodno izračunat sopstveni vektor}
    y := Xy
    y := y/||y||
    d := ||p - y||      {norma razlike vektora y i p}
    i := i + 1
end
    
```

## 2.5 ITROX algoritam za nizove nad konačnim ili diskretnim azbukama

Predloženi algoritmi ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  daju dobre rezultate za realne i kompleksne nizove, ali kada je potrebno pronaći nizove nad nekom konačnom ili diskretnom azbukom potrebno je metod naizmeničnih projekcija primeniti nad nizom skupova koji konvergiraju jedan drugom pod uticajem neke „član po član“ monotone funkcije identiteta.

**Definicija 2.5.1.** Neka je  $f(t, s): \mathbb{C} \times (\mathbb{N} \cup \{0\}) \rightarrow \mathbb{C}$ . (Funkcija  $f$  se može proširiti na skup matrica na sledeći način:  $[f(X, s)]_{i,j} = f([X]_{i,j}, s)$ )

- Za funkciju  $f$  se kaže da je monotona „član po član“ ako su za svako  $t \in \mathbb{C}$  funkcije  $|f(t, s)|$  i  $\arg(f(t, s))$  monotone. ( $|\cdot|$  predstavlja moduo, a  $\arg(\cdot)$  argument kompleksnog broja).
- Skup  $T$  konvergira ka skupu  $T'$  pod funkcijom  $f$  ako
 
$$(\forall t \in T) f(t, 0) = t \text{ i } \lim_{s \rightarrow \infty} f(t, s) \in T'$$

$$(\forall t' \in T') (\exists t \in T) \lim_{s \rightarrow \infty} f(t, s) = t'$$
- Funkcija  $f$  se naziva identitetom ako
 
$$(\forall t \in T) (\forall t' \in T') \lim_{s \rightarrow \infty} f(t, s) \in T' \implies t' \text{ je najbliži element skupa } T' \text{ elementu } t$$
- Za niz skupova  $T^{(s)} = \{f(t, s) | t \in T, s \in \mathbb{N} \cup \{0\}\}$  se kaže da je niz konvergirajućih skupova.

**Primer 2.4.1** Primeri funkcija identiteta za neke najčešće korišćene konačne ili diskretne azbuke za neko  $\nu \in \mathbb{R}^+$ :

- $T = \mathbb{R} \setminus \{0\}, T' = \{-1, 1\}, f(t, s) = \text{sgn}(t) |t| e^{-\nu s}$
- $T = \mathbb{R}, T' = \mathbb{Z}, f(t, s) = [t] + \{t\} e^{-\nu s}$
- $T = \mathbb{C} \setminus \{0\}, T' = \{\zeta \in \mathbb{C} | |\zeta| = 1\}, f(t, s) = |t| e^{-\nu s} e^{j \arg(t)}$
- $T = \mathbb{C} \setminus \{0\}, T' = \{\zeta \in \mathbb{C} | \zeta^m = 1\}, f(t, s) = |t| e^{-\nu s} e^{j \frac{2\pi}{m} (\lfloor \frac{m \arg(t)}{2\pi} \rfloor + \{\frac{m \arg(t)}{2\pi}\})} e^{-\nu s}$

Za neku azbuku  $\Omega$  sa  $\Omega_X$  obeležimo sledeći skup ( $X|X = \sum_k x_k x_k^H, x_k(l) \in \Omega$ ). Ako postoji funkcija  $f$  pod kojom skup  $\Lambda_{\mathcal{P}}(\rho)$  konvergira ka skupu  $\Omega_X$ , tada naizmenične projekcije skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  i  $\Lambda_{\mathcal{P}}(\rho)$  algoritama ITROX- $\mathcal{P}$  možemo zameniti sa uzastopnim projekcijama skupova  $\Gamma_{\mathcal{P}}(\mathbf{E}), \Lambda_{\mathcal{P}}(\rho)$  i  $\Omega_X$  na sledeći način:

**Algoritam ITROX- $\mathcal{P}$  za nizove nad konačnom ili diskretnom azbukom  $\Omega$ :**

**Ulaz:**  $n$  (dužina nizova),  $k$  (broj traženih nizova;  $1 \leq k \leq n$ ),  $\rho$  (niz dužine  $n$  nenegativnih realnih brojeva)

**Izlaz:**  $S$  (matricadimenzije  $k \times n$  u čijim redovima se nalaze komplementarni nizovi dužine  $n$ )

**begin**

$E := \rho$ . sum() {računa sumu elemenata niza  $\rho$ }

$D := \text{diagonalna\_matrica}(\rho)$  {formira se dijagonalna matrica  $D$  dimenzije  $n \times n$  sa elementima vektora  $\rho$  postavljenim na glavnoj dijagonali}

$X_0 = \text{generisati\_slučajnu\_matricu}()$  {generiše slučajnu matricu za koju važi  $X_0 \mathbf{1} = Ee_1$ }

$X := \mathcal{F}(X_0)$

$s := 0$  {broj iteracija}

**do**

$(V, D') := \text{dekompozicija}(X)$  {dekompozicija po sopstvenim vrednostima  $X = VD'V^H$ }

$Y := VDV^H$  {primenjuje se teorema 2.2.1.}

$Y' := f(Y, s)$  {funkcija  $f$  je funkcija pod kojom skup  $\Lambda_{\mathcal{P}}(\rho)$  konvergira ka skupu  $\Omega_X$ }

$Y_1 := \lim_{s \rightarrow \infty} f(Y, s)$

{primenjuje se teorema 2.2.2.}

**for**  $k := 1$  to  $n$  **do**

**for**  $l := 1$  to  $n$  **do**

$$[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y')]_{k,l} + \frac{1}{n} \left( E\delta_{k-1} - \sum_{l'=1}^n [\mathcal{F}^{-1}(Y')]_{k,l'} \right)$$

$\text{razlika} := \|X - Y'\|$  {računa se euklidska norma razlike matrica  $X$  i  $Y'$ }

$\text{razlika\_Y1\_Y}' := \|Y_1 - Y'\|$  {računa se euklidska norma razlike matrica  $Y_1$  i  $Y'$ }

$s := s + 1$

**until**  $\text{razlika} < \varepsilon_1$  or  $\text{razlika\_Y1\_Y}' < \varepsilon_2$  {uslov za prekid algoritma za neko  $\varepsilon_1 > 0$  i  $\varepsilon_2 > 0$ }

$(V, D') := \text{dekompozicija}(X)$

$S := (\sqrt{D}V^T)(0, \dots, k-1 | 0, \dots, n-1)$  {matrica  $S$  dimenzije  $k \times n$  sadrži  $k$  pronađenih nizova dužine  $n$ }

**end**

Slično se menja algoritam ITROX- $\mathcal{AP}$  za nizove nad azbukom  $\Omega$ .

## 3 Programski jezik Python

Programerski jezik Python razvio je 1990. godine holandski programer Guido van Rossum [11] sa ciljem da stvori moćan jezik otvorenog koda i jednostavne sintakse koji omogućava veoma brz razvoj programa.

Python se može koristiti za pisanje skripti, funkcionalno, proceduralno ili objektno orijentisano programiranje. Najčešće se primenjuje za pisanje skripti za web alate, pisanje programa koji se koriste u naučne svrhe (bioinformatika i fizika), mrežno programiranje, razvoj 3D alata, itd. [12]

U ovom poglavlju biće dat kratak opis programskog jezika python, a detaljnije informacije se mogu pronaći u [10].

### 3.1 Sintaksa

Za ograničavanje programskih blokova koristi se blanko karakter. Identifikatori se mogu sastojati iz slova, brojeva ili podvlake pri čemu prvo slovo ne sme biti broj.

### 3.2 Strukture podataka

Tip promenljive se dinamički određuje na osnovu njene vrednosti pa tako ista promenljiva može u različitim delovima koda nositi vrednosti različitog tipa.

Osnovni tipovi podataka su logički, celobrojni (integer ili long integer), brojevi sa pokretnim zarezom, kompleksni i stringovi.

#### Liste

Liste su nizovi proizvoljnih objekata koji mogu biti različitog tipa. Elementi lista mogu biti i druge liste. Liste se kreiraju navođenjem elemenata liste razdvojenih zarezom unutar uglastih zagrada kao u sledećem primeru:

```
l = [1,0.5,'python',(5+3j)]
```

Elementima liste se pristupa preko indeksa.

Metod append se koristi za dodavanje elementa na kraj liste dok metod insert dodaje element na tačno određeno mesto u listi. Liste je moguće nadovezivati operatorom „+“ i multiplicirati korišćenjem operatora „\*“. Operacijom isecanja („:“) se mogu izdvojiti delovi liste.

#### N-torke (tuples)

Ovo je slična struktura sa listama sa tom razlikom da kada se jednom kreira nije moguće promeniti elemente niti je moguće dodavanje novih elemenata. N-torke se kreiraju navođenjem elemenata razdvojenih zarezom unutar zagrada kao u sledećem primeru:

```
t = (1,0.5,'python',(5+3j))
```

N-torke podržavaju operacije indeksiranja, isecanja i nadovezivanja.

## Skupovi

Skupovi se koriste da bi kreirali kolekciju koja nije indeksirana kao što su to liste i n-torke. Kreira se korišćenjem funkcije `set()` kojoj se prosleđuje indeksirana struktura podataka kao u sledećem primeru:

```
s = set([1,2,3,4,5])    # kreira skup brojeva 1,2, 3,4,5  
s = set('AEIOU')      # kreira skup karaktera 'A','E','I','O','U'
```

Elementi skupa ne mogu biti duplicirani.

Nad skupovima je moguće primeniti standardne operacije kao što su: unija(`|`), presek(`&`), razlika(`-`) i simetrična razlika(`^`). Moguće je dodati novi element korišćenjem metode `add`, izmeniti postojeći element korišćenjem metode `update` ili ga obrisati korišćenjem metode `remove`.

## Rečnici

Rečnici predstavljaju heš tabele čiji se objekti indeksiraju putem ključa. Kreiraju se navođenjem niza parova ključeva i objekata (koji se razdvajaju dvotačkom „:“) ovičenih vitičastom zagradom. Parovi se razdvajaju zarezom.

Primer kreiranja rečnika:

```
r = {"student": "jasna", "predmet": "master"}
```

Ključevi mogu biti tipa string, brojevi ili n-torke. Moguće je dodati nove objekte i menjati ili obrisati postojeće. Metodom `has_key` se proverava da li postoji zadati ključ u rečniku, a metoda `keys` prikazuje listu ključeva.

## 3.3 Kontrola toka

Slično većini programskih jezika Python poseduje uslovne naredbe (`if`, `if-else` i `if-elif-else`) kao i naredbe `while` i `for` za formiranje petlji.

### Uslovne naredbe

Osnovni oblik uslovne naredbe izgleda ovako:

```
if uslov:  
    naredbe  
elif uslov:  
    naredbe
```

```
elif uslov:  
    naredbe  
    ...  
else:  
    naredbe
```

Moguće je izostaviti elif i else delove.

### **While i for naredba**

Osnovni oblik while naredbe izgleda ovako:

```
while uslov:  
    naredbe
```

Osnovni oblik for naredbe izgleda ovako:

```
for a in s:  
    naredbe
```

For petlja za iteraciju koristi objekte kao što su liste, n-torke ili stringovi, ali i bilo koji drugi tip koji podržava iteraciju.

Za izlazak iz petlje koristi se naredba „break“, a naredba „continue“ za skok na novu iteraciju.

## **3.4 Funkcije**

Funkcije se definišu pomoću naredbe „def“.

```
def naziv_funkcije(argumenti):  
    naredbe  
    return rezultat
```

Argumentima je moguće je dodeliti podrazumevanu vrednosti tada oni postaju opcioni.

## **3.5 Klase**

Python podržava objektno orijentisano programiranje i samim tim i slične principe kreiranja i nasleđivanja klasa kao u drugim objektno orijentisanim jezicima.

Klase se definišu pomoću naredbe „class“.

```
class naziv_klase:  
    naredbe
```



### 3.6 Rad sa bibliotekama

Veliki programi se obično organizuju u više modula i paketa. Bilo koji izvorni python file može biti korišćen kao modul. Da bi se u neki kod uvezao modul koristi se naredba import na sledeći način:

Import naziv\_modula

Moguće je uvezenom modulu dodeliti neko drugo ime:

import numpy as np

Moduli se prilikom uvoza traže u direktorijumima koji su navedeni u listi sys.path. Da bi se dodao novi direktorijum u listu koristi se naredba „append“.

### 3.7 Ulaz i izlaz

Python omogućava rad sa standardnim ulazom/izlazom kao i sa filovima. Standardni ulaz i izlaz su definisani u modulu sys kao sys.stdin i sys.stdout. Metodi „read“ i „raw\_input“ se koriste za čitanje stdin dok se „print“ koristi za upisivanje u stdout.

Objekat fajla se kreira korišćenjem naredbi „open“ ili „file“:

open(naziv\_fajla [,mode [,bufsize]])

file(naziv\_fajla [,mode [,bufsize]])

Primer:

```
f = open('foo','r')
```

```
f = file('primer','w')
```

Metodiza čitanje fajla su „read“, „readline“, „readlines“, „xreadlines“, a metodi za upisivanje u fajl su „write“ i „writelines“. Pored ovih metoda postoje i metod za promenu pozicije fajla („seek“), brisanje izlaznog bafera („flush“) i drugi.

Objekat fajla se zatvara korišćenjem metode „close“.

## 4 Implementacija algoritama ITROX-P i ITROX-AP

Program ITROX za implementaciju algoritama ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  je napisan korišćenjem programskog jezika Python. Jedan od razloga za izbor Python programskog jezika je postojanje moćnih modula numpy i scipy koji sadrže veoma efikasne funkcije za rad sa matricama.

Za pisanje i testiranje koda programa korišćen je Eclipse Juno SR1 sa PyDev for Eclipse (v 2.8.2) za Python verziju 2.7.3.

Program se sastoji iz dvafajla ITROX.py i ITROXClass.py.

ITROX.py sadrži sledeće klase:

- UI\_MainWindow koja sadrži objekte grafičkog okruženja i objekat itroxObj klase itroxClass.
- MyListModel koja definiše model prikaza i editovanja objekata klase ListView.
- MyTableModel koja definiše model prikaza i editovanja objekata klase TableView.

ITROXClass.py sadrži klasu itroxClass u kojoj su definisane:

- promenljive ulaza i izlaza,
- procedura za generisanje početnog matrice,
- procedure za inicijalizaciju ulaznih podataka,
- procedura za generisanje nizova sa dobrim korelacionim osobinama - (periodična ili aperiodična) autokorelacija nad (realnim ili binarnim skupom),
- proceduru za prikaz rezultata (skup nizova i zbir uvijenog proizvoda elemenata tog skupa).

### 4.1 Grafičko okruženje

Za kreiranje grafičkog okruženja korišćeni su open source alati Eric4 (v 4.5.8) i Qt Designer (v 4.8.3). Eric4 predstavlja integrisano razvojno okruženje za programski jezik Python dok Qt Designer služi za generisanje grafičkog korisničkog okruženja. Eric4 je korišćen samo zbog lakšeg prevođenja Qt UI fajla „mainWindow.ui“ (kreiranom u Qt Designer-u) u python kod.

Klasa UI\_MainWindow predstavlja prevod „mainWindow.ui“ fajla u python kod kojoj su dodate funkcije i procedure za povezivanje događaja Qt objekata i funkcija i procedura klase itroxClass preko glavnog objekta itroxObj. Ova klasa sadrži i procedure za učitavanje i snimanje ulaznih podataka i proceduru za snimanje rezultata.

Program ITROX sadrži jedan prozor (definisano klasom UI\_MainWindow) u kome se sa leve strane nalazi deo u kome se definišu ulazni podaci programa i to:

- Dužina niza ( $n$ )
- Broj traženih nizova ( $k$ )
- Izbor azbuke (skup realnih brojeva ili skup binarnih brojeva)

- Izbor algoritma (ITROX-P za periodičnu autokorelaciju ili ITROX-AP za aperiodičnu autokorelaciju)
- Niz energetske komponente traženih nizova (vektor  $r$ )
- Početni niz ( $niz_0$ ) – moguće je uneti vrednosti ili pokrenuti proceduru za generisanje početnog niza)

Ulazne podatke je moguće učitati iz postojećeg fajla.

Ispod dela sa ulaznim podacima prikazuje se tablica koja sadrži elemente početne matrice  $X_0$  koja se izračunava na osnovu niza  $niz_0$ .

U desnom delu se prikazuju rezultati u obliku tabele koja sadrži skup od  $k$  generisanih nizova zadate dužine  $n$  i zbir uvijenog proizvoda elemenata tog skupa. Rezultati se mogu grafički prikazati pokretanjem procedure za prikaz rezultata. Moguće je snimiti rezultat u tekstualnom formatu ili snimiti grafički prikaz u formatu slike.

Za svaku iteraciju algoritma u delu „log“ se prikazuju informacije o svakoj iteraciji algoritma (poslednje rastojanje para pronađenih matrica i ukoliko je izabrana binarna azbuka rastojanju poslednje matrice koja je rezultat preslikavanja funkcije identiteta za binarnu azbuku u tekućoj iteraciji i graničnoj vrednosti tog preslikavanja kada broj iteracija teži beskonačnosti). Nakon poslednje iteracije u „log“ delu se ispisuje ukupan broj iteracija, ukupno vreme trajanja algoritma (u sekundama) i rezultujući PSL (maksimalna necentralna korelacija), ISL (integrisana necentralna korelacija) i MF (faktor kvaliteta).

## 4.2 Klasa ITROXClass

ITROXClass.py sadrži sledeće atribute:

- $n$  - dužina niza
- $k$  - broj traženih nizova
- $r$  – niz dužine  $k$  sa svim pozitivnim vrednostima
- $niz_0$  - početni niz
- $X_0$  - početna matrica
- $S$  - matrica čiji su redovi rezultujući nizovi (dimenzije  $k \times n$ )
- $S_*$  - matrica čiji su redovi rezultujući reani nizovi za slučaj da se generise nad binarnim skupom (dimenzije  $k \times n$ )
- $matricaUvijeniProizvod$  - zbir uvijenog proizvoda dobijenih nizova

### 4.2.1 Implementacija funkcija $\mathcal{F}$ i $\mathcal{F}'$

#### Funkcija $\mathcal{F}$

Za implementaciju funkcija  $\mathcal{F}$  iz definicije 2.1.2 u funkciju „F\_permutacija“ koristi se sledeći stav:

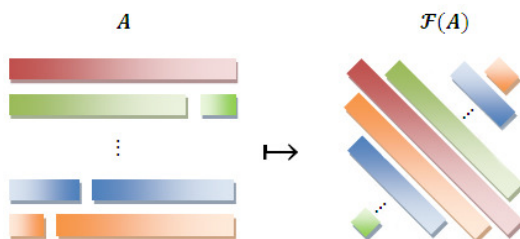
**Stav 4.2.1.** Neka je  $A = (a_{i,j})_{i,j=1}^n \in \mathbb{C}^{n \times n}$ . Za funkciju  $\mathcal{F}$  iz definicije 2.1.2 važi:

$$[\mathcal{F}(A)]_{i,j} = \begin{cases} a_{j-i+1,i}, & j \geq i \\ a_{(n+1)-(i-j),i}, & j < i \end{cases} \text{ za } i, j = \overline{1..n} \quad (4.1)$$

Dokaz: Primetimo da funkcija  $\mathcal{F}$  redove jedne matrice preslikava u dijagonale druge matrice. Naime, ako za matrice  $A$  i  $\mathcal{F}(A)$  iz prethodnog stava obeležimo sa:

- $r_k$  (za  $1 \leq k \leq n$ )  $k$ -ti red matrice  $A$ ,
- $d_0$  glavnu dijagonalu matrice  $\mathcal{F}(A)$ ,
- $d_k$  (za  $1 \leq k \leq n - 1$ )  $k$ -tu naddijagonalu matrice  $\mathcal{F}(A)$  i
- $d_{-k}$  (za  $1 \leq k \leq n - 1$ )  $k$ -tu poddijagonalu matrice  $\mathcal{F}(A)$ ,

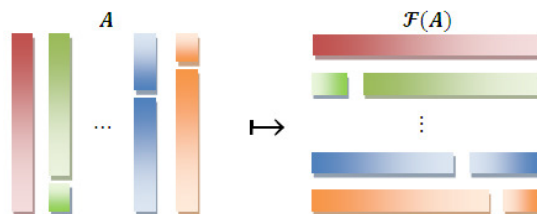
Tada će se prvih  $n - k$  elemenata reda  $r_k$  redom preslikati u dijagonalu  $d_k$ , a ostalih  $k$  elemenata redom u dijagonalu  $d_{-k}$  što je ilustrovano na slici 4.1.



Slika 4.1 Šema preslikavanja  $\mathcal{F}$

Elementi na glavnoj dijagonali matrice  $\mathcal{F}(A)$  imaju vrednost  $[\mathcal{F}(A)]_{i,i} = a_{1,i}$  za  $1 \leq i \leq n$ . Kada je  $1 \leq i < j \leq n$  tada  $[\mathcal{F}(A)]_{i,j}$  pripada naddijagonali  $d_{(j-i)}$ , a to znači da je  $[\mathcal{F}(A)]_{i,j} = a_{j-i+1,i}$ . Ako je  $1 \leq j < i \leq n$  onda  $[\mathcal{F}(A)]_{i,j}$  pripada poddijagonali  $d_{-(i-j)}$  pa prema tome  $[\mathcal{F}(A)]_{i,j} = a_{(n+1)-(i-j),i}$  ■

Sa druge strane, iz formule (4.1) se vidi da se  $k$ -ta kolona matrice  $A$  kada se šiftuje za  $k - 1$  mesta unapred preslikava u  $k$ -ti red matrice  $\mathcal{F}(A)$  što je ilustrovano na slici 4.2.



Slika 4.2 Šema preslikavanja  $\mathcal{F}$  sa šiftovanjem kolona

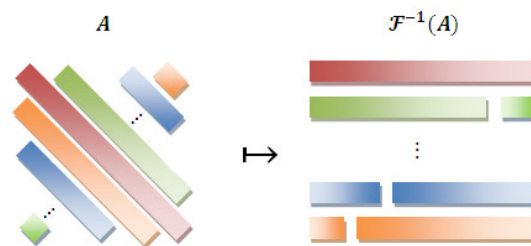
## Funkcija $\mathcal{F}^{-1}$

Slično je implementirana funkcija  $\mathcal{F}^{-1}$  „F\_permutacija\_inv“ za koju se koristi sledeći stav:

**Stav 4.2.2.** Neka je  $A = (a_{i,j})_{i,j=1}^n$ . Za funkciju  $\mathcal{F}^{-1}$  važi:

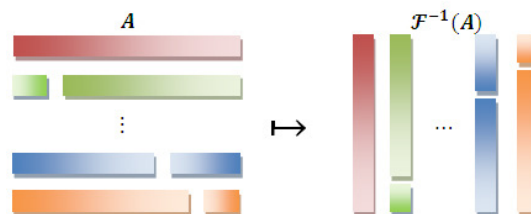
$$\mathcal{F}^{-1}(A)_{i,j} = \begin{cases} a_{j,i+j-1}, & i+j \leq n+1 \\ a, & i+j > n+1 \end{cases} \text{ za } i,j = \overline{1..n} \quad (4.2)$$

Primitimo da funkcija  $\mathcal{F}^{-1}$  dijagonale jedne matrice preslikava u redove druge matrice što je ilustrovano na slici 4.3.



Slika 4.3 Šema preslikavanja  $\mathcal{F}^{-1}$

Iz formule (4.2) se vidi da se  $k$ -ti red matrice  $A$  kada se šiftuje za  $k-1$  mesta unazad preslikava u  $k$ -tu kolonu matrice  $\mathcal{F}^{-1}(A)$  što je ilustrovano na slici 4.4.



Slika 4.4 Šema preslikavanja  $\mathcal{F}^{-1}$  sa šiftovanjem redova

Funkcije „F\_permutacija“ i „F\_permutacija\_inv“ koriste funkciju „roll“ iz numpy modula kojom se vrste i kolone matrica mogu šiftovati za određeni broj mesta.

## 4.2.2 Implementacija naizmeničnih projekcija

Pre opisa procedura koje implementiraju naizmenične projekcije skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  i  $\Lambda_{\mathcal{P}}(\mathbf{r})$  definisanih u drugom poglavlju primetimo da projekcija definisana teoremom 2.2.2 preslikava Hermitsku matricu u Hermitsku matricu koja pripada skupu  $\Gamma_{\mathcal{P}}(\mathbf{E})$  što je iskazano sledećim stavom.

**Stav 4.2.3.** Neka je  $A = (a_{i,j})_{i,j=1}^n \in \mathbb{C}^{n \times n}$  Hermitska matrica. Neka je  $B = (b_{i,j})_{i,j=1}^n$  matrica iz  $\mathbb{C}^{n \times n}$  takva da za neko  $E \in \mathbb{R}$  važi:

$$[\mathcal{F}^{-1}(B)]_{k,l} = [\mathcal{F}^{-1}(A)]_{k,l} + \frac{1}{n}(E\delta_{k-1} - \sum_{l'=1}^n [\mathcal{F}^{-1}(A)]_{k,l'}), \quad (4.3)$$

gde je  $\delta_k$  Kronekerova delta funkcija ( $\delta_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases}$ ). Tada je  $B$  Hermitska matrica.

Dokaz: Neka je  $A' = \mathcal{F}^{-1}(A) = (a'_{i,j})_{i,j=1}^n$ , odnosno  $A = \mathcal{F}(A') = (a_{i,j})_{i,j=1}^n$ . Tada iz (4.2) sledi

$$a'_{i,j} = \begin{cases} a_{j,i+j-1}, & i+j \leq n+1 \\ a_{j,i+j-(n+1)}, & i+j > n+1 \end{cases} \text{ za } i, j = \overline{1..n}. \quad (4.4)$$

Neka je  $T' = (t'_{i,j})_{i,j=1}^n \in \mathbb{C}^{n \times n}$  matrica definisana sa:

$$t'_{i,j} = \frac{1}{n}(\sum_{k=1}^n a'_{i,k}) \text{ za } i, j = \overline{1..n}$$

Kako je  $A$  Hermitska matrica tada iz (4.4) sledi

$$\begin{aligned} t'_{i,j} &= \frac{1}{n} \left( \sum_{k=1}^{n-i+1} a_{k,i+k-1} + \sum_{k=n-i+2}^n a_{k,i+k-n-1} \right) \\ &= \frac{1}{n} \left( \sum_{k=1}^{n-i+1} a_{k,i+k-1} + \sum_{k=n-i+2}^n \overline{a_{1+k-n-1,k}} \right) \end{aligned}$$

za  $i, j = \overline{1..n}$ .

Ako sa  $F' = (f'_{i,j})_{i,j=1}^n \in \mathbb{C}^{n \times n}$  obeležimo matricu koja u prvoj vrsti sadrži elemente koji imaju vrednost  $E$ , a u ostalim vrstama su sve nule, odnosno  $f'_{i,j} = \begin{cases} E, & i = 1 \\ 0, & i \neq 1 \end{cases}$  i neka je  $B' = \mathcal{F}^{-1}(B) = (b'_{i,j})_{i,j=1}^n$ , tada se izraz (4.3) može napisati na sledeći način

$$B' = A' - T' + F'$$

pa je  $B = \mathcal{F}(B') = \mathcal{F}(A' - T' + F')$ , a kako je funkcija  $\mathcal{F}$  linearna to važi da je

$$B = A - \mathcal{F}(T') + \mathcal{F}(F'). \quad (4.5)$$

Kako je prema (4.1)  $[\mathcal{F}(F')]_{i,j} = \begin{cases} f'_{j-i+1,i}, & j \geq i \\ f'_{n+1+j-i,i}, & j < i \end{cases}$  za  $i, j = \overline{1..n}$  to je  $[\mathcal{F}(F')]_{i,j} = \begin{cases} E, & i = j \\ 0, & i \neq j \end{cases}$  za  $i, j = \overline{1..n}$  pa je matrica  $\mathcal{F}(F')$  dijagonalna, a samim tim i Hermitska.

Posmatrajmo sada  $[\mathcal{F}(T')]_{i,j}$  za  $i < j$ :

$$[\mathcal{F}(T')]_{i,j} = t'_{n+1+j-i,i}$$

$$\begin{aligned}
 &= \frac{1}{n} \left( \sum_{k=1}^{n-j+i} a_{k,j-i+k} + \sum_{k=n-j+i+1}^n a_{k,j-i+k-n} \right) \\
 &= \frac{1}{n} \left( \sum_{k=1}^{n-j+i} a_{k,j-i+k} + \sum_{k=n-j+i+1}^n \overline{a_{j-i+k-n,k}} \right) \\
 &= \frac{1}{n} \left( \sum_{k=1}^{j-i} \overline{a_{k,k+n-j+1}} + \sum_{k=1}^{n-j+i} a_{k,j-i+k} \right) \\
 &= \frac{1}{n} \left( \sum_{k=1}^{j-1} a_{k,n+k-j+1} + \sum_{k=1}^{n-j+1} \overline{a_{k,k+j-1}} \right) \\
 &= \frac{1}{n} \left( \sum_{k=1}^{j-1} a_{k,n+k-j+1} + \sum_{k=n-j+1+1}^n \overline{a_{k-j+1,k}} \right) \\
 &= \frac{1}{n} \left( \sum_{k=1}^{j-i} a_{k,n+k-j+i} + \sum_{k=n-j+i+1}^n a_{k,k-j+i} \right) \\
 &= t'_{n+1+i-j,j} = [\mathcal{F}(T')]_{j,i}
 \end{aligned}$$

Poslednja jednakost pokazuje da je matrica  $\mathcal{F}(T')$  Hermitska, pa iz (4.5) važi da je i matrica B Hermitska. ■

**Napomena 4.2.1.** *Primetimo da za matrice A i T = F(T') iz dokaza prethodnog stava važi:*

- elementi glavne dijagonale matrice T imaju vrednost traga matrice A podeljenog sa n,
- elementi i-te naddijagonale i (n - i)-te poddijagonale (1 ≤ i ≤ n - 1) matrice T imaju vrednost zbira elemenata i-te naddijagonale i (n - i)-te poddijagonale matrice A podeljenog sa n.

Ako sa  $tr_0$  označimo trag matrice A, sa  $tr_i, 1 \leq i \leq n - 1$  zbir elemenata i-te naddijagonale, a sa  $tr_{-i}, 1 \leq i \leq n - 1$  zbir elemenata i-te poddijagonale, sa  $t_i = \frac{1}{n}(tr_i + tr_{-(n-i)})$  za  $1 \leq i \leq n - 1$ , a sa  $t_0 = \frac{tr_0}{n}$ , tada matrica T izgleda ovako:

$$T = \frac{1}{n} \begin{bmatrix} t_0 & t_1 & t_2 & \dots & \overline{t_2} & \overline{t_1} \\ \overline{t_1} & t_0 & t_1 & \dots & \overline{t_3} & \overline{t_2} \\ \overline{t_2} & \overline{t_1} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & t_1 & t_2 \\ t_2 & t_3 & \dots & \overline{t_1} & t_0 & t_1 \\ t_1 & t_2 & \dots & \overline{t_2} & \overline{t_1} & t_0 \end{bmatrix} \quad (4.6)$$

Stav 4.2.3 pokazuje da će se počev od neke Hermitske matrice  $X_0$  iz  $\Gamma_{\mathcal{P}}(\mathbf{E})$  naizmeničnim projekcijama zasnovanim na teoremama 2.2.1 i 2.2.2 dobiti niz Hermitskih matrica  $X_1, X_2, \dots, X_s \in \Gamma_{\mathcal{P}}(\mathbf{E})$  i  $Y_1, Y_2, \dots, Y_s \in \Lambda_{\mathcal{P}}(\mathbf{r})$ .

Slično se dokazuje i sledeći stav koji pokazuje da se počev od neke matrice  $X_0$  iz  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  naizmeničnim projekcijama zasnovanim na teoremama 2.3.1 i 2.3.2 dobiti niz Hermitskih matrica  $X_1, X_2, \dots, X_s \in \Gamma_{\mathcal{AP}}(\mathbf{E})$  i  $Y_1, Y_2, \dots, Y_s \in \Lambda_{\mathcal{AP}}(\mathbf{r})$ .

**Stav 4.2.3.** Neka je  $A = (a_{i,j})_{i,j=1}^{2n-1} \in \mathbb{C}^{(2n-1) \times (2n-1)}$  Hermitska matrica. Neka je  $B = (b_{i,j})_{i,j=1}^{2n-1}$  matrica iz  $\mathbb{C}^{(2n-1) \times (2n-1)}$  takva da za neko  $E \in \mathbb{R}$  važi:

$$[\mathcal{F}^{-1}(X)]_{k,l} = [\mathcal{F}^{-1}(Y)]_{k,l} + \frac{1}{\omega_k} (E \delta_{k-1} - \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M \odot Y)]_{k,l'})$$

gde je  $M = \begin{bmatrix} \mathbf{1}_{n \times n} & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}$  i  $\omega_k = \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M)]_{k,l'} = \begin{cases} n-k+1, & k \leq n \\ k-n, & k > n \end{cases}$  za  $1 \leq k \leq n$ . Tada je i  $B$  Hermitska matrica. ■

**Napomena 4.2.2.** Ako za matricu  $A$  iz prethodnog stava obeležimo sa  $T' = \frac{1}{\omega_k} \sum_{l'=1}^{2n-1} [\mathcal{F}^{-1}(M \odot Y)]_{k,l'}$  matricu definisanu sa  $t'_{i,j} = \frac{1}{\omega_i} (\sum_{k=1}^{2n-1} a'_{i,k})$  za  $i, j = \{1, \dots, 2n-1\}$  tada za matrice  $A$  i  $T = \mathcal{F}(T')$  važi:

- elementi glavne dijagonale matrice  $T$  imaju vrednost traga matrice  $A$  podeljenog sa  $\omega_1$ ,
- kada je  $(1 \leq i \leq n-1)$  elementi  $i$ -te naddijagonale i  $(2n-1-i)$ -te poddijagonale matrice  $T$  imaju vrednost zbira elemenata  $i$ -te naddijagonale matrice  $A$  podeljenog sa  $\omega_{i+1}$  i
- kada je  $(n \leq i \leq 2n-2)$  elementi  $i$ -te naddijagonale i  $(2n-1-i)$ -te poddijagonale matrice  $T$  imaju vrednost zbira elemenata  $(2n-1-i)$ -te poddijagonale matrice  $A$  podeljenog sa  $\omega_{i+1}$ .

Ako sa  $tr_0$  označimo trag matrice  $A$ , sa  $tr_k, 1 \leq k \leq n-1$  zbir elemenata  $i$ -te naddijagonale, a sa  $tr_{-k}, 1 \leq k \leq n-1$  zbir elemenata  $i$ -te poddijagonale, sa

$$t_k = \begin{cases} \frac{1}{\omega_{k+1}} tr_k, & 0 \leq k \leq n-1 \\ \frac{1}{\omega_{k+1}} tr_{-(2n-1-k)}, & n \leq k \leq 2n-2 \end{cases}$$

tada matrica  $T$  izgleda ovako:

$$T = \begin{bmatrix} \boxed{\frac{t_0}{\omega_1}} & \dots & \boxed{\frac{t_{n-1}}{\omega_{n-1}}} & & & \\ \vdots & \ddots & \vdots & & & \\ \boxed{\frac{t_{n-1}}{\omega_{n-1}}} & \dots & \boxed{\frac{t_0}{\omega_0}} & & & \\ & & & & \mathbf{0}_{n \times (n-1)} & \\ & & & & & & \mathbf{0}_{(n-1) \times n} & & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \quad (4.7)$$

Stav 4.2.3 pokazuje da će se počev od neke Hermitske matrice  $X_0$  iz  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  naizmeničnim projekcijama zasnovanim na teoremama 2.3.1 i 2.3.2 kroz s iteracija dobiti niz Hermitskih matrica



$$X_i = \begin{bmatrix} X'_i & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \in \Gamma_{\mathcal{AP}}(\mathbf{E}) \text{ i } Y_i = \begin{bmatrix} Y'_i & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \in \Lambda_{\mathcal{AP}}(\mathbf{r}) \text{ gde su} \\ X'_i \in \Gamma_{\mathcal{P}} \text{ i } Y'_i \in \Lambda_{\mathcal{P}}(\mathbf{r}) \text{ za } i \in \{1, \dots, s\}.$$

### $\Gamma_{\mathcal{P}}(\mathbf{E})$ u $\Lambda_{\mathcal{P}}(\mathbf{r})$ i $\Gamma_{\mathcal{AP}}(\mathbf{E})$ u $\Lambda_{\mathcal{AP}}(\mathbf{r})$

Preslikavanje skupa  $\Gamma_{\mathcal{P}}(\mathbf{E})$  u  $\Lambda_{\mathcal{P}}(\mathbf{r})$  za periodične autokorelacije određeno teoremom 2.2.1 implementirano je u funkciju „GL“ koja prvo odredi dekompoziciju koristeći funkciju „eigh“ iz modula scipy (koja se koristi za hermitske matrice i vraća niz sopstvenih vrednosti i matricu sopstvenih vektora), a zatim zameni dijagonalnu matricu sa

$$D = \begin{bmatrix} r_0 & \dots & 0 & & \\ \vdots & \ddots & \vdots & & \\ 0 & \dots & r_k & & \\ \mathbf{0}_{k \times (n-k)} & & & \mathbf{0}_{(n-k) \times (n-k)} & \end{bmatrix}.$$

Ako je  $k = 1$ , odnosno kada je potrebno pronaći jedan niz, koristi se metod stepenovanja umesto dekompozicije.

Funkcija „GuL“ vraća matrice  $VDV^H$  i  $V\sqrt{D}$ . Ova poslednja se koristi kod binarne azbuke da bi se razdvojili nizovi koji su generisani u toj iteraciji, a sadrže se u kolonama matrice  $V\sqrt{D}$ .

Prema napomeni 4.2.2 matrica  $X \in \Gamma_{\mathcal{AP}}(\mathbf{E})$  ima oblik  $X = \begin{bmatrix} X' & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix}$  gde je  $X' \in \Gamma_{\mathcal{P}}(\mathbf{E})$  i ako je  $VD'V^H$  dekompozicija matrice  $X'$  tada je dekompozicija matrice  $X$  jednaka:

$$X = \begin{bmatrix} V & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & I_{(n-1)} \end{bmatrix} \begin{bmatrix} D' & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \begin{bmatrix} V^H & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & I_{(n-1)} \end{bmatrix} \\ = \begin{bmatrix} VD'V^H & \mathbf{0}_{n \times (n-1)} \\ \mathbf{0}_{(n-1) \times n} & \mathbf{0}_{(n-1) \times (n-1)} \end{bmatrix} \in \Lambda_{\mathcal{AP}}(\mathbf{r})$$

Zato će zbog povećanja efikasnosti programa samo prvih  $n$  redova i prvih  $n$  kolona matrice  $X \in \Gamma_{\mathcal{AP}}(\mathbf{E})$  činiti ulaz funkcije „GL“ dok će rezultat biti matrice:

- $VDV^H$  (koja čini prvih  $n$  redova i prvih  $n$  kolona matrice iz  $\Lambda_{\mathcal{AP}}(\mathbf{r})$ ) i
- $V\sqrt{D}$  (koja se koristi za slučaj binarne azbuke).

### $\Lambda_{\mathcal{P}}(\mathbf{r})$ u $\Gamma_{\mathcal{P}}(\mathbf{E})$

Preslikavanje skupa  $\Lambda_{\mathcal{P}}(\mathbf{r})$  u  $\Gamma_{\mathcal{P}}(\mathbf{E})$  određeno teoremom 2.2.2 implementirano je korišćenjem funkcije „LpGp“ koja prvo kreira Toeplitz-ovu matricu  $T$  dimenzije  $n \times n$  na način opisanu u napomeni 4.2.3. Zatim se za ulaznu matricu  $Y$  dimenzije  $n \times n$  računa razlika  $Y - T$  kojoj se dodaje dijagonalna matrica dimenzije  $n \times n$  sa svim elementima na dijagonali jednakim  $\frac{E}{n}$ .

## $\Lambda_{\mathcal{AP}}(\mathbf{r})$ u $\Gamma_{\mathcal{AP}}(\mathbf{E})$

Preslikavanje skupa  $\Lambda_{\mathcal{AP}}(\mathbf{r})$  u  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  određeno teoremom 2.3.2 implementirano je u funkciju „LapGap“ koja slično funkciji LpGp koristi Toeplitz-ovu matricu  $T$  dimenzije  $n \times n$  na način opisan u napomeni 4.2.3. Zatim se za ulaznu matricu  $Y$  dimenzije  $n \times n$  računa razlika  $Y - T$  kojoj se dodaje dijagonalna matrica dimenzije  $n \times n$  sa svim elementima na dijagonali jednakim  $\frac{E}{n}$ . Ovde je iskorišćena činjenica da su elementi matrice iz  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  i  $\Lambda_{\mathcal{AP}}(\mathbf{r})$  jednaki nuli svuda osim u prvom  $n \times n$  delu pa zbog povećanja efikasnosti programa samo prvih  $n$  redova i prvih  $n$  kolona tih tih matrica čine ulaz i izlaz funkcije „LapGap“.

### 4.2.3 Generisanje početne matrice

Za kreiranje početne matrice  $X_0$  koristi se procedure „generisiPocetnuMatricuP“ (za periodične autokorelacije) i „generisiPocetnuMatricuAP“ (za aperiodične autokorelacije) koje bi trebalo da se pozovu pre procedura „ITROXP“ i „ITROXAP“.

Procedura „generisiPocetnuMatricuP“ prvo kreira „slučajni“ niz **niz0** dužine  $n$  sa elementima iz intervala  $(-1,1)$ , a zatim kreira matricu  $P$  kao uvijeni proizvod niza **niz0** sa samim sobom, tj.  $P = \mathbf{niz0} \cup \mathbf{niz0}^H$  koja se preslikava u  $X_0$  korišćenjem funkcije „LpGp“.

Procedura „generisiPocetnuMatricuAP“ prvo kreira „slučajni“ niz **niz0** dužine  $n$  sa elementima iz intervala  $(-1,1)$  i proširuje ga sa  $n - 1$  nula, a zatim kreira matricu  $P$  kao uvijeni proizvod niza **niz0** sa samim sobom, tj.  $P = \mathbf{niz0} \cup \mathbf{niz0}^H$  koja se preslikava u  $X_0$  korišćenjem funkcije „LapGap“.

Funkcija koja računa uvijeni proizvod pod nazivom „uvijeniProizvod“ prvo računa proizvod  $\mathbf{niz0} \cdot \mathbf{niz0}^H$ , a zatim koristi funkciju „F\_permutacija\_inv“ (implementacija funkcije  $\mathcal{F}^{-1}$  koristeći stav 4.2.2).

### 4.2.4 Procedure „ITROXP“ i „ITROXAP“

Procedure „ITROXP“ i „ITROXAP“ predstavljaju implementacije algoritama ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  opisanih u drugom poglavlju.

Ulazni atribut procedura je:

- **izborAzbuke** - string koji može imati vrednost „realni“ ili „binarni“, a određuje azbuku nad kojom se generišu skupovi nizova.

Promenljive koje procedura vraća su:

- **broj\_koraka** - broj iteracija izabranog algoritma
- **razlika** - euklidska norma razlike matrica  $X$  i  $Y$  dobijenih u poslednjoj iteraciji
- **log\_tekst** - tekst koji će se prepisati u logTextBox glavnog prozora
- **psl** - PSL pri poslednjoj iteraciji algoritma

- **isl** – ISL pri poslednjoj iteraciji algoritma
- **mf** – MF pri poslednjoj iteraciji algoritma

Rezultat rada procedura je kreiranje matrice **S** dimenzije  $\mathbf{k} \times \mathbf{n}$  koja u redovima sadrži elemente **k** nizova dužine **n**. Ukoliko je izabrana opcija da se pronalaze binarni nizovi, formira se još i matrica **S<sub>-</sub>** iste dimenzije kao **S** koja sadrži nizove realnih vrednosti dok matrica **S** sadrži vrednosti iz **S<sub>-</sub>** klipovane na skup  $\{-1,1\}$ . Formira se i matrica **matricaUvijeniProizvod** koja predstavlja zbir uvijenih proizvoda nizova koji su smešteni u redovima matrice **S**.

### ITROXP

Glavna petlja procedure „ITROXP“ za periodične autokorelacije prvo preslikava matricu  $X$  u  $Y$  koristeći funkciju „GL“.

Ukoliko je potrebno pronaći nizove iz binarne azbuke koristi se funkcija iz primera 2.4.1 pod (a) koja je implementirana u funkciju „f\_za\_binarne“. Kako funkcija „GL“ vraća matrice matrice  $Y = VDV^H$  i  $S = V\sqrt{D}$  prvo se izračuna  $S' = f\_za\_binarne(\mathbf{n}, S, i, 0.0001)$  gde je  $i$  broj trenutne iteracije, a zatim se izračuna  $Y = S' \cdot S'^H$ . Još se računa  $S1 = np.sign(S')$  kao  $\lim_{i \rightarrow \infty} f\_za\_binarne(\mathbf{n}, S, i, 0.0001)$  i  $Y1 = S1 \cdot S1^H \in \Omega_X$ .

Zatim, koristeći funkciju „LpGp“ matrica  $Y$  se preslikava u  $X$ .

Pri kraju svake iteracije do-until petlje, računa se euklidska norma razlike matrica  $X$  i  $Y$  dobijenih u toj iteraciji i dodeljuje promenljivoj „razlika“. Za binarnu azbuku računa se i norma razlike matrica  $Y$  i  $Y1$  i dodeljuje promenljivoj „razlika\_Y1\_Y“.

Izlazak iz petlje je omogućen ako je razlika  $< 10^{-13}$  ili razlika\_Y1\_Y  $< 0.3\mathbf{n}$  ili ako je broj iteracija veći od 2701.

Na kraju se funkcijom „izdvojSekvence“ formira matrica **S** dimenzije  $\mathbf{k} \times \mathbf{n}$  u čijim redovima su smešteni rezultujući nizovi. Ukoliko se traže binarni nizovi, formira se i matrica **S<sub>-</sub>**. Funkcija „izdvojSekvence“ koristi matricu  $X$  dobijenu u poslednjoj iteraciji tako što prvo izračuna dekompoziciju matrice  $X$  i tako dobije matricu  $V$  (koja sadrži sopstvene vektore matrice  $X$ ). Kako je  $VDV^H = (V\sqrt{D})(V\sqrt{D})^H$  gde je  $D$  dijagonalna matrica sa vrednostima  $\mathbf{r}$  na dijagonali to se traženi nizovi nalaze u kolonama matrice  $V\sqrt{D}$ , ali se zbog lakšeg pregleda **S** dobija od prvih  $\mathbf{k}$  vrsta matrice  $(V\sqrt{D})^T = \sqrt{D}V^T$ . Tako se traženi nizovi nalaze u redovima matrice **S**.

### ITROXAP

Ukoliko se radi sa aperiodičnim autokorelacijama poziva se procedura „ITROXAP“ koja za preslikavanja koristi funkcije „GL“ i „LpaGpa“ kao i sličan postupak u slučaju da je izabrana binarna azbuka. Ovde se kao u proceduri „ITROXP“ računaju „razlika“ i „razlika\_Y1\_Y“, a izlazak iz petlje je omogućen ako je razlika  $< 10^{-3}$  ili razlika\_Y1\_Y  $< 0.3\mathbf{n}$  ili ako je broj iteracija veći od 2701. Na kraju se, takođe, koristi funkcija „izdvojSekvence“ da bi se formirale matrice **S** dimenzije  $\mathbf{k} \times \mathbf{n}$  i **S<sub>-</sub>** od matrice  $X$  dobijene u poslednjoj iteraciji.

## Složenost

Složenost funkcije „ITROXP” je jednaka složenosti dekompozicije Hermitskih matrica  $\mathcal{O}(\mathbf{k} \cdot \mathbf{n}^2)$ . Kako „ITROXP” koristi istu funkciju kao „ITROXP” za dekompoziciju  $\mathbf{n} \times \mathbf{n}$  nenula dela matrica, složenost je takođe jednaka  $\mathcal{O}(\mathbf{k} \cdot \mathbf{n}^2)$ . Algoritam CAN za unimodularne nizove je efikasniji od ITROX algoritama zato što koristi brzu Furijeovu transformaciju. Ove godine razvijen je algoritam CANARY kao proširenje CAN algoritma) za komplementarne skupove nizova koji je takođe baziran na brznoj Furijeovoj transformaciji i ima složenost  $\mathcal{O}(\mathbf{k} \cdot \mathbf{n} \cdot \log \mathbf{n})$  [9] (str. 2097).

## 5 Rezultati rada programa ITROX

Primere korišćenja programa ITROX u zavisnosti od izbora algoritma (za periodične ili aperiodične autokorelacije) i izbora azbuke (skup realnih brojeva ili binarni skup  $\{-1, 1\}$ ) ćemo radi lakšeg prikazivanja rezultata obeležiti različitim oznakama koje prikazuje sledeća tabela:

oznaka	opis	Ulazni parametri programa ITROX		
		Izbor algoritma	Izbor azbuke	k
PR1	Pronalaženje realnog niza sa dobrim periodičnim autokorelacionim osobinama	ITROX-P	realni	1
PB1	Pronalaženje binarnog niza sa dobrim periodičnim autokorelacionim osobinama	ITROX-P	binarni	1
PR2	Pronalaženje realnog para sa dobrim periodičnim autokorelacionim osobinama	ITROX-P	realni	2
PB2	Pronalaženje binarnog para sa dobrim periodičnim autokorelacionim osobinama	ITROX-P	binarni	2
APR1	Pronalaženje realnog niza sa dobrim aperiodičnim autokorelacionim osobinama	ITROX-AP	realni	1
APB1	Pronalaženje binarnog niza sa dobrim aperiodičnim autokorelacionim osobinama	ITROX-AP	binarni	1
APR2	Pronalaženje realnog para sa dobrim aperiodičnim autokorelacionim osobinama	ITROX-AP	realni	2
APB2	Pronalaženje binarnog para sa dobrim aperiodičnim autokorelacionim osobinama	ITROX-AP	binarni	2

Tabela 5.1 Oznake primera korišćenja programa ITROX

### 5.1 Brzina izvršavanja različitih implementacija programa ITROX

U zavisnosti od načina implementacije projekcija skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  i  $\Lambda_{\mathcal{P}}(\mathbf{r})$  (ili  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  i  $\Lambda_{\mathcal{AP}}(\mathbf{r})$ ) menja se vreme pronalaženja odgovarajućih nizova. Tako možemo posmatrati četiri različite implementacije:

- „ITROX” - verzija opisana u četvrtom poglavlju
- „ITROX sa proširenim matricama” - verzija koja umesto funkcije „GL” ima dve funkcije „GpLp” i „GapLap” gde je funkcija „GpLp” identična sa „GL” i koristi se kod periodičnih nizova, a za aperiodične nizove koristi se funkcija „GapLap” koja je veoma slična sa „GL” sa tom razlikom što radi sa matricama dimenzija  $(2\mathbf{n} - 1) \times (2\mathbf{n} - 1)$ . U ovoj verziji se i funkcija „LapGap” menja tako što se takođe radi sa matricama dimenzija  $(2\mathbf{n} - 1) \times (2\mathbf{n} - 1)$ .
- „ITROX sa permutacijama” - verzija u kojoj se funkcija „LpGp” menja tako što se ulazna matrica  $Y$  permutuje funkcijom „F\_permutacija\_inv” u matricu  $P$ , a zatim se računa matrica  $T$  koja u  $k$ -tom redu ( $1 \leq k \leq n$ ) ima elemente jednake sumi elemenata  $k$ -tog reda matrice

$P$ , a elementima prvog reda se još oduzme vrednost  $\frac{E}{n}$ . Sada se računa razlika  $P - T$  koja se potom permutuje funkcijom „F\_permutacija“. Slično se menja i funkcija „LapGap“.

- (d) „ITROX sa proširenim matricama i permutacijama“ - verzija koja ima izmene kao verzije (b) i (c).

Tabela 4.2 prikazuje vremena trajanja za navedene četiri implementacije za računare od 1GB RAM i 1.83 GHz CPU i 3GB RAM i 2.5 GHz CPU za svaki od osam različitih primera:

	ITROX		ITROX sa proširenim matricama		ITROX sa permutacijama		ITROX sa proširenim matricama i permutacijama	
	računar: 1GB RAM 1.83 GHz	računar: 3GB RAM 2.5 GHz	računar: 1GB RAM 1.83 GHz	računar: 3GB RAM 2.5 GHz	računar: 1GB RAM 1.83 GHz	računar: 3GB RAM 2.5 GHz	računar: 1GB RAM 1.83 GHz	računar: 3GB RAM 2.5 GHz
<b>Niz sa dobrim periodičnim autokorelacionim osobinama</b>								
PR1 - realni nizovi	4	3	4	3	11.5	8	11.5	8
PB1 - binarni nizovi	1.2	1	1.2	1	2.8	2	2.8	2
<b>Komplementarni par sa dobrim periodičnim autokorelacionim osobinama</b>								
PR2 - realni nizovi	8	6	8	6	16	11	16	11
PB2 - binarni nizovi	2.5	2	2.5	2	4.65	3	4.65	3
<b>Niz sa dobrim aperiodičnim autokorelacionim osobinama</b>								
APR1 - realni niz	11	9	16	14	58	42	60	49
APB1 - binarni niz	2.3	2	4.2	4	8.7	6	10.28	9
<b>Komplementarni par sa dobrim aperiodičnim autokorelacionim osobinama</b>								
APR2 - realni nizovi	7	6	9.2	10	24	18	26	22
APB2 - binarni nizovi	4	3	6.3	7	11.68	9	13.6	11

Tabela 5.2 Brzine različitih implementacija (u sekundama)

## 5.2 Testiranje programa ITROX

Rezultati koji će biti prikazani u ovom delu su dobijeni primenom sledeća dva različita testa za svaki od primera navedenih u tabeli 4.1.:

**Test A:** Program ITROX je pokrenut 100 puta za dužinu nizova 64 i sa energetske komponentama jednakim 64. Svako pokretanje programa je inicirano generisanjem „slučajne“ realne početne sekvence dužine 64.

Dodatak A sadrži (za svaki od primera navedenih u tabeli 4.1. ) sledeće grafike:

- Prvi grafik prikazuje raspodelu PSL, ISL i MF za 100 nizova ili parova nizova. Za binarne nizove prikazuju se PSL i MF dok se za realne nizove prikazuju PSL i ISL izraženi u decibelima po formuli:

$$PSL \text{ (dB)} = 20 \cdot \log \frac{PSL}{E} \quad \text{i} \quad ISL \text{ (dB)} = 10 \cdot \log \frac{2 \cdot ISL}{E^2} \quad (4.8)$$

- Drugi grafik prikazuje najbolji niz ili par nizova u odnosu na PSL, MF i potreban broj iteracija. Ovde se prikazuje i grafik autokorelacija dobijenog niza ili para nizova. Opet se za realne nizove autokorelacije prikazuju u decibelima po formuli:

$$c_k \text{ (dB)} = 20 \cdot \log \frac{c_k}{c_0}, 1 \leq k \leq (n - 1)$$

- Treći grafik prikazuje promenu PSL, ISL i MF kroz iteracije programa za pronalaženje nizova ili parova nizova prikazanih u drugom grafiku. Kao kod drugog grafika, za binarne nizove prikazuju se PSL i MF, a za realne nizove PSL i ISL izraženi u decibelima po formuli (4.8).

Tabela 4.3 daje uporedni pregled najboljih rezultata testa A sa rezultatima objavljenim u radu [1] (str. 2188-2191).

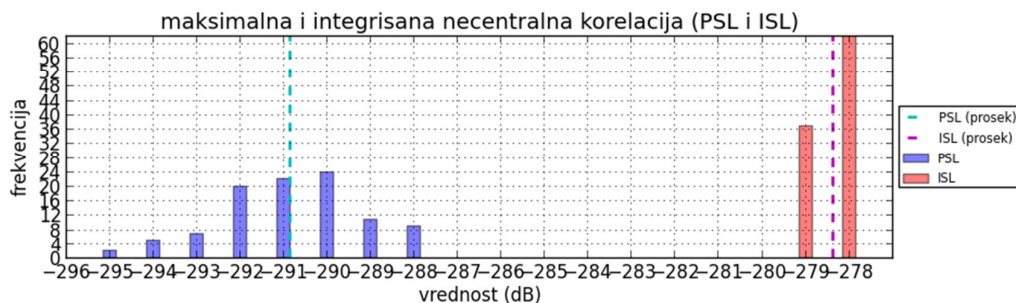
	Rezultati iz rada [1]		Rezultati programa ITROX						
	PSL	trajanje (sekundi)	PSL	računar: 1GB RAM 1.83 GHz	računar: 3GB RAM 2.5 GHz	broj iteracija	$\ X - Y\ $ u poslednjoj iteraciji	$\ Y - Y1\ $ u poslednjoj iteraciji	Faktor kvaliteta (MF)
<b>Niz sa dobrim periodičnim autokorelacionim osobinama</b>									
PR1	$5 \cdot 10^{-13}$	18	$10^{-13}$	4	3	1116	$10^{-13}$		
PB1	4	13	8	1.2	1	250	1.86	19.2	1.5
<b>Komplementarni par sa dobrim periodičnim autokorelacionim osobinama</b>									
PR2	$10^{-7}$	23	$2 \cdot 10^{-13}$	8	7	1660	$10^{-13}$		
PB2	8	16	8	2.5	3	534	1.9	19.1	6.4
<b>Niz sa dobrim aperiodičnim autokorelacionim osobinama</b>									
APR1	0.001	26	0.1	11	8	2702	0.12		
APB1	6	16	7	2.3	2	372	4.31	19.14	3.66
<b>Komplementarni par sa dobrim aperiodičnim autokorelacionim osobinama</b>									
APR2			0.0005	7	9.5	1863	$10^{-3}$		
APB2			8	4	2.5	392	2.93	19.2	10.34

Tabela 5.3 Pregled najboljih rezultata testa A

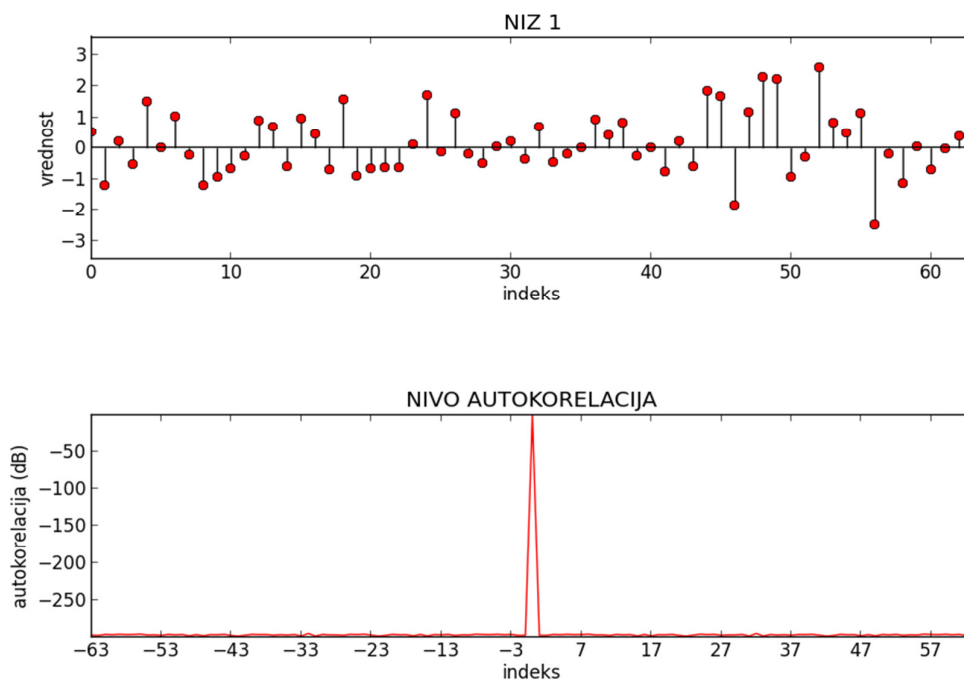
**Test B:** Program ITROX je za binarne nizove dužina od 2 do 100 pokrenut 10 puta. Energetske komponente su jednake odgovarajućim dužinama nizova. Svako pokretanje programa je inicirano generisanjem „slučajne“ realne početne sekvence odgovarajuće dužine. Dodatak B sadrži grafikone koji prikazuju najbolje dobijene vrednosti PSL.

## Dodatak A

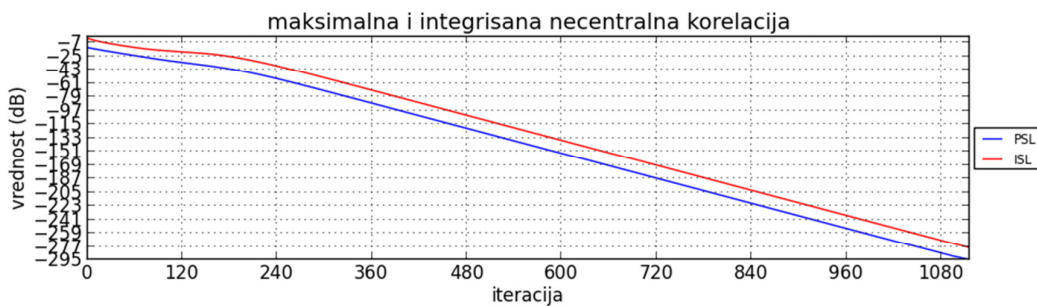
### PR1 - Pronalaženje realnog niza sa dobrim periodičnim autokorelacionim osobinama



Slika A.1 Raspodela PSL i ISL za 100 realnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama

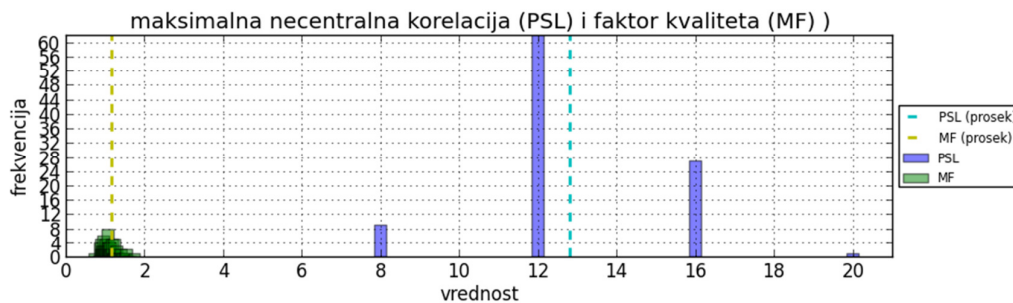


Slika A.2 Realni niz dužine 64 sa dobrim periodičnim autokorelacionim osobinama

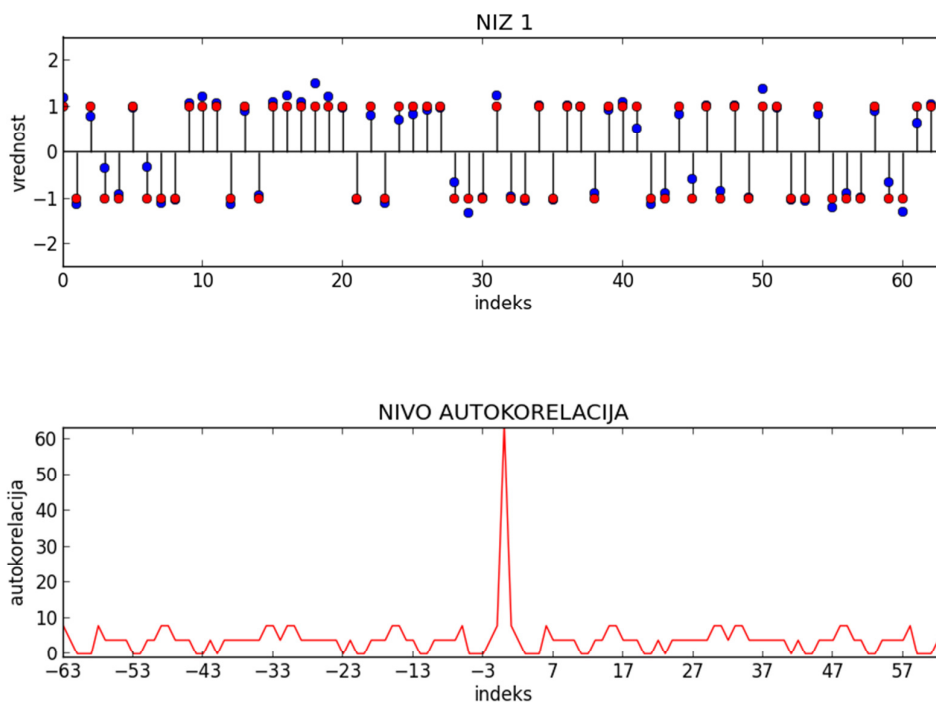


Slika A.3 PSL i ISL za realni niz sa slike A.2

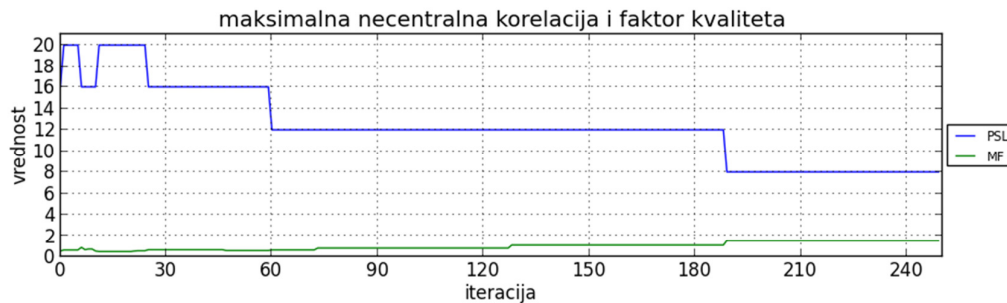
### PB1 - Pronalaženje binarnog niza sa dobrim periodičnim autokorelacionim osobinama



Slika A.4 Raspodela PSL i MF za 100 binarnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama



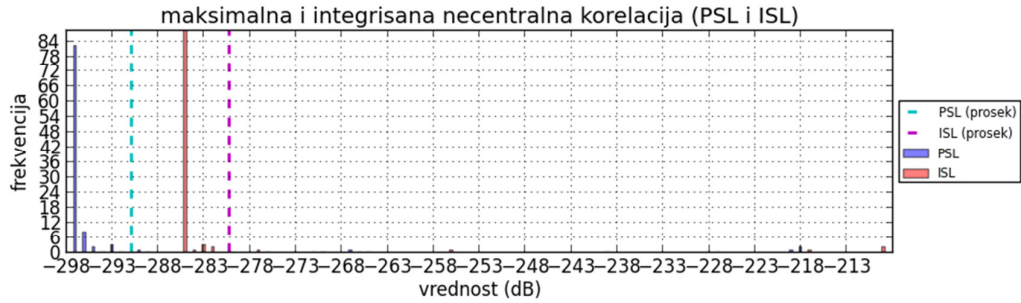
Slika A.5 Binarni niz dužine 64 sa dobrim periodičnim autokorelacionim osobinama



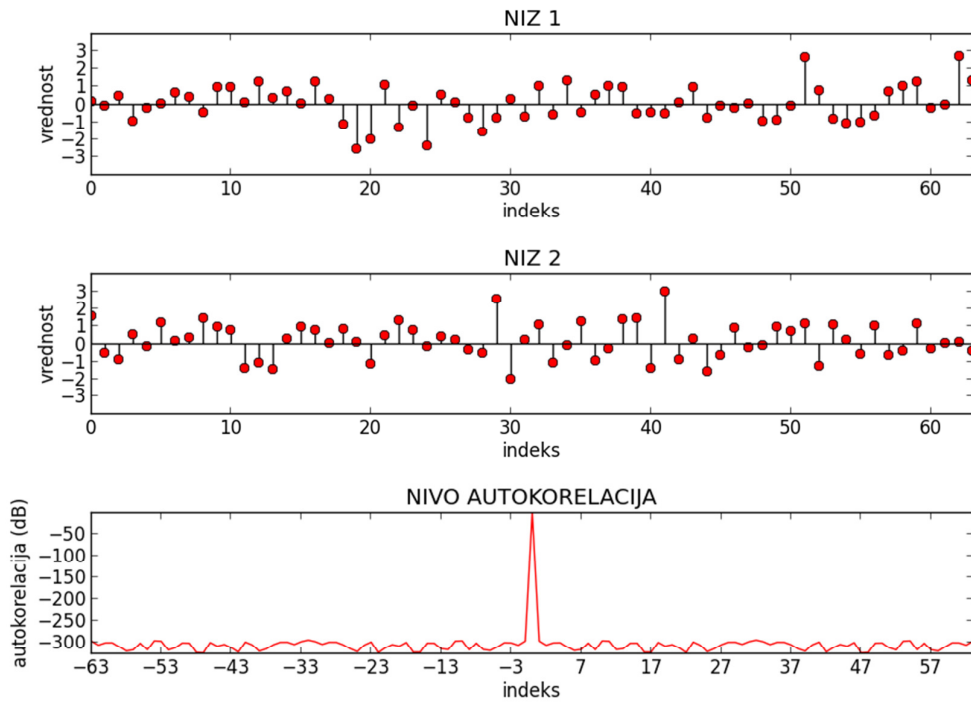
Slika A.6 PSL i MF za binarni niz sa slike A.5



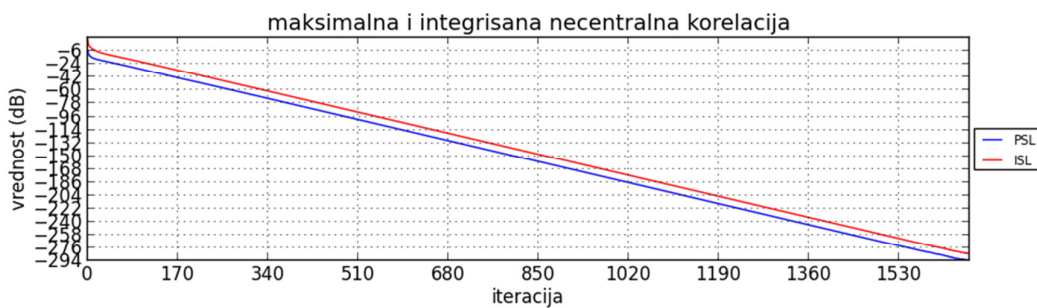
## PR2 - Pronalaženje komplementarnog para realnih nizova sa dobrim periodičnim autokorelacionim osobinama



Slika A.7 Raspodela PSL i ISL za 100 realnih parova dužine 64 sa dobrim periodičnim autokorelacionim osobinama

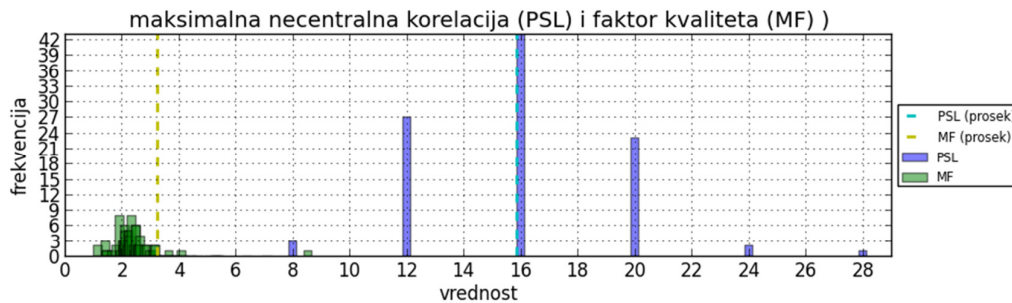


Slika A.8 Par realnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama

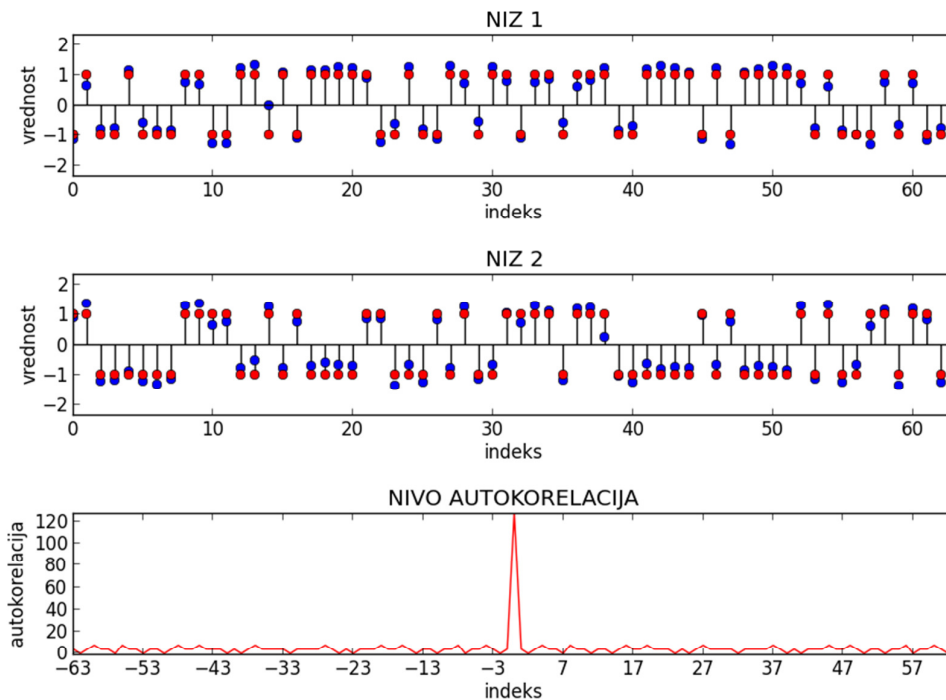


Slika A.9 PSL i ISL za par sa slike A.8

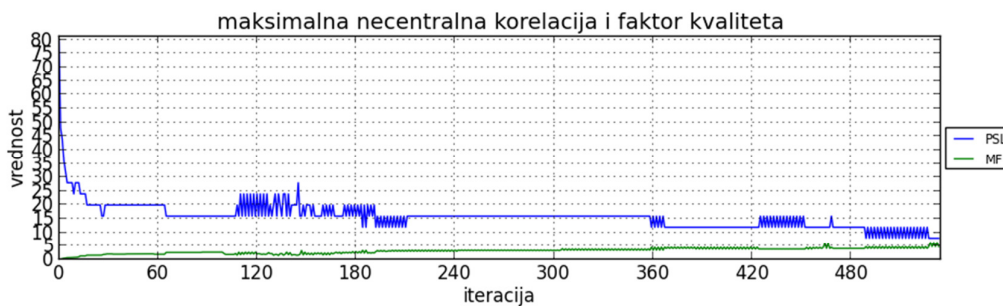
### PB2 - Pronalaženje komplementarnog para binarnih nizova sa dobrim periodičnim autokorelacionim osobinama



Slika A.10 Raspodela PSL i MF za 100 binarnih parova dužine 64 sa dobrim periodičnim autokorelacionim osobinama

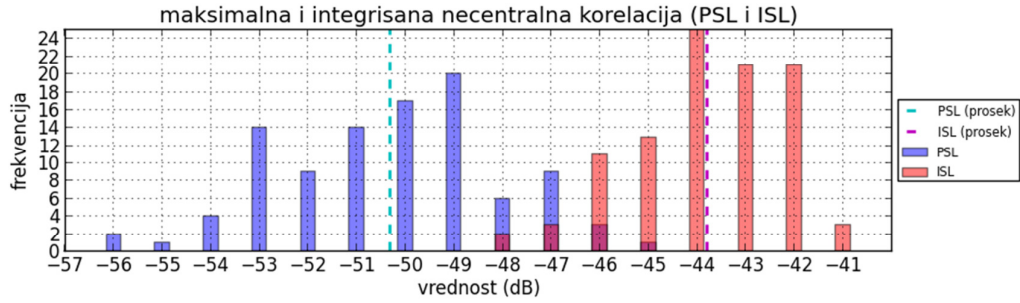


Slika A.11 Par binarnih nizova dužine 64 sa dobrim periodičnim autokorelacionim osobinama

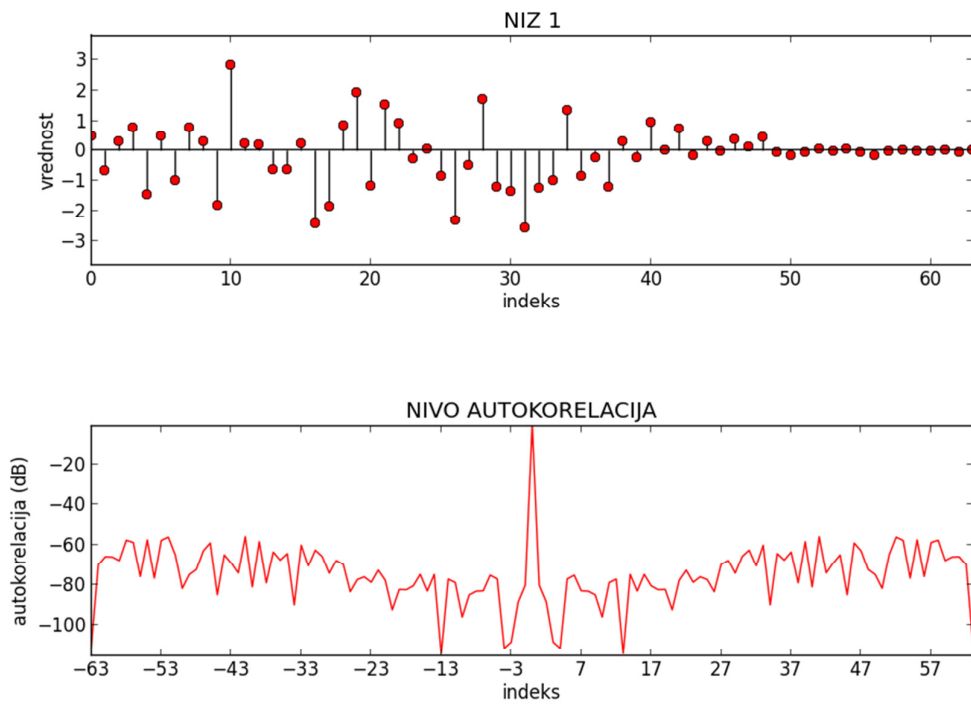


Slika A.12 PSL i MF za par sa slike A.11

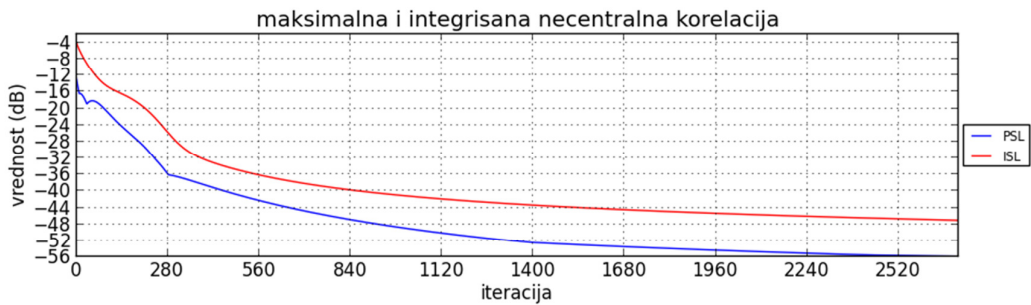
### APR1 - Pronalaženje realnog niza sa dobrim aperiodičnim autokorelacionim osobinama



Slika A.13 Raspodela PSL i ISL za 100 realnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

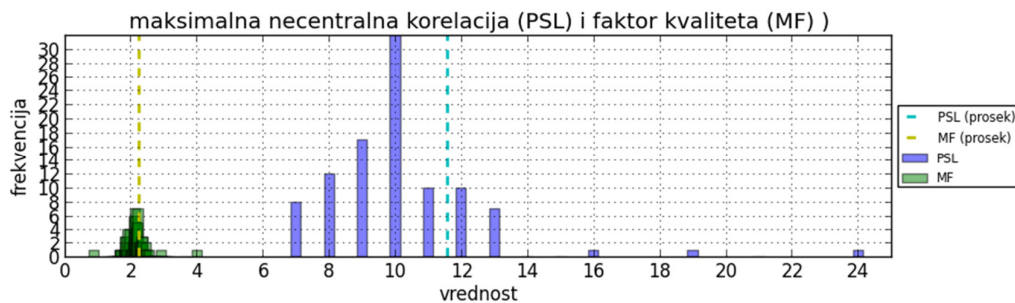


Slika A.14 Realni niz dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

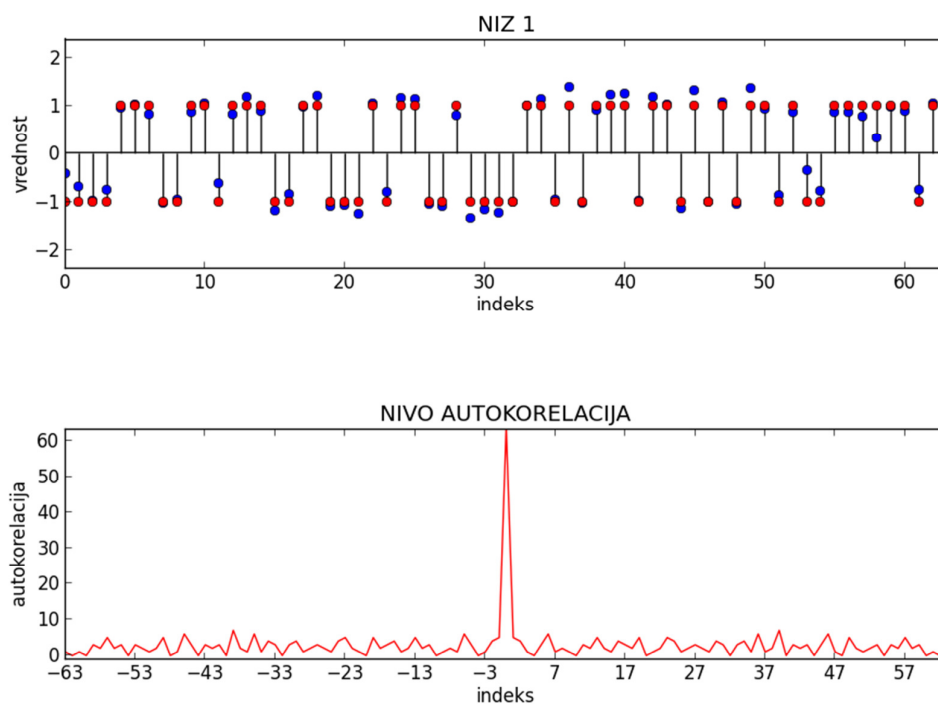


Slika A.15 PSL i ISL za realni niz sa slike A.14

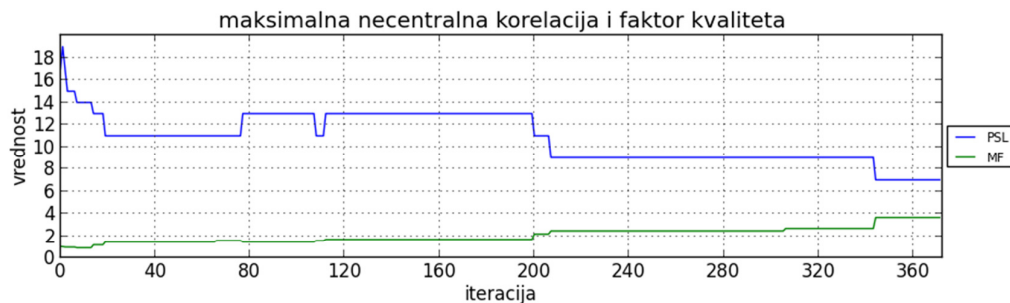
### APB1 - Pronalaženje binarnog niza sa dobrim aperiodičnim autokorelacionim osobinama



Slika A.16 Raspodela PSL i MF za 100 binarnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

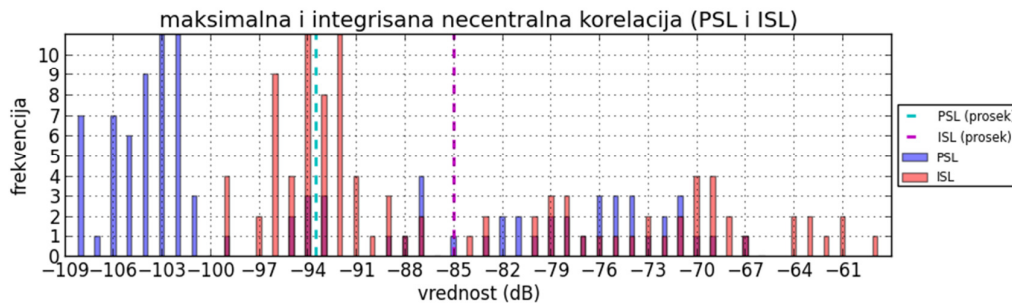


Slika A.17 Binarni niz dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

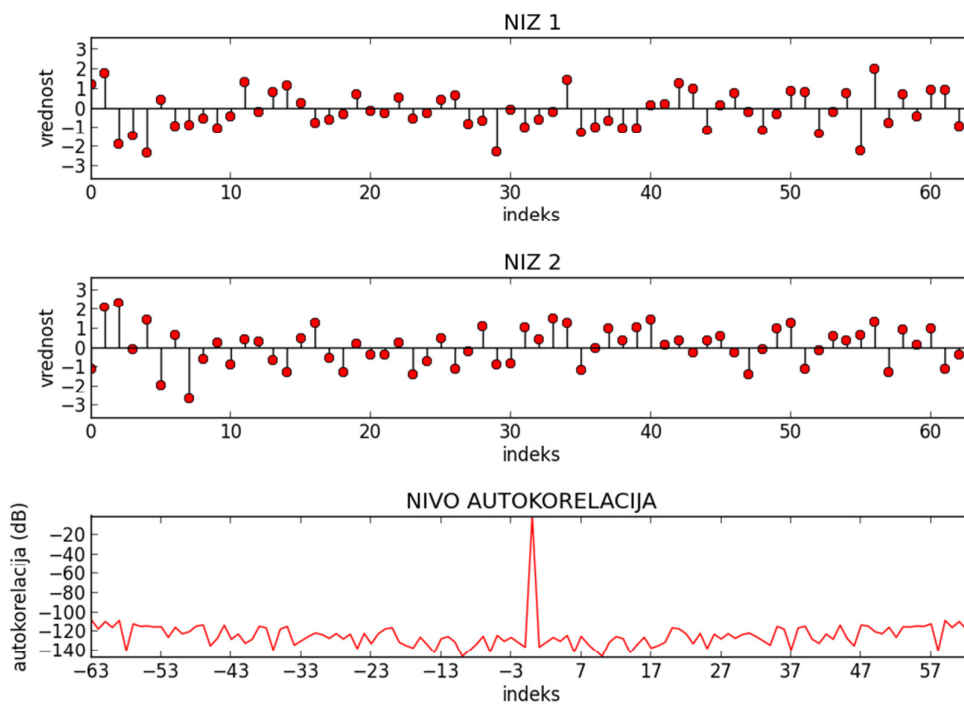


Slika A.18 PSL i MF za niz sa slike A.17

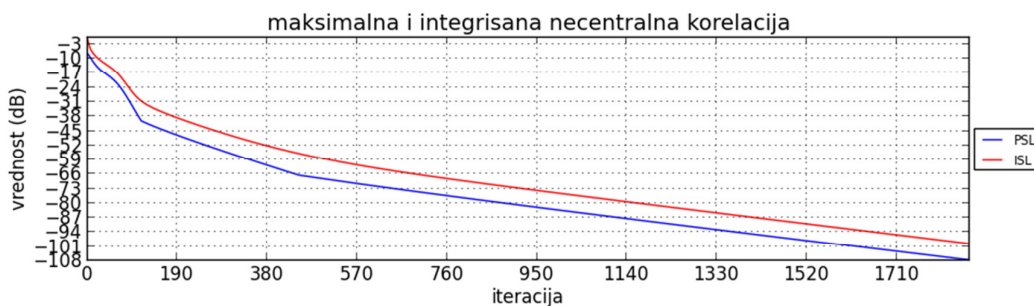
**APR2 - Pronalaženje komplementarnog para realnih nizova sa dobrim aperiodičnim autokorelacionim osobinama**



Slika A.19 Raspodela PSL i ISL za 100 realnih parova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

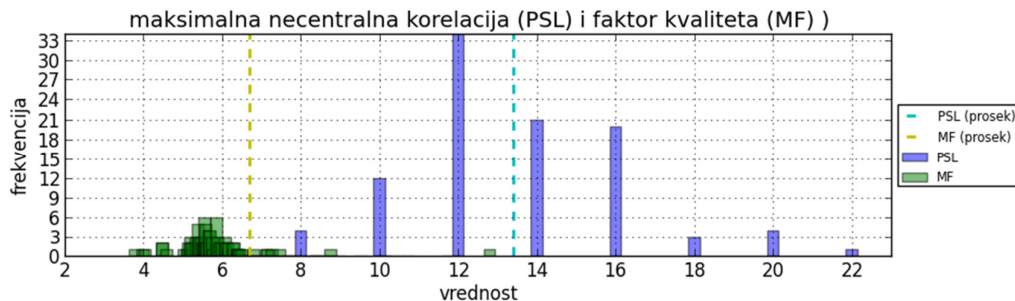


Slika A.20 Par realnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama

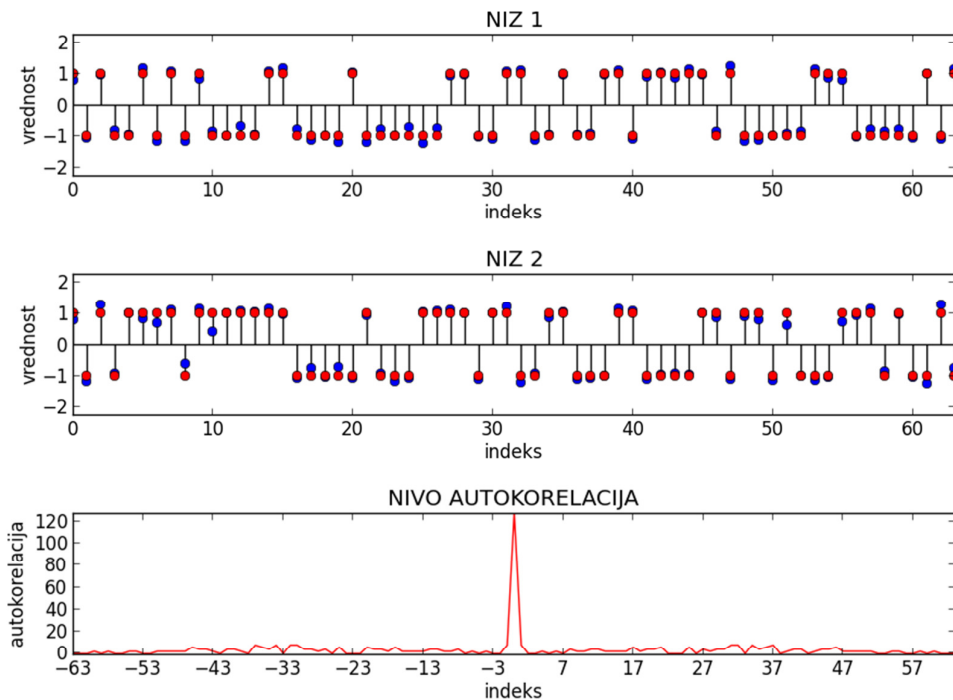


Slika A.21 PSL i ISL za par sa slike A.20

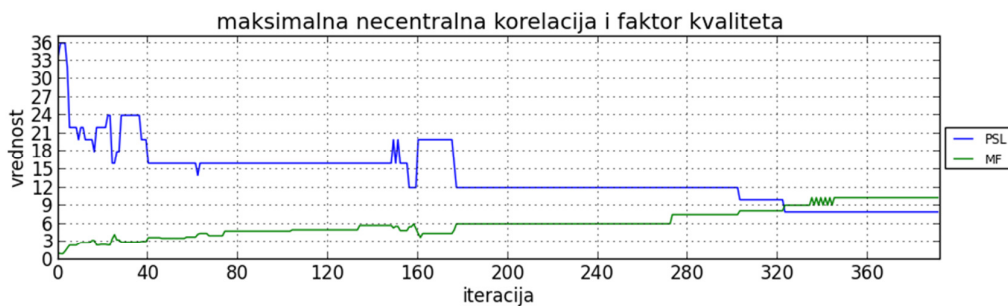
**APB2 - Pronalaženje komplementarnog para binarnih nizova sa dobrim aperiodičnim autokorelacionim osobinama**



Slika A.22 Raspodela PSL i MF za 100 binarnih parova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama



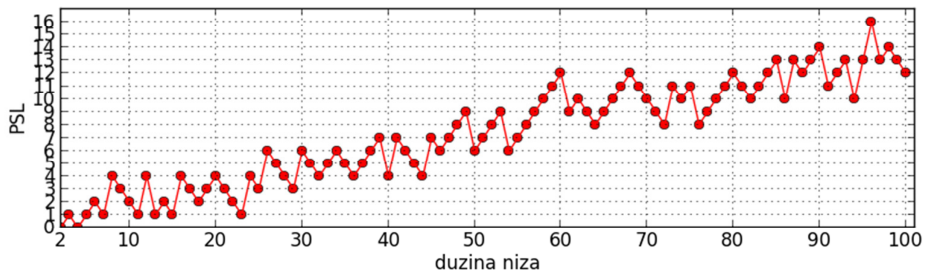
Slika A.23 Par binarnih nizova dužine 64 sa dobrim aperiodičnim autokorelacionim osobinama



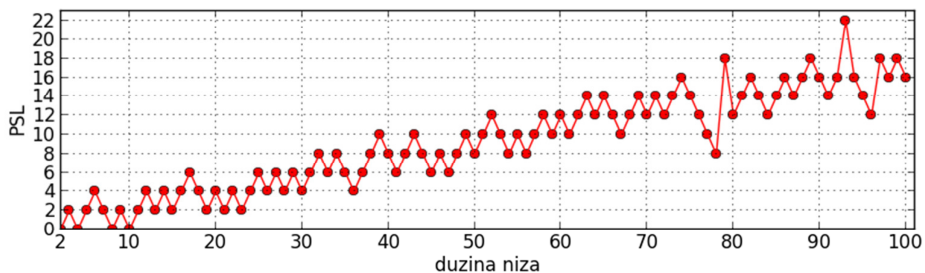
Slika A.24 PSL i MF za par sa slike A.23



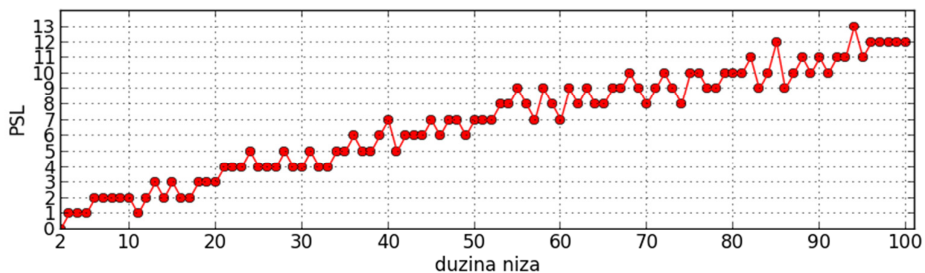
## Dodatak B



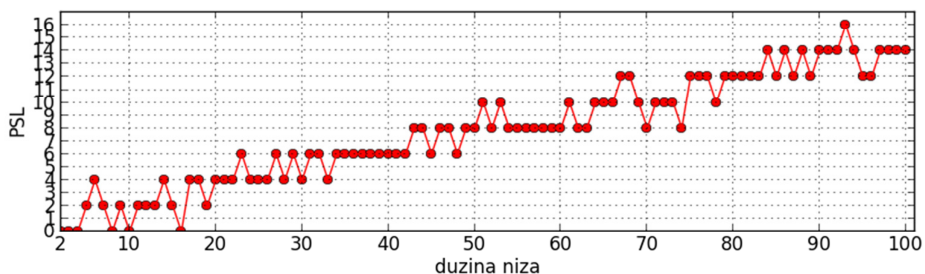
Slika B.1 Najbolji PSL za binarne nizove sa dobrim periodičnim autokorelacionim osobinama



Slika B.2 Najbolji PSL za binarne parove sa dobrim periodičnim autokorelacionim osobinama



Slika B.3 Najbolji PSL za binarne nizove sa dobrim aperiodičnim autokorelacionim osobinama



Slika B.4 Najbolji PSL za binarne parove sa dobrim aperiodičnim autokorelacionim osobinama

## Zaključak

Problem pronalaženja skupa komplementarnih nizova sa dobrim autokorelacionim osobinama je veoma aktuelan i razvijeni su različiti algoritmi za njegovo rešavanje. Algoritmi koji su predloženi u radu [1] (ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$ ) koji koriste metod naizmeničnih projekcija i bazirani su na aproksimaciji matrica korišćenjem uvijenog proizvoda se mogu koristiti za pronalaženje kako jednog niza sa dobrim (periodičnim ili aperiodičnim) autokorelacionim osobinama tako i za konstruisanje skupa komplementarnih nizova sa dobrim autokorelacionim (periodičnim ili aperiodičnim) osobinama. Predloženi su i izmenjeni ITROX algoritmi za rad sa ograničenim azbukama.

Iako je ove godine predložen algoritam CANARY [9] (u osnovi CAN algoritam [6]) koji je efikasniji od ITROX algoritama zbog korišćenja brze Furijeove transformacije, uvek postoji potreba za različitim algoritmima koji daju različita rešenja ovog problema.

U radu su opisani algoritmi ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  kao i izmene algoritama za nizove nad konačnom ili diskretnom azbukom i u slučaju pronalaženja samo jednog niza (korišćenjem metode stepenovanja umesto dekompozicije po sopstvenim vrednostima).

Dalje je dat kratak opis programskog jezika Python i razlozi za izbor tog jezika za implementaciju ITROX algoritama.

Detaljno je opisana implementacija ITROX algoritama posebno za

- implementaciju funkcije  $\mathcal{F}$  sa metodom šiftovanja kolona i njenog inverza  $\mathcal{F}^{-1}$  sa metodom šiftovanja redova,
- implementaciju naizmeničnih projekcija skupova  $\Gamma_{\mathcal{P}}(\mathbf{E})$  i  $\Lambda_{\mathcal{P}}(\mathbf{r})$  za periodične autokorelacije i  $\Gamma_{\mathcal{AP}}(\mathbf{E})$  i  $\Lambda_{\mathcal{AP}}(\mathbf{r})$  za aperiodične autokorelacije sa korišćenjem Toeplitz-ovih matrica,
- i konačno implementaciju ITROX- $\mathcal{P}$  i ITROX- $\mathcal{AP}$  sa izmenama za nizove nad binarnom azbukom  $\{-1,1\}$ .

Dalje su prikazani rezultati testiranja programa ITROX napisanog korišćenjem programskog jezika Python za periodične i aperiodične nizove i komplementarne parove nad realnom i binarnom azbukom. Možemo zaključiti da su dostignuti PSL i faktor kvaliteta uporedivi sa rezultatima objavljenim u radu [1] (str. 2188-2191) dok je vreme izvršavanja programa nešto brže u implementaciji opisanoj u ovom radu.



## Literatura

- [1] **M. Soltanalian and P. Stoica**, „*Computational Design of Sequences With Good Correlation Properties*“ IEEE Transactions on Signal Processing, vol. 60, no. 5, pp. 2180-2192, maj 2012.  
[Online] <http://www.it.uu.se/katalog/mojs0279/p3.pdf> [pristupljeno: septembar 2013]
- [2] **S. W. Golomb and G. Gong**, „*Signal Design for Good Correlation: For Wireless Communication, Cryptography, Radar*“, Cambridge, U.K.:Cambridge Univ. Press, 2005.
- [3] **M. G. Parker , C. Tellambura and K. G. Paterson**, „*Golay Complementary Sequences*“, in Wiley Encyclopedia of Telecommunications, John G. Proakis, ed., Wiley, 2003. [Online] Information Security Group/Kenny Paterson's Homepage <http://www.isg.rhul.ac.uk/~kp/golaysurvey.pdf> [pristupljeno: septembar 2013]
- [4] **S. Boyd and J. Dattorro**, „*Notes on alternating projections*“, Stanford University, Class EE392o: Optimization Projects, 2003. [Online] [http://www.stanford.edu/class/ee392o/alt\\_proj.pdf](http://www.stanford.edu/class/ee392o/alt_proj.pdf) [pristupljeno: septembar 2013]
- [5] **H. He**, „*Waveform design for active sensing systems - A computational approach*“, Ph.D. thesis, University Of Florida, 2011. [Online][http://etd.fcla.edu/UF/UFE0043119/he\\_h.pdf](http://etd.fcla.edu/UF/UFE0043119/he_h.pdf) [pristupljeno: oktobar 2013]
- [6] **P. Stoica, H. He, J. Li**, „*New algorithms for designing unimod-ular sequences with good correlation properties*“, IEEE Transactions on Signal Processing, vol. 57, no. 4, pp. 1415–1425, 1415-1425, april 2009. [Online] <http://plaza.ufl.edu/haohe/papers/CAN.pdf> [pristupljeno: septembar 2013]
- [7] **H. He, P. Stoica, J. Li**, „*Designing unimodular sequence sets with good correlations–Including an application to MIMO radar*“, IEEE Transactions on Signal Processing, vol. 57, no. 11, pp. 4391–4405, novembar 2009. [Online] <http://plaza.ufl.edu/haohe/papers/MIMO-CAN.pdf> [pristupljeno: septembar 2013]
- [8] **P. Stoica, H. He, J. Li**, „*On designing sequences with impulse-like periodic correlation*“, IEEE Signal Process. Lett., vol. 16, no. 8, pp. 703–706, avgust 2009.  
[Online]<http://plaza.ufl.edu/haohe/papers/PeCAN.pdf> [pristupljeno: septembar 2013]
- [9] **M. Soltanalian, M. M. Naghsh, P. Stoica** „*A fast algorithm for designing complementary sets of sequences*“, Signal Processing, vol. 93, pp. 2096-2102, 2013.  
[Online]<http://www.it.uu.se/katalog/mojs0279/p5.pdf> [pristupljeno: oktobar 2013]
- [10] **Beazley, D.** „*Python Essential Reference, Third Edition*“, Indianapolis : Sams Publishing, 2006.
- [11] **Wikipedia**, „*Guido van Rossum*“, [Online][http://en.wikipedia.org/wiki/Guido\\_van\\_Rossum](http://en.wikipedia.org/wiki/Guido_van_Rossum) [pristupljeno: septembar 2013]
- [12] **Python Programming Language - Official Website**, „*Application Domains*“, [Online]<http://www.python.org/about/apps/> [pristupljeno: septembar 2013]
- [13] **Wikipedia**, „*Power iteration*“, [Online] [http://en.wikipedia.org/wiki/Power\\_iteration](http://en.wikipedia.org/wiki/Power_iteration) [pristupljeno: oktobar 2013]
- [14] **Wikipedia**, „*Code Division Multiple Access*“, [Online][http://en.wikipedia.org/wiki/Code\\_division\\_multiple\\_access](http://en.wikipedia.org/wiki/Code_division_multiple_access)