

Elektrotehnički fakultet u Beogradu, 11.12.2019.

13E054MAS 2019

Metode analize elektrofizioloških signala

CIKLUS II: 5. Primer

*Vremensko – frekvencijska analiza*

*signala primenom spektrograma i*

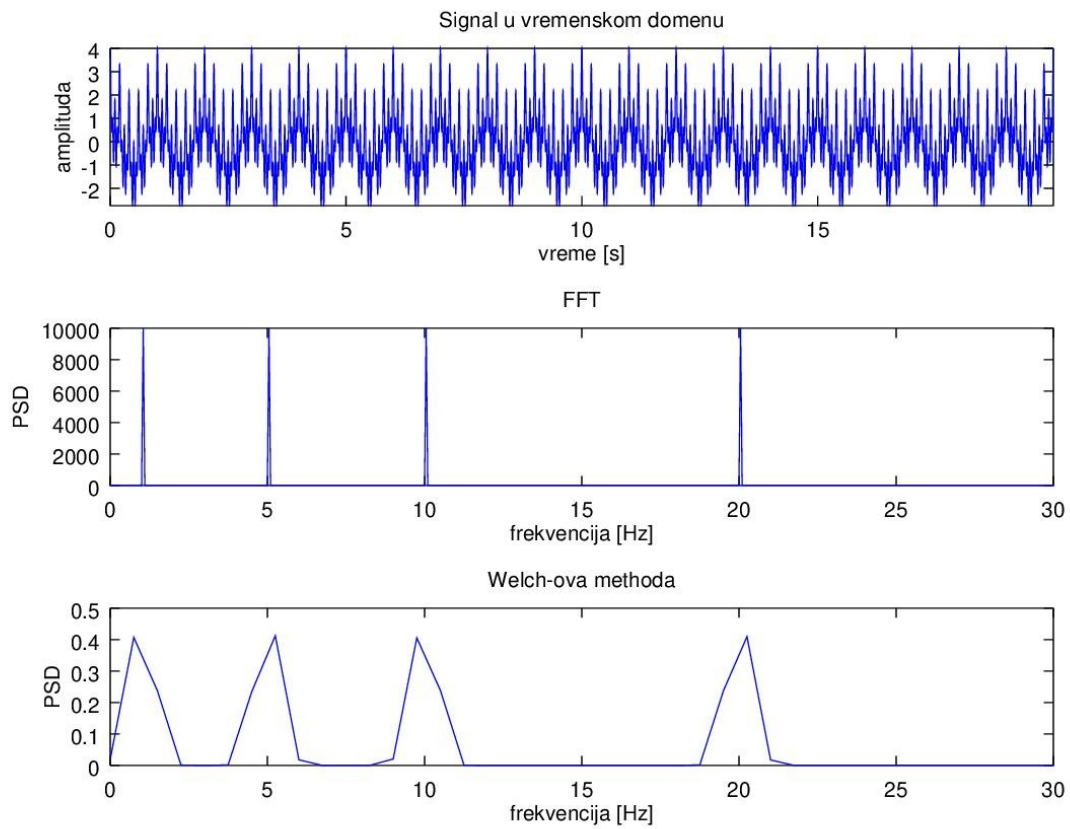
*wavelet transformacije*

Ema Pajić

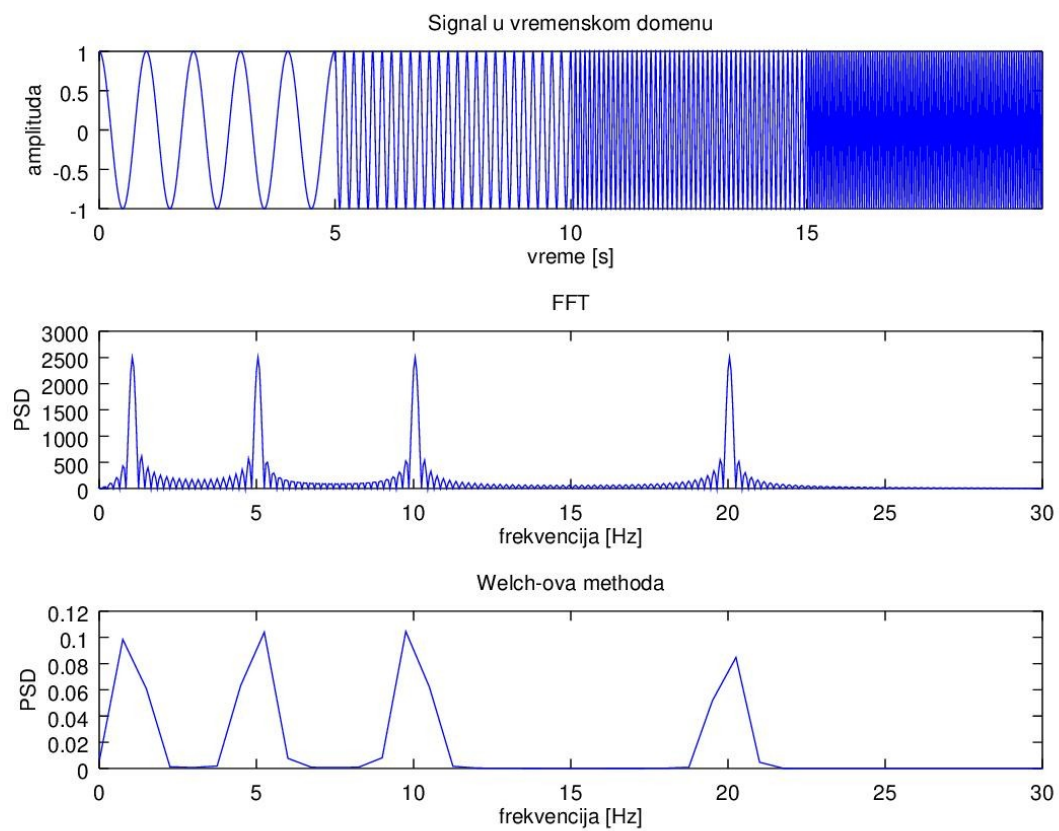
br. Indeksa: 2016/0017

1. **Zadatak** – Proučiti PSD\_welch.m funkciju i primentiti na spektar snage test signala (1 - 5)? Prikazati rezultat. Da li se primenom Welch metode bolje izdvajaju globalna (oblik spektra signala) ili lokalna (pojedinačna obeležja kao npr. pikovi) spektralna obeležja?

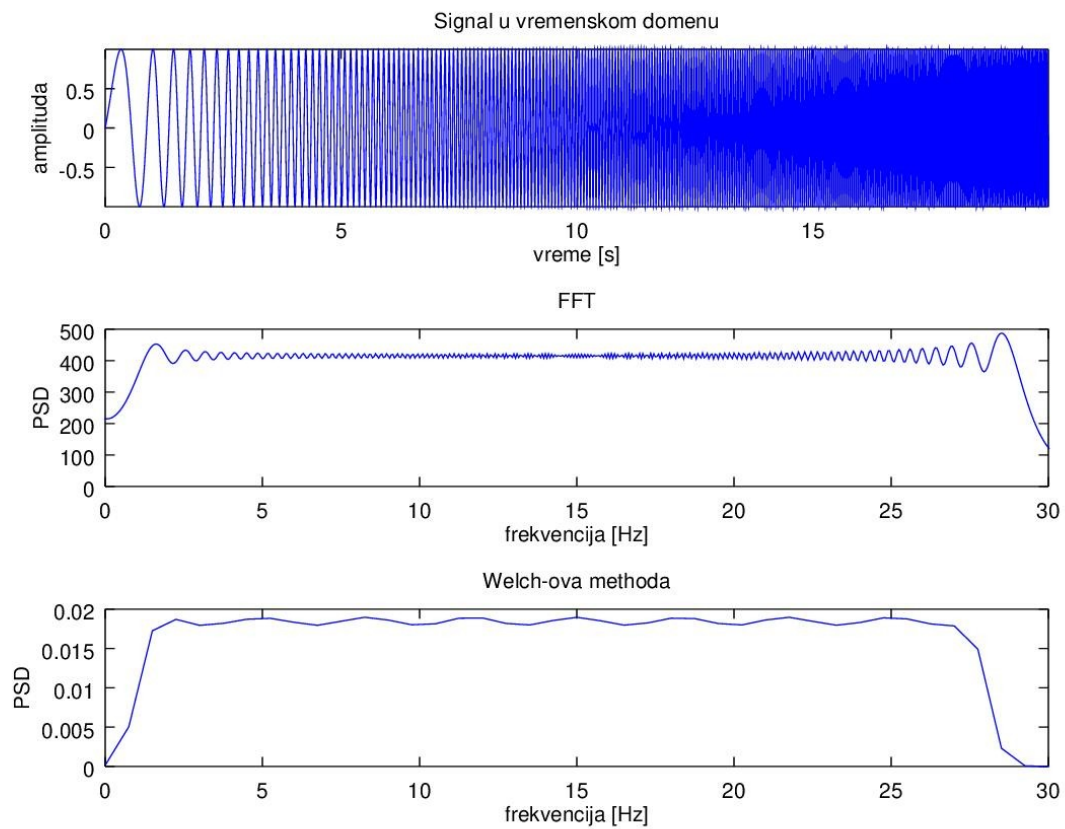
- Bolje se izdvajaju globalna spektralna obeležja. Na graficima se dobije obod onoga što bi bio spektrogram otprilike



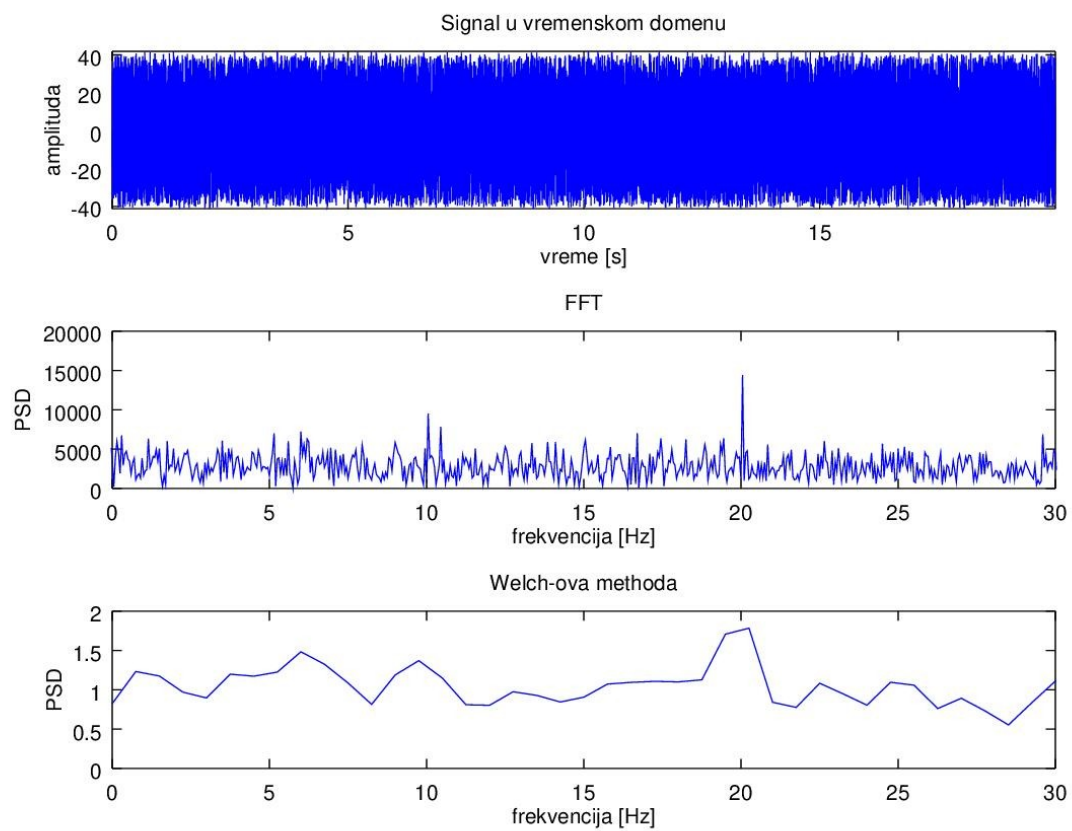
**Slika 1, Prikaz FFT i Welch-ove metode na stacionarnom signalu**



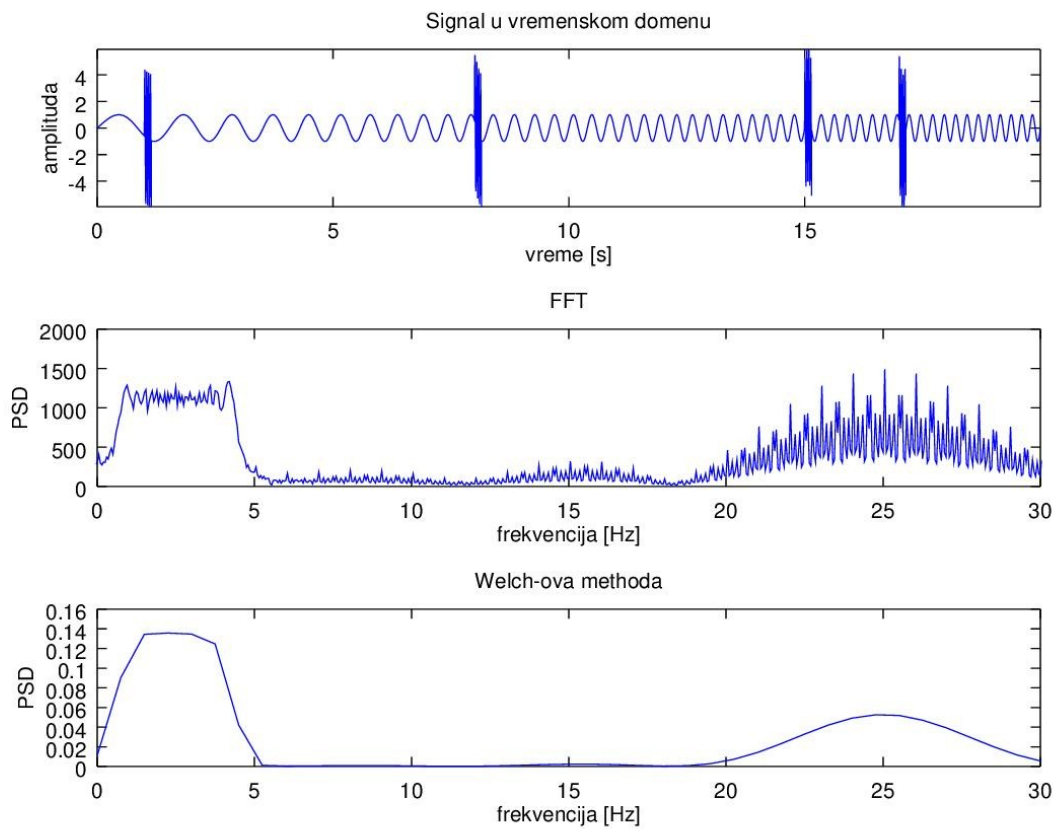
**Slika 2, Prikaz FFT i Welch-ove metode na nestacionarnom signalu**



**Slika 3, Prikaz FFT i Welch-ove metode na chirp signalu**



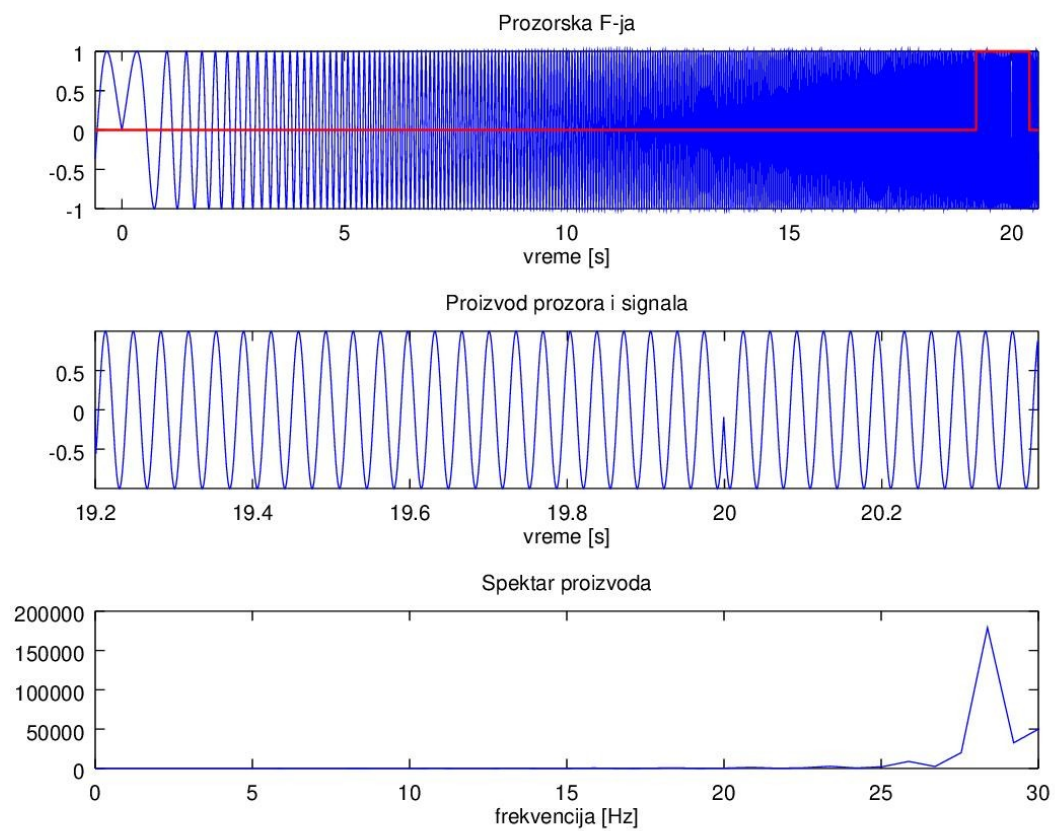
**Slika 4, Prikaz FFT i Welch-ove metode na sinusnom signalu sa šumom**



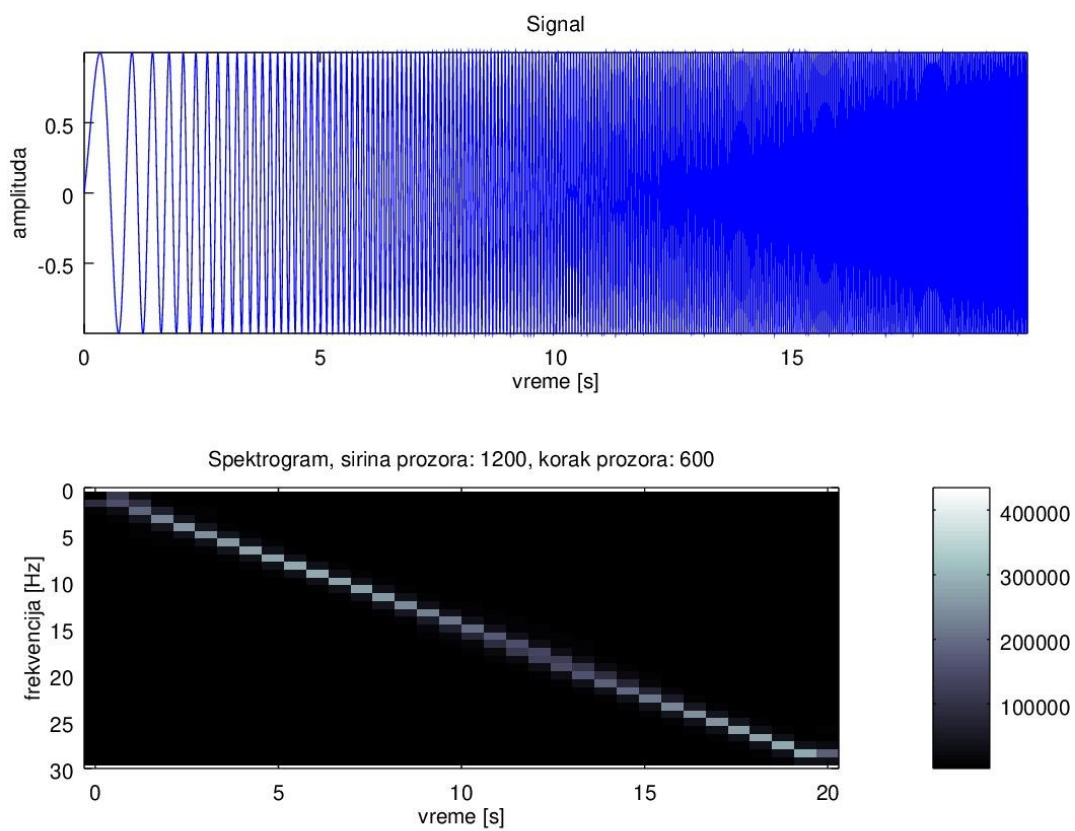
**Slika 5, Prikaz FFT i Welch-ove metode na sporooscilujućem signalu**

2. **Zadatak** – Proučiti STFT.m funkciju. Ispitati i objasniti uticaj izbora različitih širina i oblika prozorskih funkcija na spektrogram jednog test signala po izboru (1 - 5). Pored svih spektrograma dodati u prikazu i colorbar. Odgovore ilustrirati odgovarajućim prikazom spektrograma signala.

Posmatračemo STFT funkciju na chirp signalu. Što se tiče širine prozora, širina 1200 sa preklapanjem 600 daje dobre rezultate. Ako stavimo premalu širinu, vremenska rezolucija je dobra, ali je frekvencijska loša i spektrogram izgleda loše. Takođe, ukoliko stavimo preveliku širinu prozora, dolazi do toga da se previše vremenskih podataka izgubi, pa spektrogram opet izgleda loše. Što se tiče prozorske funkcije, na graficima su prikazani block window i normalized hanning. Sa grafika koji slede vidimo da je normalized hanning više smooth.

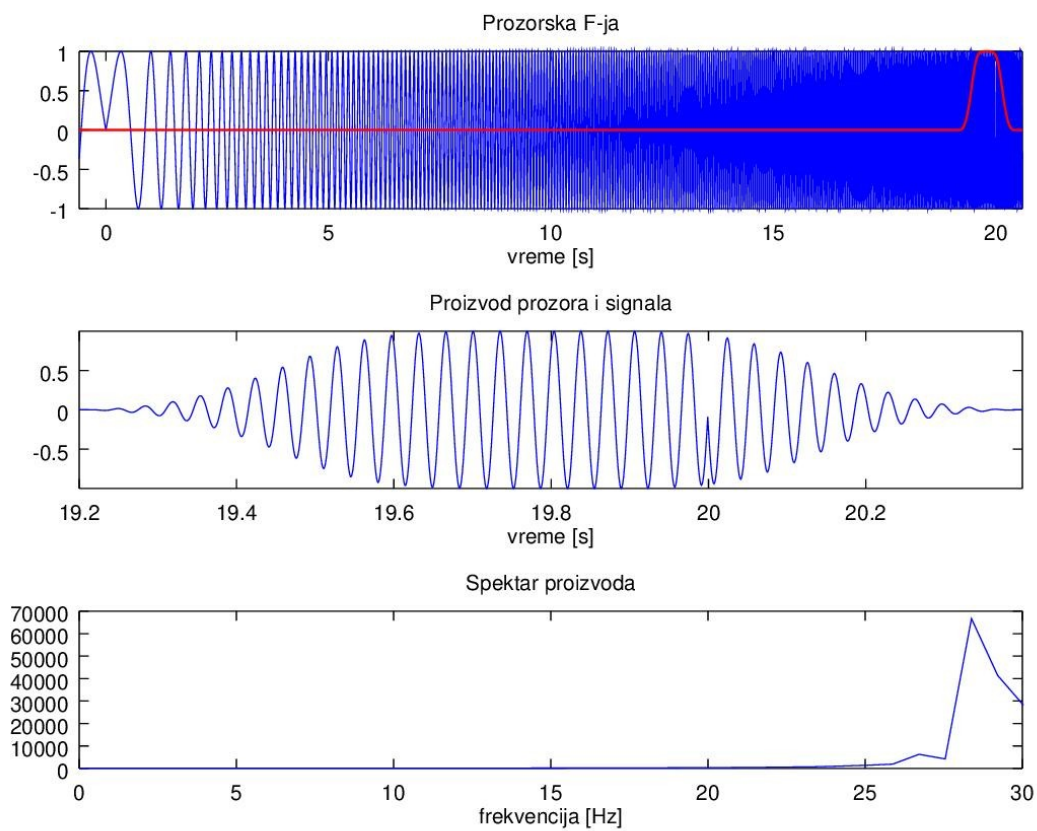


**Slika 6, Block window na chirp signalu sa širinom 1200 odbiraka i preklapanjem 600**

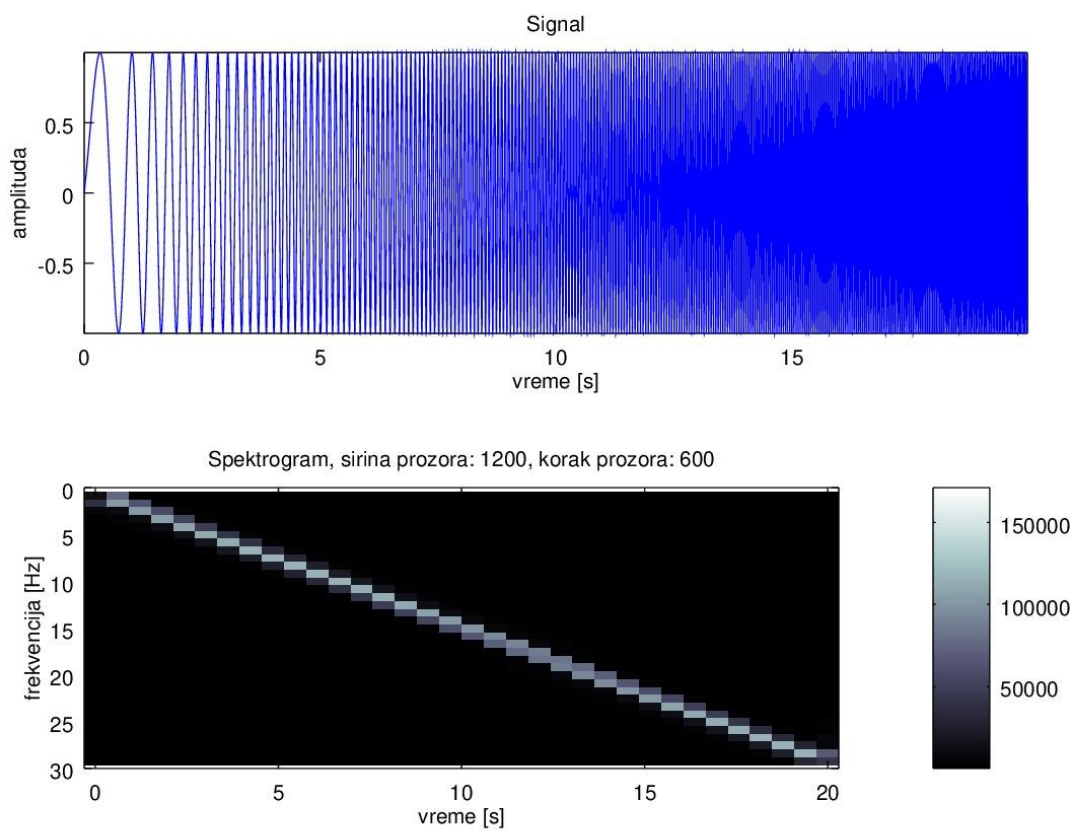


**Slika 7, Spektrogram chirp signala sa block windowom širine 1200 sa preklapanjem 600 odbiraka**

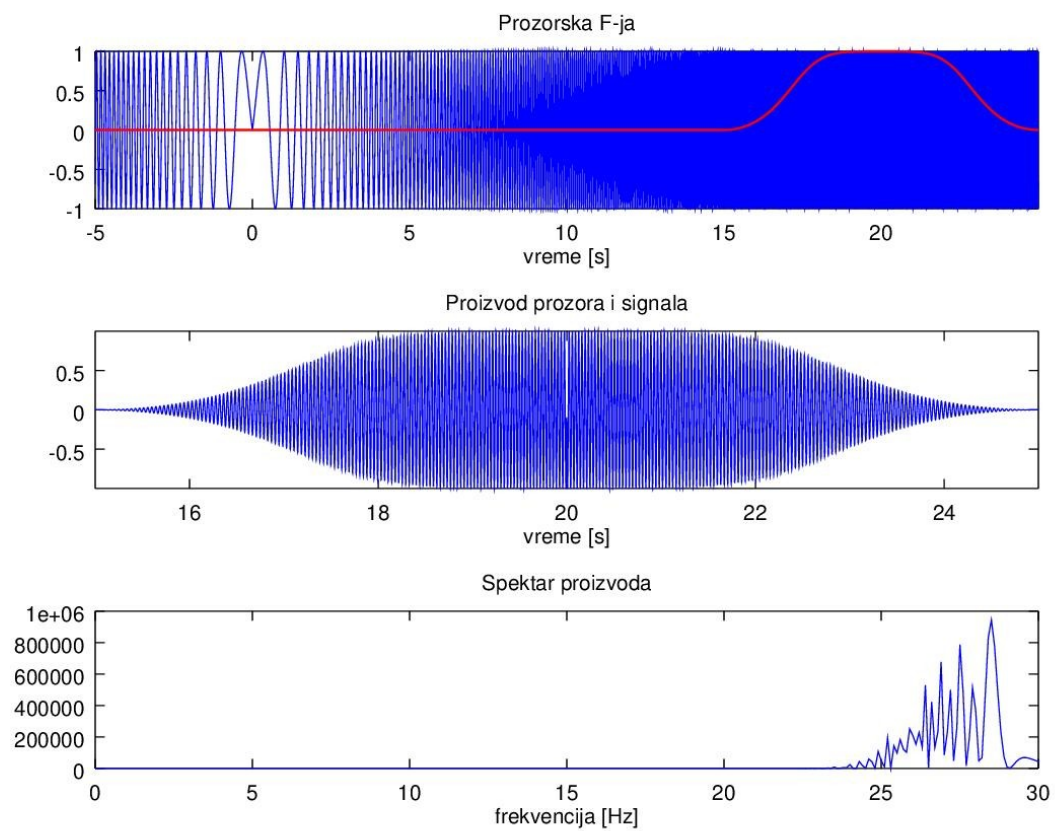




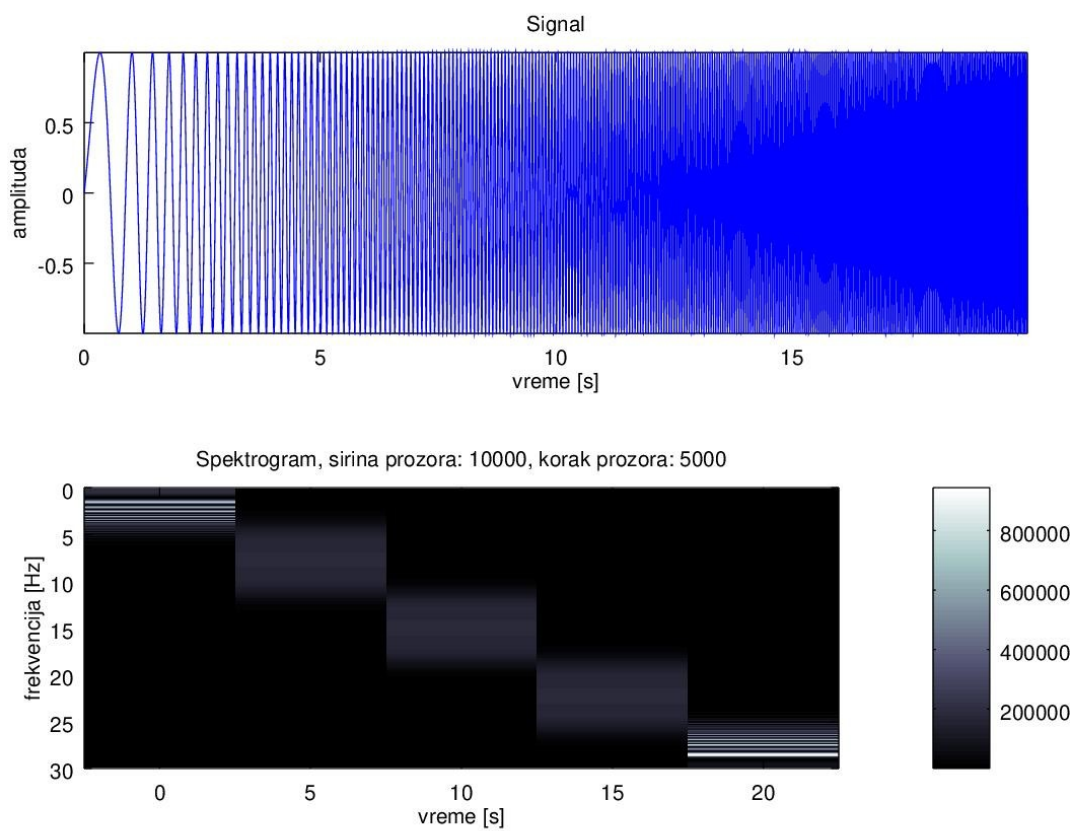
**Slika 8, Normalized hanning window na chirp signalu sa širinom 1200 odbiraka i preklapanjem 600**



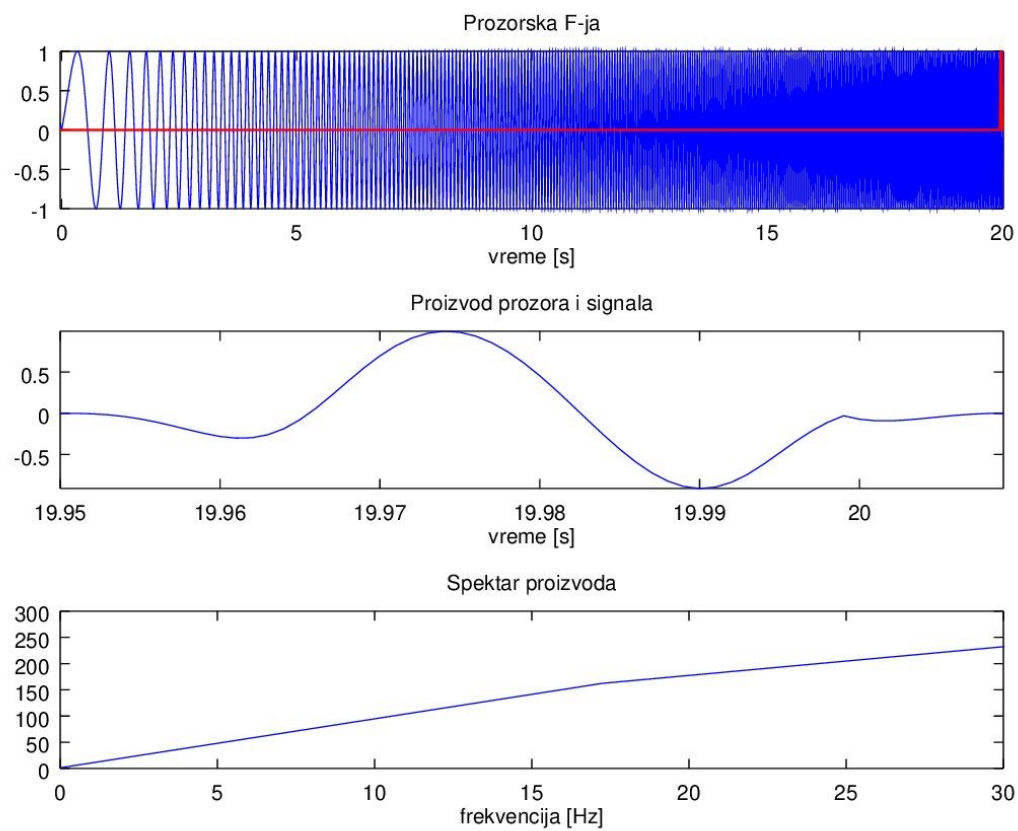
**Slika 9, Spektrogram chirp signala sa normalized chirp windowom širine 1200 sa preklapanjem 600 odbiraka**



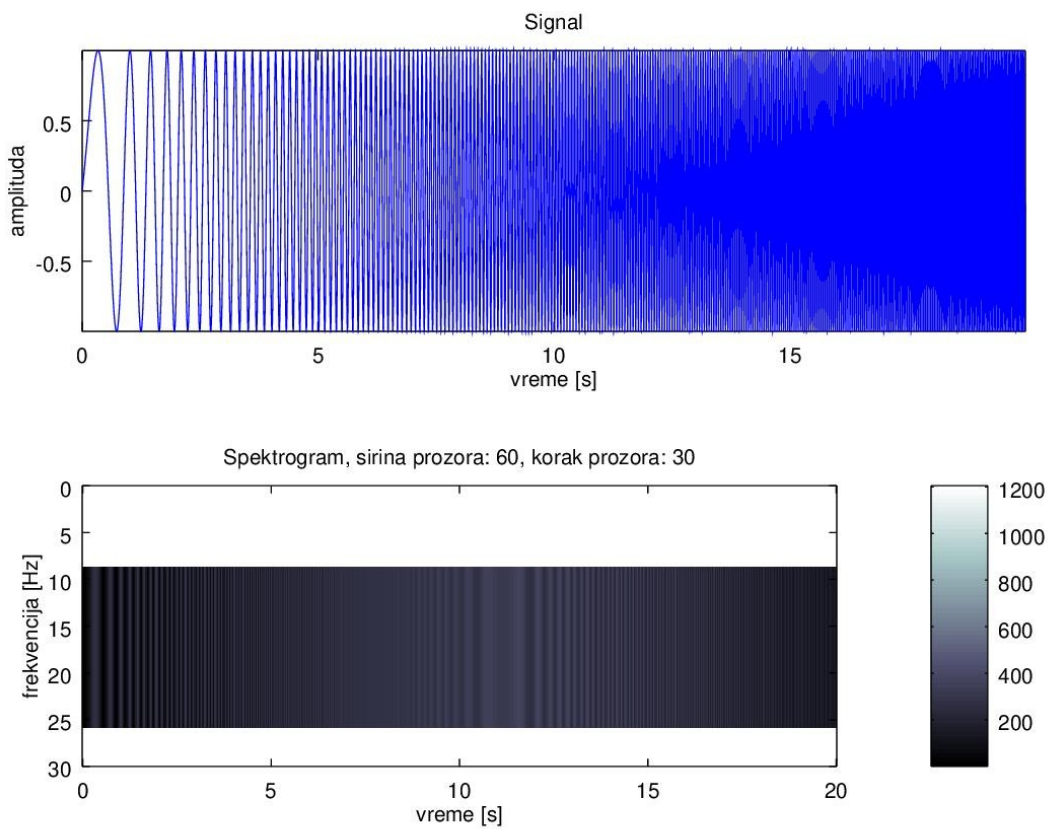
**Slika 10, Block window na chirp signalu sa širinom prevelikom širinom prozora**



**Slika 11, Spektrogram chirp signala sa normalized chirp windowom prevelike širine**



**Slika 12, Normalized haning window na chirp signalu sa premalom širinom prozora**



**Slika 13, Spektrogram chirp signala sa normalized chirp windowom premale širine**

Vidimo da najbrže radi CCA, a najsporije JADE.

Srednje vreme izvršavanja:

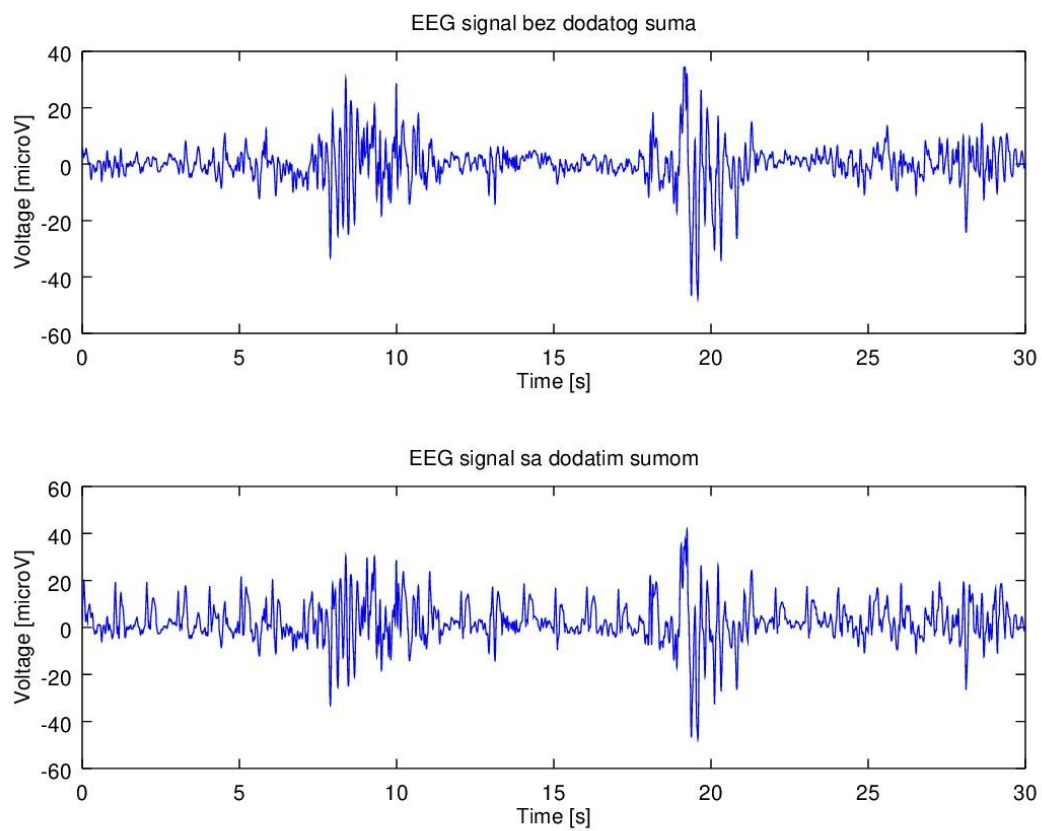
robustICA: 1.0229473, standardna devijacija 0.12015823938647

SOBI: 1.2823812, standardna devijacija 0.2851888079247

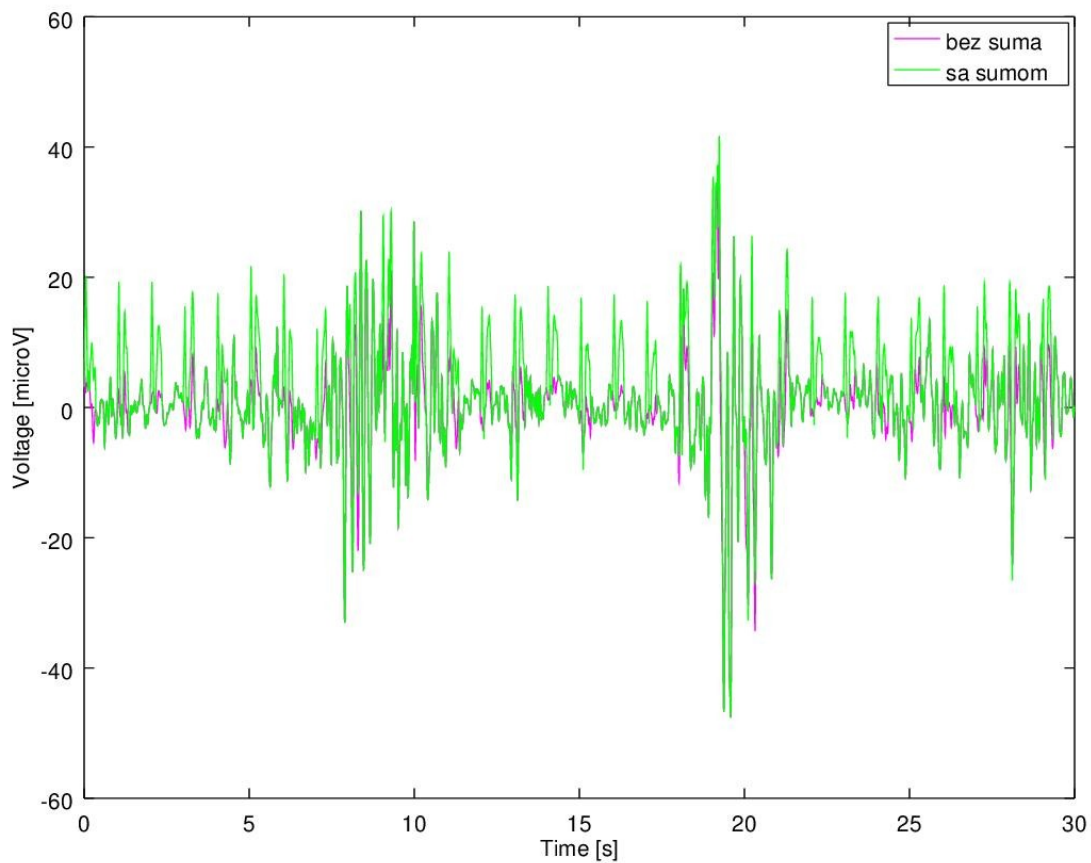
JADE: 2.668313, standardna devijacija 0.60446659792011

CCA: 0.01993449, standardna devijacija 0.00076698578206437

- 3. Zadatak** – Prikazati EEG signale sa jednog kanala bez šuma i sa dodatim šumom (na panelima pomodu subplot funkcije) i na istom grafiku (pomodu hold on komande).



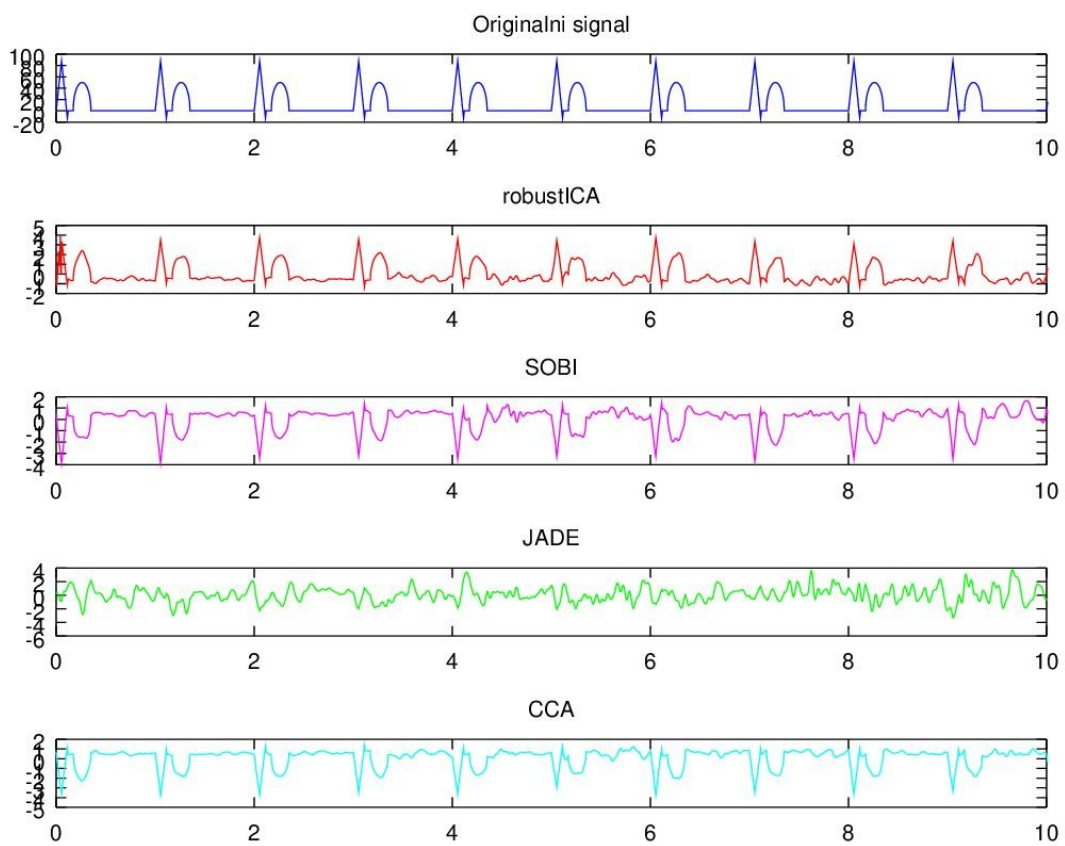
**Slika 1, EEG signal bez dodatog šuma I sa dodatim šumom, redom**



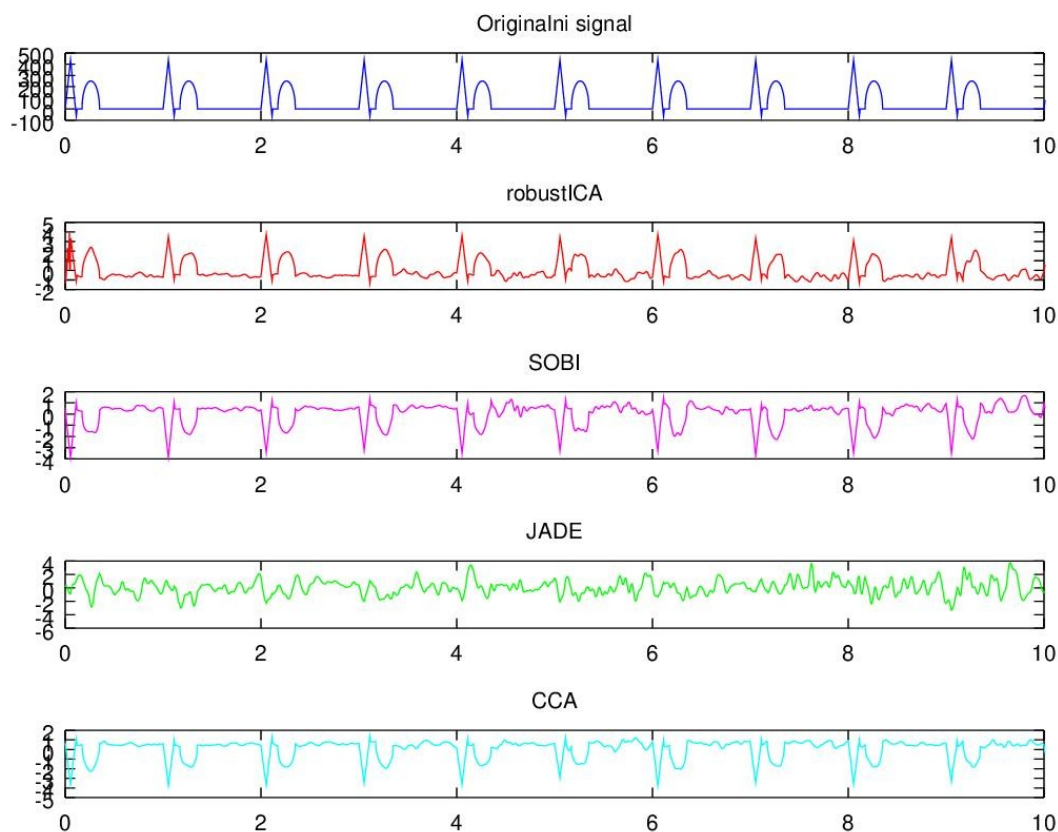
**Slika 2, EEG signal bez šuma i sa dodatim šumom prikazani na istom grafiku**

4. Uporediti izvornu komponentu šuma za svaki od algoritama sa sintetičkim EKG signalom i prikazati rezultate (tj. izvorne komponente) koje se dobijaju kao rezultat izvršavanja koda. Koji algoritam daje najbolje rezultate (utvrditi vizuelnom inspekcijom i na osnovu teorijskih pretpostavki)?





**Slika 3, Rezultati poređenih algoritama pri manjem šumu**



**Slika 4, Rezultati poređenih algoritama pri većem šumu**

Vizuelnom inspekcijom vidimo da su svi algoritmi osim JADE ok radili, na ovim slikama deluje da su možda robustICA ili CCA bili najbolji, što se i poklapa sa teorijskim očekivanjima.

RobustICA je radila dobro, međutim računski je kompleksna i poseduje druge nedostatke, pa se retko koristi u praksi. SOBI je radio ok, međutim za njega očekujemo da radi bolje pri malim odnosima signal/šum. JADE važi za dosta korišćen algoritam međutim, nije dobar za ovu primenu. CCA rešava problem tako što kreira zakašnjenu verziju originalne matrice podataka da bi se nezavisne komponente uslovile da budu maksimalno autokorelisane, a pošto ovde imamo EKG koji je poprilično periodičan, radi dobro.

5. Editovati kod u fajlu Primer\_ICA.m tako da se u okviru dodate šeste celine poredi rezultati poređenja algoritama korišćenjem RMS (eng. Root Mean Square) metode i korelacije. Komentarisati rezultate.

- Dodati je kod koji čuva date podatke u .mat fajlovima.
- Iz tabela koje slede možemo zaključiti: sa svih korelacija se jasno vidi da JADE radi dosta gore nego ostali algoritmi, a takođe i da su robustICA i CCA malo bolji od SOBI-ja. Istu stvar možemo zaključiti i iz rms-a, doduše manje uočljivo.
- `r_rms`

23.065	23.071	23.330	23.055
118.674	118.679	118.932	118.665

- AbsCor1:

0.97116	0.95606	0.70682	0.97593
---------	---------	---------	---------

- AbsCor2:

0.97116	0.95606	0.70682	0.97593
---------	---------	---------	---------

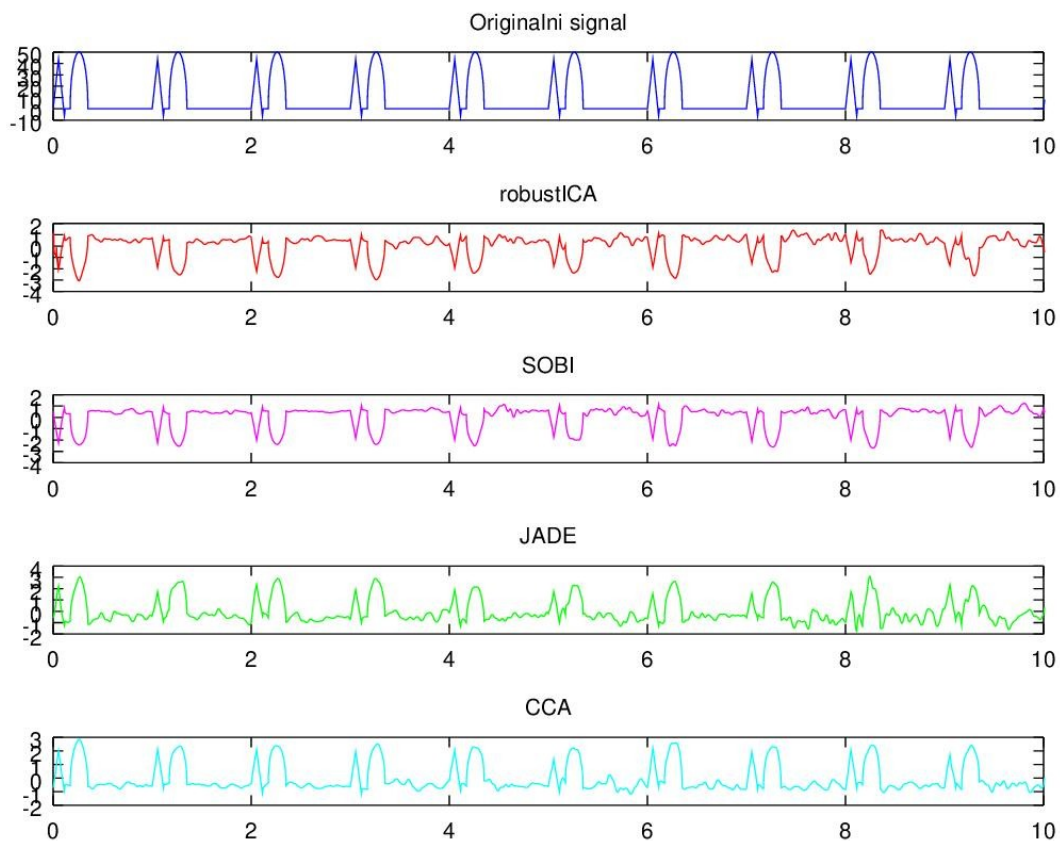
- AbsCorSpearman1:

0.78329	0.77298	0.62988	0.78516
---------	---------	---------	---------

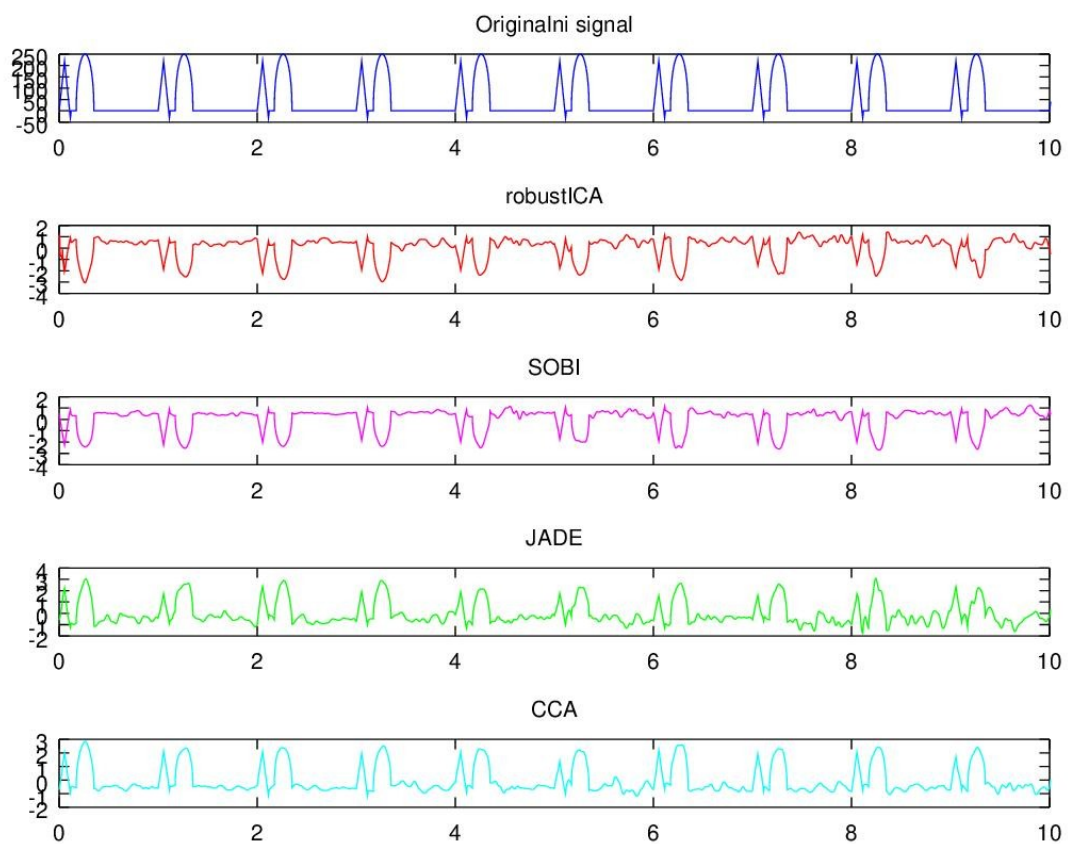
- AbsCorSpearman2:

0.78329	0.77298	0.62988	0.78516
---------	---------	---------	---------

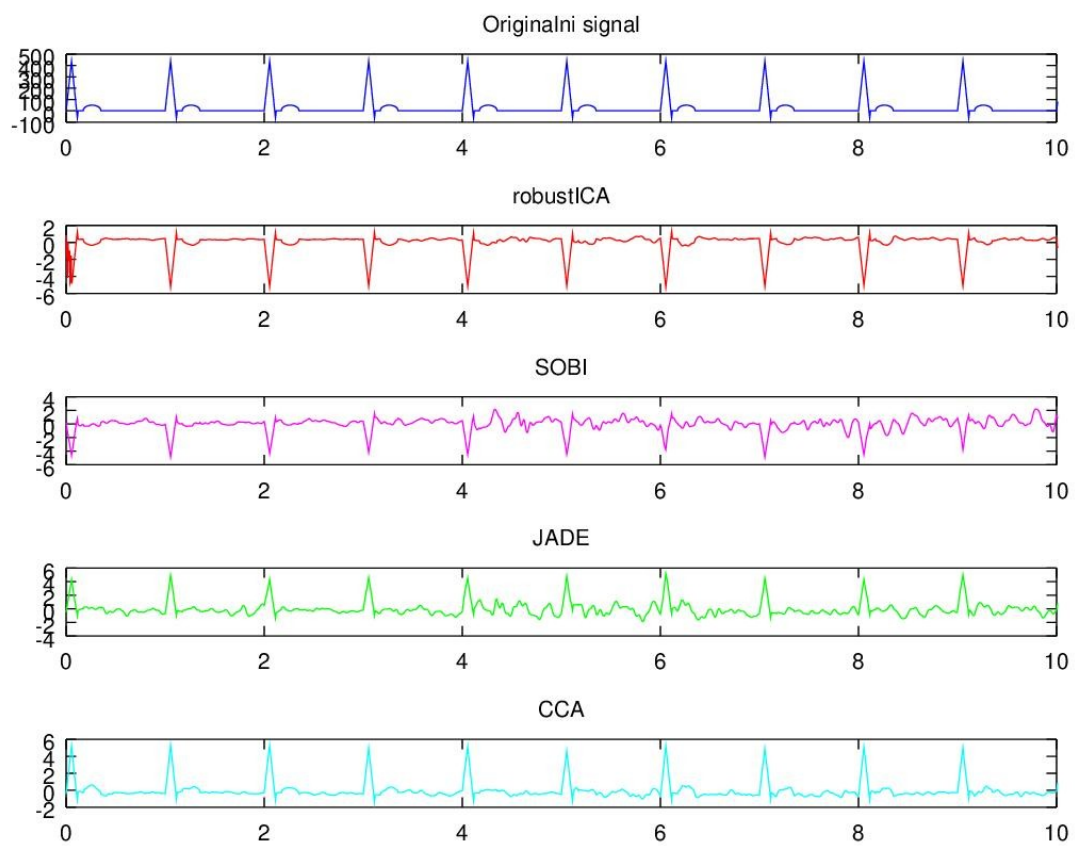
6. Promeniti amplitudu šuma (amplituda EKG signala) i prikazati rezultate algoritama. Za koje amplitude šuma su algoritmi efikasniji (manje ili veće)? Promeniti dužinu sekvence signala koji se učitava i prikazati nezavisne komponente na slici. Da li postoje razlike u rezultatima?



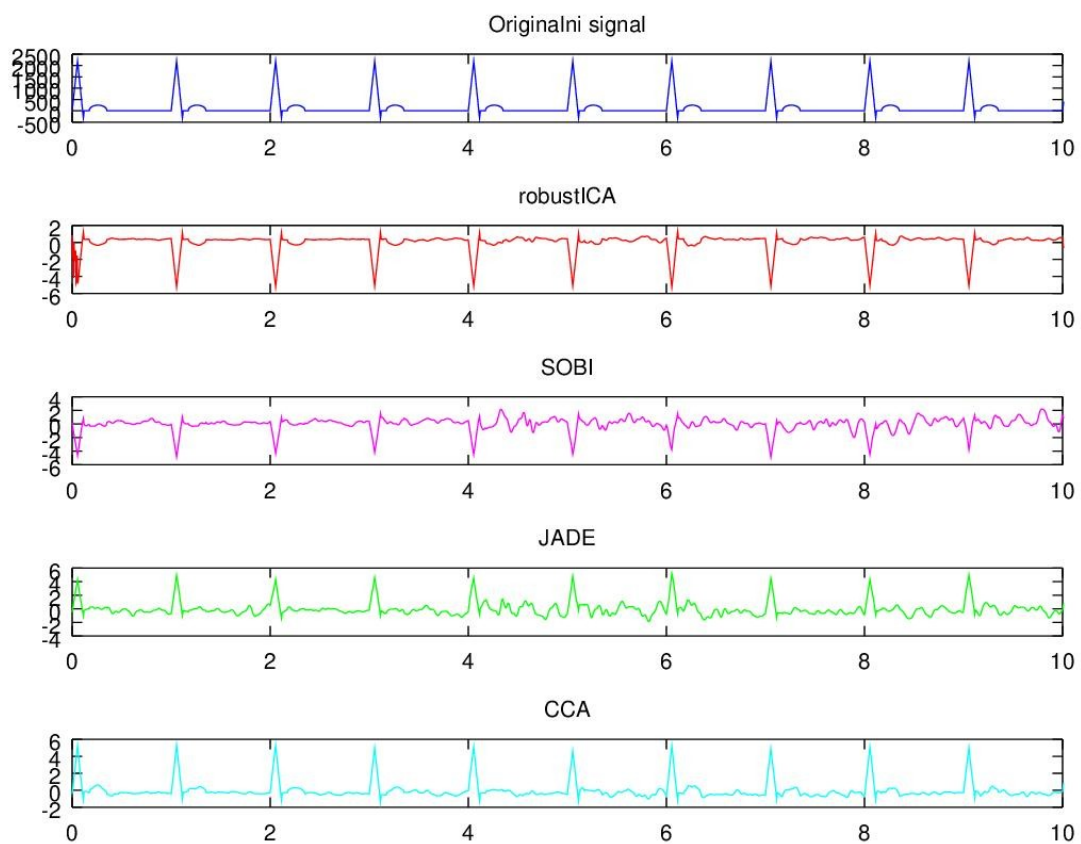
**Slika 5, Rezultati poređenih algoritama pri manjem šumu, sa smanjenom amplitudom šuma**



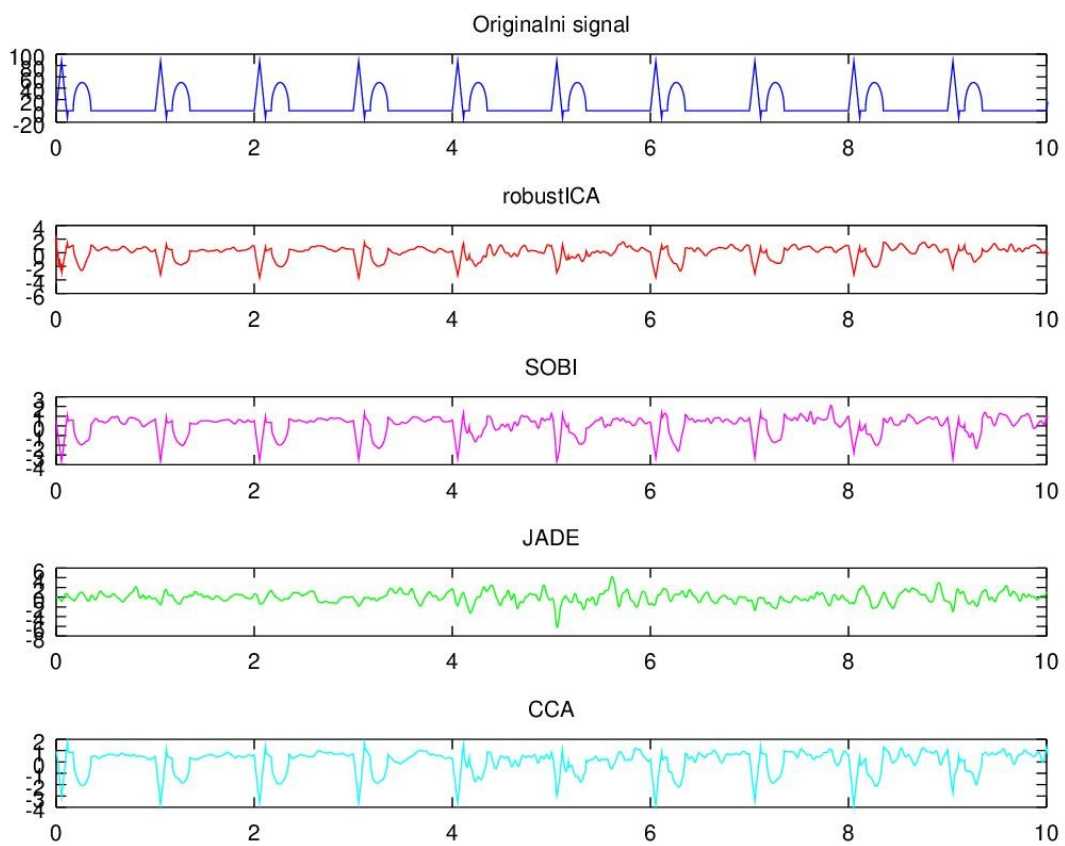
**Slika 6, Rezultati poređenih algoritama pri većem šumu, sa smanjenom amplitudom QRS kompleksa**



**Slika 7, Rezultati poređenih algoritama pri manjem šumu, sa povećanom amplitudom QRS kompleksa**

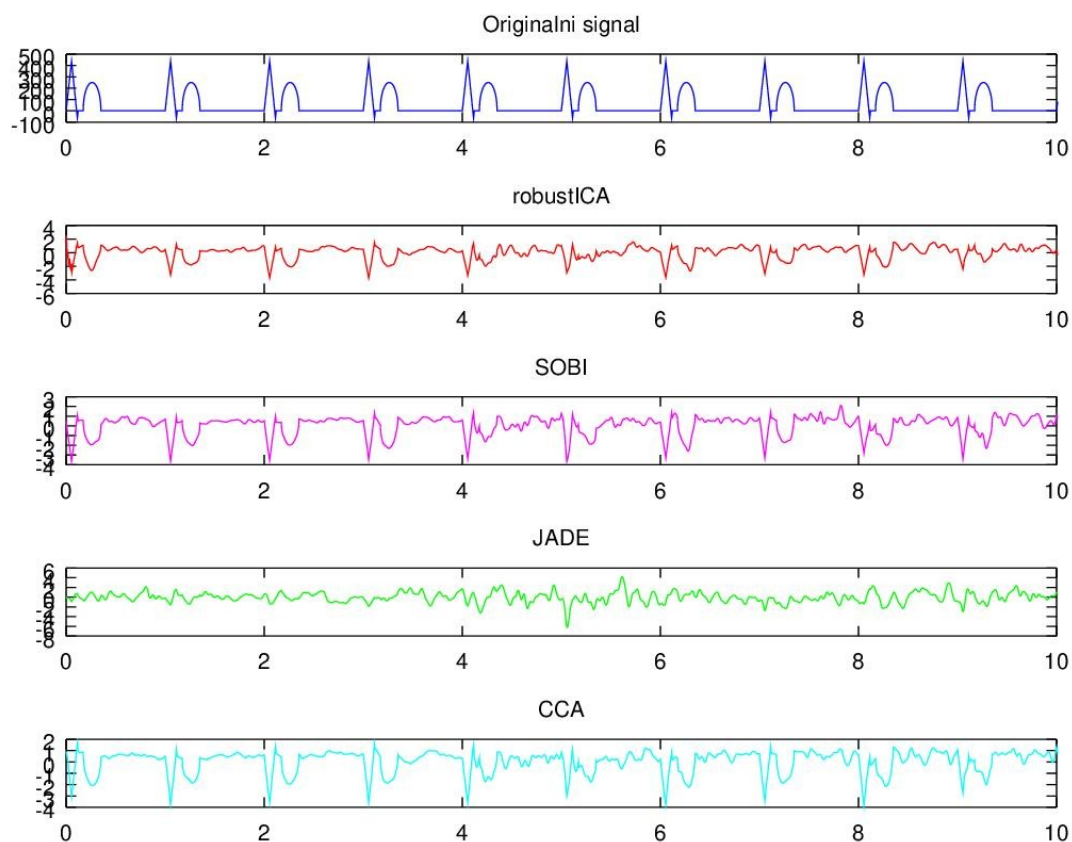


**Slika 8, Rezultati poređenih algoritama pri većem šumu, sa povećanom amplitudom QRS kompleksa**



**Slika 9, Rezultati poređenih algoritama pri manjem šumu, sa skraćenim trajanjem signala**





**Slika 10, Rezultati poređenih algoritama pri manjem šumu, sa skraćenim trajanjem signala**

Sa ovih grafika možemo zaključiti:

- Pri smanjenoj amplitudi QRS kompleksa:
  - SOBI radi najbolje, što se slaže sa teorijom, pošto znamo da je mnogo bolji pri manjem šumu. RobustICA idalje radi ok, ali rezultat nije popravljen smanjenjem šuma. JADE idalje radi loše. CCA radi dobro, a rezultati jesu popravljeni smanjenjem šuma.
- Pri povećanoj amplitudi QRS kompleksa:
  - SOBI radi baš loše. RobustICA i CCA idalje rade dobro, a ovaj put i JADE radi dobro.
- Pri povećanju amplitude samog šuma:
  - Ne menja se ništa. Amplituda šuma, dokle god je oblik šuma isti, ne utiče.
- Pri smanjenom trajanju signala:
  - Skraćenje trajanja signala poprilično utiče na sve metode, JADE ne funkcioniše uopšte, a sa ostalih se idalje može prepoznati EKG signal, doduše veoma zašumljen. Deluje mi da je SOBI ipak najviše prepoznatljiv.

7. Pokrenuti program `Zadatak_Cocktail_problem.m`. Objasniti celine u kodu. Odslušati dobijene rezultate nakon primene ICA algoritama.

- 1. celina: Učitavanje miksovanih signala iz WAV formata u matricu
- 2. celina: Primena algoritama (robustICA, SOBI, JADE, CCA)
- 3. celina: Čuvanje rezultata u nove WAV fajlove
- Nakon odslušanih fajlova, vidimo da su komponente dobro izdvojene.



8. Navesti barem jedno ograničenje ICA teorije koja se može uočiti na rezultatima u ovom zadatku.

- Moramo da imamo najmanje jednak broj ulaznih signala (mikrofona) kao broj nezavisnih komponenti
- Ne može se odrediti poredak izvornih komponenti
- U Primer\_ICA.m zadatku, vidi se i to da se ne može egzaktno odrediti amplituda i znak izvornih promenljivih.

■ NAPOMENE:

- ◆ PDS\_welch - U funkciji pwelch drugačije se prosleđuje overlap. U Matlabu se daje broj sample-ova koji treba da se preklapa, a u Octave-u procenat (0-1) windowa. Znači, treba promeniti  $noverlap = \text{round}(\text{window}/2)$ ; na  $noverlap = 0.5$ ; (Što tehnički nije neophodno jer je u toj funkciji u Octaveu podrazumevani overlap 0.5).
- ◆ STFT – funkcija mora da se zove isto kao ime fajla da bi se pozvala iz drugog fajla.
- ◆ U kodu za Zadatak\_Cocktail\_problem.m postoji typo, treba prepraviti na acsorbirao da bi se kod pokrenuo.
- ◆ U Octave-u nije još implementiran table, tako da nisam imala tu opciju, već sam čuvala u .mat fajlovima