Katedra za Signale i sisteme
Elektrotehnički fakultet
Univerzitet u Beogradu

13E054OPG - Obrada i prepoznavanje govora

Domaći zadatak

Aleksa Janjić 2019/0021

Sadržaj

1 Zadatak 1

- Korišćenjem komercijalnog mikrofona u programskom okruženju **MATLAB**, snimiti govornu sekvencu u dužini od 20-ak sekundi. Sekvencu snimiti sa frekvencijom odabiranja 8 ili 10kHz i ona treba da se sastoji od desetak jasno segmentiranih reči.
- Korišćenjem kratkovremenske energije i kratkovremenske brzine prolaska kroz nulu izvršiti određivanje početka i kraja pojedinih reči. Dobijeni rezultat prikazati grafički. Preslušati segmentirane delove zvučne sekvence i komentarisati dobijeni rezultat. (Po želji se ovaj postupak može ponoviti primenom *Teager* energije).
- Snimiti novu sekvencu od par reči (bogatih samoglasnicima, recimo onomatopeja...) i na osnovu tako snimljene sekvence proceniti *pitch* periodu sopstvenog glasa. Koristiti dve različite metode pa uporediti i komentarisati dobijene rezultate.

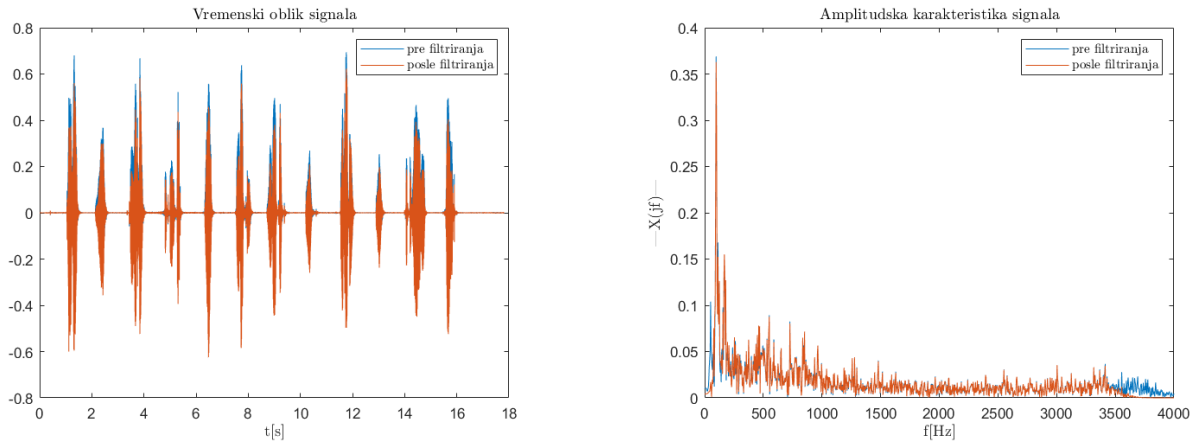
1.1 Kratkovremenska energija i kratkovremenska brzina prolaska kroz nulu

Snimljena je govorna sekvenca od 12 reči sa pauzama između njih radi lakše segmentacije. Korišćena je frekvencija odabiranja od 8kHz. Pre dalje analize, najpre je signal propušten kroz *Butterworth*-ov filtar propusnik učestanosti od 60 do 3500 Hz kako bi se eliminisao šum, a pritom informacija sadržana u govornom signalu ostala nepromenjena. Segmentacija je vršena pomoću kratkovremenske energije i kratkovremenske brzine prolaska kroz nulu. Pošto je govorni signal nestacionaran, vrši se prvo prozorovanje signala i u okviru prozora možemo smatrati da su te sekvence signala stacionarne, stoga možemo računati kratkovremensku energiju (engl. *Short-Time Energy*) i kratkovremensku brzinu prolaska kroz nulu (engl. *Short-Time Zero Crossing Rate*) po sledećim relacijama:

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2$$

$$Z_n = \frac{1}{2L} \sum_{m=-\infty}^{\infty} [sgn(x(m)) - sgn(x(m-1))]\, w(n-m),$$

pri čemu je x govorni signal, w pravougaona prozorska funkcija dužine L. Za dužinu prozorske funkcije uzeto je 20 ms, odnosno, 160 odabiraka. Na Slici 1 prikazani su vremenski oblik signala i njegov amplitudski spektar pre i posle filtriranja.

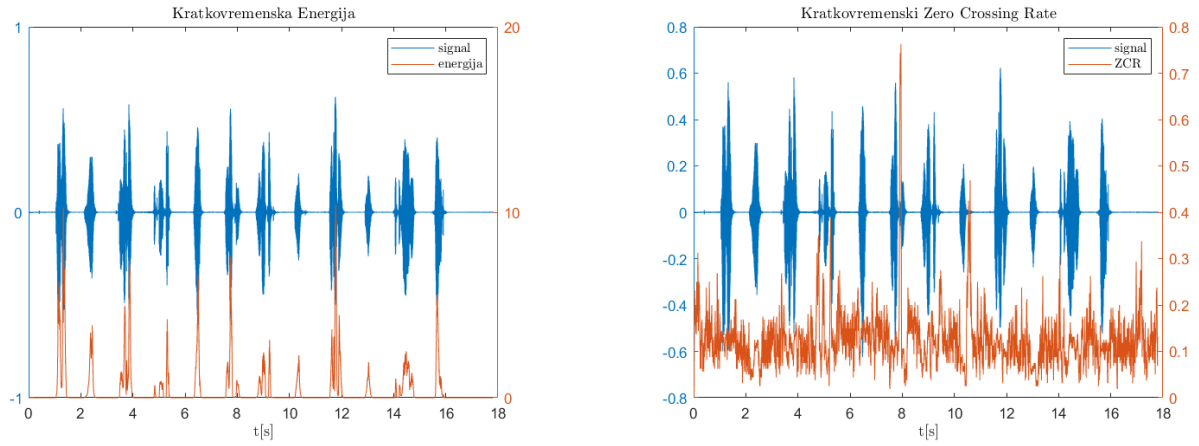


(a) Vremenski oblik signala (b) Amplitudska karakteristika signala

Slika 1: Vremenski oblik signala i njegova amplitudska karakteristika pre i posle filtriranja

Uočavamo da se u vremenskom obliku signala pojavljuju 12 podoblasti koje predstavljaju izgovorene reči koje su jasno razdvojene, te ne bi trebalo da bude većih problema prilikom segmentacije (osim možda odsecanja krajeva reči koje se završavaju bezvučnim glasovima). Takođe, primećuje se da efekat filtriranja nije jasno vidljiv u vremenskom obliku signala, ali zato se na amplitudskoj karakteristici vidi da su odsečene frekvencijske komponente na izuzetno niskim i izuzetno visokim učestanostima za koje smo gotovo sigurni da ne potiču od govornog signala.

Na sledećoj Slici 2 prikazane su kratkovremenska energija i kratkovremenska brzina prolaska kroz nulu filtriranog signala.

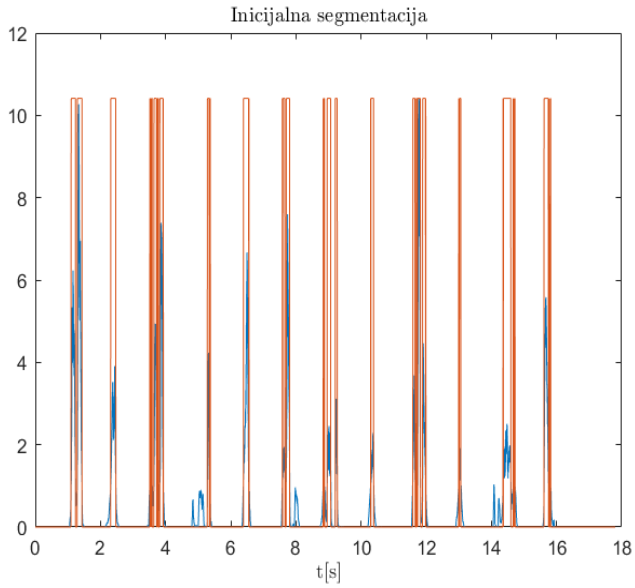


(a) Kratkovremenska energija signala

(b) Kratkovremenska brzina prolaska kroz nulu

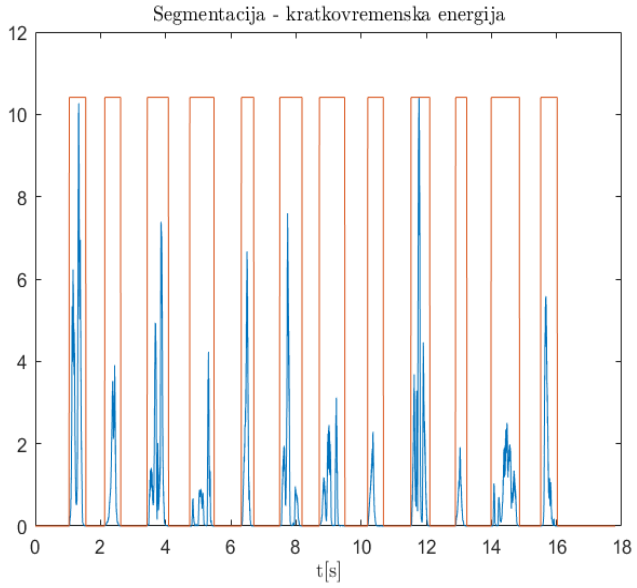
Slika 2: Kratkovremenska energija i kratkovremenska brzina prolaska kroz nulu signala

Možemo primetiti da je energija velika tamo gde ima izgovorenih reči, a mala gde je tišina. Za brzinu prolaska kroz nulu važi suprotno, odnosno, ona je velika kada je tišina nego kada postoji govor. Koristeći ove dve osobine moguće je segmentirati reči iz govorne sekvence koristeći sledeći postupak. Prvo odredimo veliku granicu ITU (10% maksimalne energije signala). Energija se prvo upoređuje sa ovim pragom i tamo gde je energija veća od praga, sigurni smo da postoji govor, odnosno, reč.



Slika 3: Inicijalna segmentacija reči - kratkovremenska energija

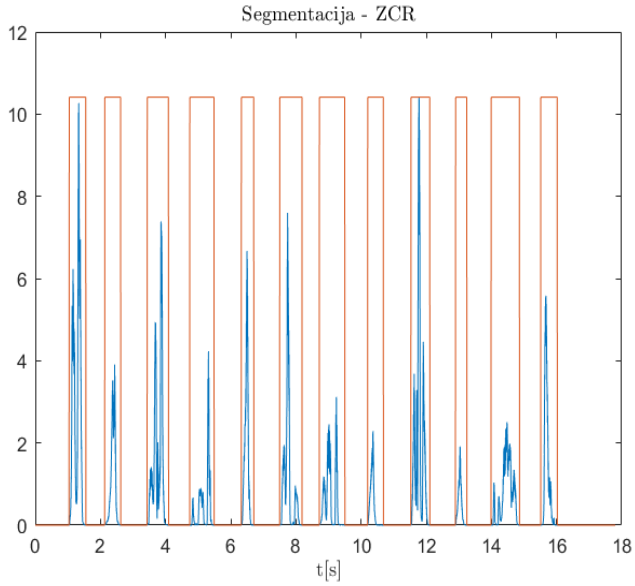
Na slici 3 vidimo da postoji više segmenata nego što ima reči, odnosno, zanemareni su neki delovi reči koji nisu toliko zvučni i reč se na taj način deli na više segmenata. Zbog ovoga se uvodi takozvana niska granica ITL (takođe se odreduje kao procenat maksimalne energije, konkretno ovde je uzeto 0.005%). Sada za svaku reč krenemo od njenog početka i idemo ulevo sve dok energija ne padne ispod ITL i ta tačka predstavlja novi početak reči. Slično i za kraj reči, ide se udesno sve dok je energija veća od ITL i to nam predstavlja novi završetak reči. Finalna segmentacija reči uz pomoć kratkovremenske energije signala je predstavljena na sledećoj slici 4.



Slika 4: Finalna segmentacija reči - kratkovremenska energija

Uočavamo sada da postoji 12 segmenata govornog signala, tačno onoliko koliko i postoji reči u sekvenci i svaka reč je jasno razumljiva nakon preslušavanja. Ipak, u cilju poboljšanja pomeranja početka i kraja reči, primenićemo i segmentaciju reči baziranu na kratkovremenskoj brzini prolaska kroz nulu. Algoritam za ovu popravku kaže da je neophodno krenuti od granice reči i gledamo 25 prozorskih funkcija ulevo, odnosno, udesno. Ako je u ovom opsegu kratkovremenska brzina prolaska kroz nulu makar tri puta presekla prag IZCT (u našem slučaju je on određen eksperimentalno posmatrajući histogram ZCR) početak, odnosno, kraj reči se postavlja na poslednje mesto gde je brzina prolaska kroz nulu bila veće od IZCT. Nakon ove izmene, segmentacija na bazi kratkovremenske brzine prolaska kroz nulu prikazana je na sledećoj slici 5.

Na slici 5, kao ni preslušavanjem reči, ne uočavaju se razlike u odnosu na kratkovremensku energiju koja svakako daje zadovoljavajuće rezultate.



Slika 5: Finalna segmentacija reči - kratkovremenska brzina prolaska kroz nulu

1.2 Estimacija *pitch* periode

Pitch frekvencija je frekvencija otvaranja i zatvaranja glasnih žica prilikom strujanja vazduha kroz njih. Ona je jedinstvena za svakog govornika, i razlikuje se kod muškaraca (80-160Hz), žena (130-280Hz), dece (200-400Hz) i novorođenčadi (oko 600Hz). Prvi metod za procenu *pitch* frekvencije koji koristimo je **metod paralelnog procesiranja**. On se sastoji od sledećih komponenti:

- filtra - signal filtriramo filtrom propusnika opsega, čije granice predstavljaju interval u kojem očekujemo *pitch* periodu, a tako izbacujemo i sve ostale neželjene frekvencije,
- generator impulsa - generisemo šest različitih impulsnih sekvenci,
- *pitch* period estimator - za svaku dobijenu sekvencu, estimator procenjuje *pitch* periodu, tako da ona ne bude manja od očekivane, što obezbeđujemo definisanje *blanking* perioda τ i da ne bude duža od očekivane, što obezbeđujemo definisanjem parametra λ,
- finalni *pitch* period estimator - finalni estimator prihvata procene svakog **PPE** bloka i daje konačnu procenu. Kako neki od blokova mogu dati vrlo pogrešne procene, potrebno je da **FPPE** bude veoma robustan. Zato on donosi procenu korišćenjem *median* funkcije. Kako koristimo medijanu, bilo bi dobro da joj šaljemo neparan broj odbiraka, pa zato zajedno sa šest estimiranih perioda, šaljemo i prethodnu finalnu procenu *pitch* periode.

Snimljena je sekvenca od 3 reči, a pre izvršavanja ovog algoritma izvršena je predobrada govornog signala, tako što su grubo eliminisani bezvučni segmenti, a zatim se tako izmenjena sekvenca propušta kroz filtar i dalje delove algoritma. Procenjena *pitch* perioda iznosi 106.67Hz i ona je u skladu sa očekivanom vrednošću, odnosno, nalazi se u opsegu frekvencija koje odgovaraju odraslom muškarcu.

Drugi metod za estimaciju *pitch* periode je **metod procene autokorelacione funkcije**. Naime, za autokorelacionu metodu znamo da ostaje periodična za periodične signale, i to sa istom periodom kao i signal za koju se računa. Zato za govornu sekvencu procenjujemo autokorelacionu funkciju, u kojoj očekujemo pik tamo gde se nalazi *pitch* perioda. Da bismo procenili *pitch* periodu korišćenjem ove metode, izvršili smo sledeće korake:

- klipovanje signala,
- određivanje autokorelacione funkcije govornog signala,
- nalaženje pikova u autokorelacionoj funkciji,
- nalaženje *threshold*-a prvog bitnog pika i uzimanje relevantnih pikova,
- računanje srednje udaljenosti između pikova,
- računanje *pitch* frekvencije.

Ovim načinom dobija se procena *pitch* frekvencije koja iznosi 100Hz. Vidimo da obe metode daju približno sličnu procenu *pitch* frekvencije, koja se nalazi u očekivanom opsegu za muškog govornika.

2 Zadatak 2

- Korišćenjem komercijalnog mikrofona u programskom okruženju **MATLAB**, snimiti govornu sekvencu u dužini od 20-ak sekundi. Sekvencu snimiti sa frekvencijom odabiranja 8kHz u šesnaestobitnoj (*default*) rezoluciji.
- Isprojektovati $\mu = 100$ i $\mu = 500$ kompanding kvantizator sa 4, 8 i 12 bita i za njih odrediti zavisnost odnosa signal-šum za različite vrednosti odnosa (X_{max}/σ_x). Ovaj odnos menjati promenom varijanse korisnog signala, prostim skaliranjem početne snimljene sekvence. Prikazati rezultate grafički.
- Isprojektovati Δ kvantizator za sekvencu iz prve tačke. Adekvatno podesiti parametar Δ tako da se dobije što bolji kvalitet kvantizacije. Uporediti oblike originalnog i kvantizovanovog signala. Šta se dešava kada je korak kvantizacije Δ previše mali ili previše veliki? Da li se histogram priraštaja može koristiti za određivanje adekvatnog parametra Δ? Pratiti kvalitet zvuka i promene u amplitudi za svaki slučaj.

2.1 μ-kompanding kvantizator

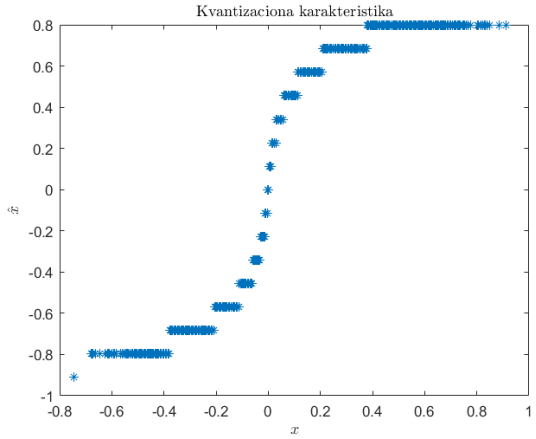
μ-kompanding kvantizator ima za cilj da poboljša rezultate uniformnog kvantizatora tako što će pre kvantizacije signal propustiti kroz sledeću funkciju transformacije:

$$F(x[n]) = X_{max} \frac{log\left[1 + \mu \frac{|x[n]|}{X_{max}}\right]}{log[1+\mu]} sign(x[n]).$$

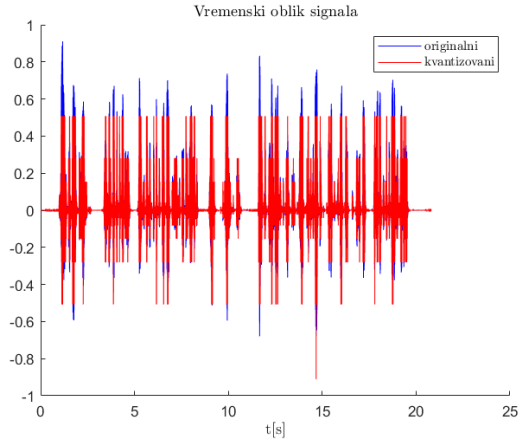
Nakon kvantizacije signal se kodira i transformiš na željenu lokaciju gde se pre dekodiranja propušta kroz inverznu funkciju $F^{-1}(x[n])$. Na ovaj način odnos signal-šum (engl. skraćenica **SNR**) opada značajno sporije (u odnosu na uniformni kvantizaotor), stoga je kvalitet kvantizacije značajno bolji za glasne signale nego kod uniformnog kvantizatora. Odnos signal-šum za ovaj kvantizator je dat sledećim izrazom:

$$SNR = 6.02B - 4.77 - 20log(log(1+\mu)) - 10log\left[1 + \sqrt{2}\frac{X_{max}}{\mu\sigma_x} + \left(\frac{X_{max}}{\mu\sigma_x}\right)^2\right].$$

Pored ovog izraza, odnos signal-šum je bio računat i eksperimentalno kao odnos varijanse govornog signala i varijanse greške kvantizacije. Na sledećim graficima su prikazane kvantizacione karakteristike kao i vremenski oblik originalnog i kvantizovanog signala za različite vrednosti μ (100 i 500) i različite brojeve bita b (4, 8 ili 12). Možemo primetiti da sa povećanjem broja bita kvantizacije, kvantizovani signal postaje sličniji originalnom, a sa promenom μ se menja oblik kvantizacione karakteristike.

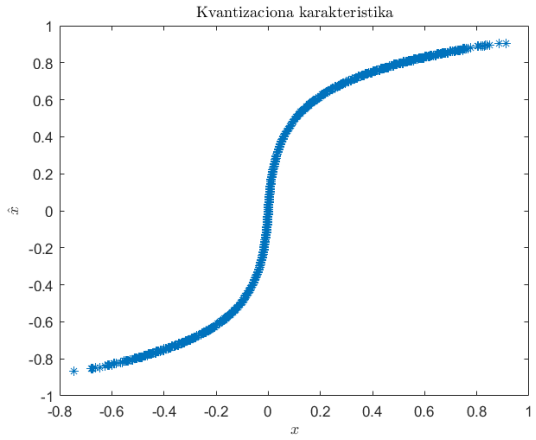


(a) Kvantizaciona karakteristika

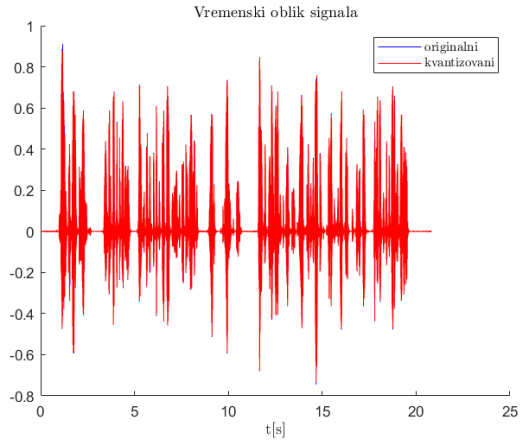


(b) Vremenski oblik signala

Slika 6: $\mu = 100$, $b = 4$

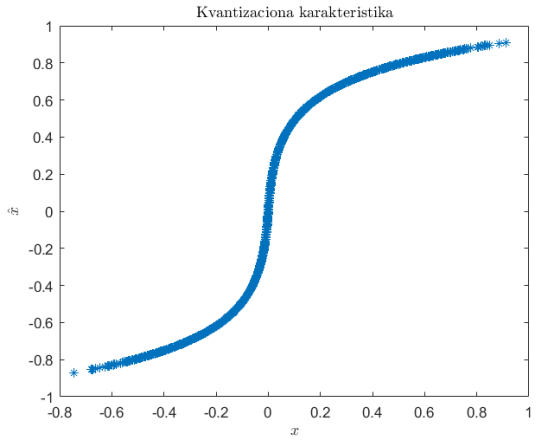


(a) Kvantizaciona karakteristika

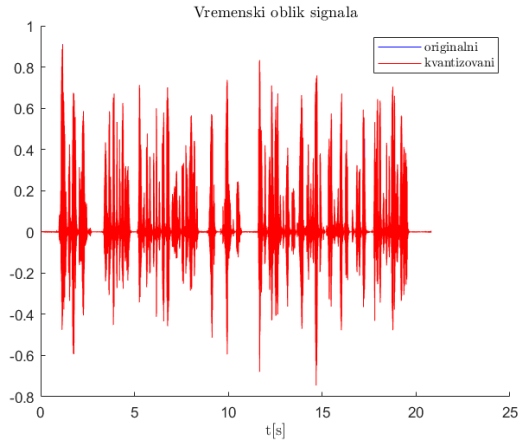


(b) Vremenski oblik signala

Slika 7: $\mu = 100$, $b = 8$

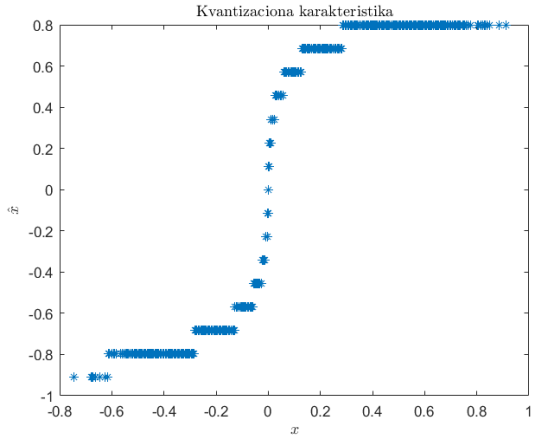


(a) Kvantizaciona karakteristika

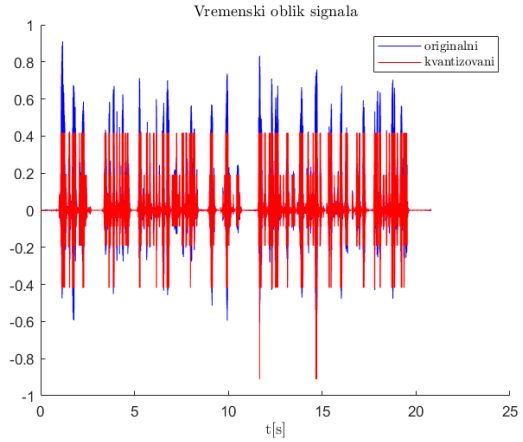


(b) Vremenski oblik signala

Slika 8: $\mu = 100$, $b = 12$

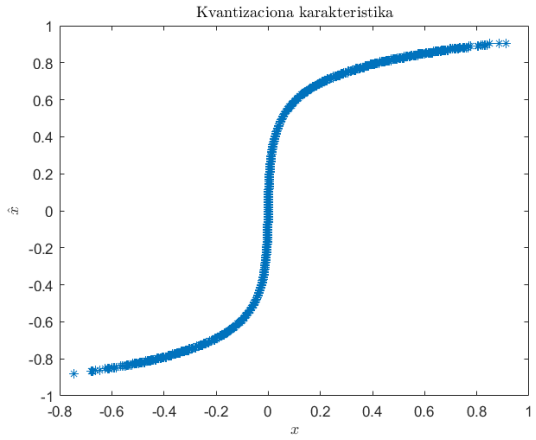


(a) Kvantizaciona karakteristika

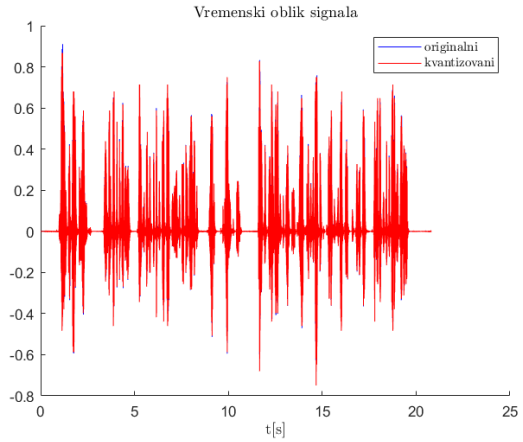


(b) Vremenski oblik signala

Slika 9: $\mu = 500$, $b = 4$

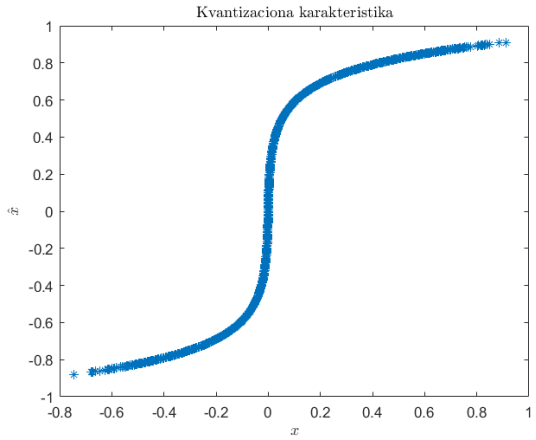


(a) Kvantizaciona karakteristika

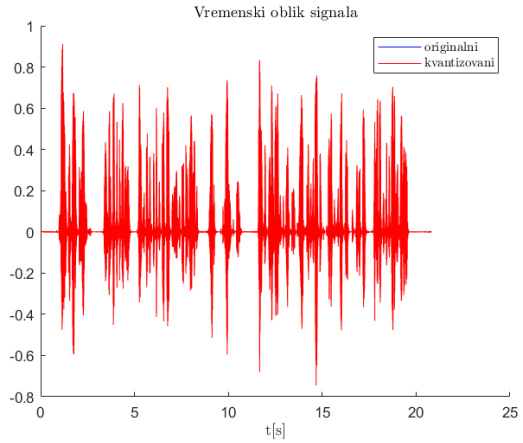


(b) Vremenski oblik signala

Slika 10: $\mu = 500$, $b = 8$



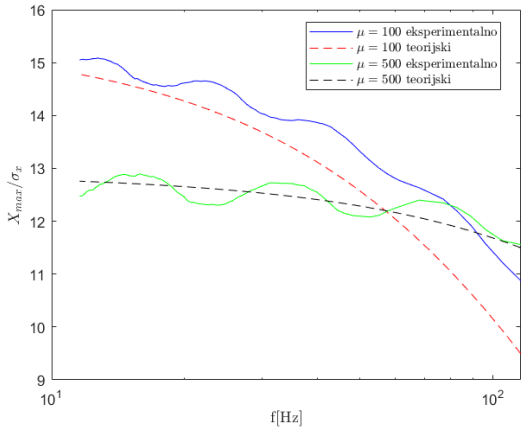
(a) Kvantizaciona karakteristika



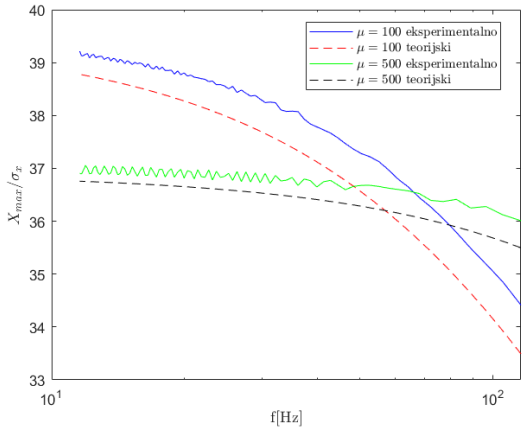
(b) Vremenski oblik signala

Slika 11: $\mu = 500$, $b = 12$

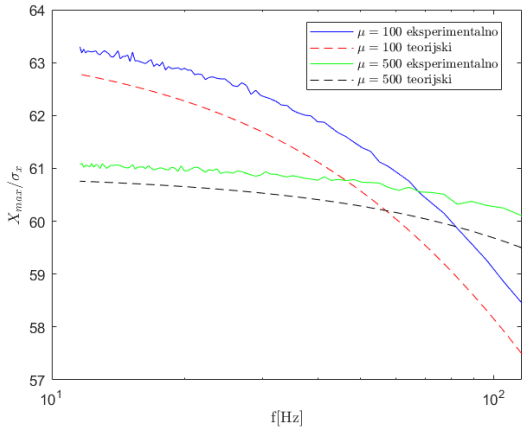
Na sledećoj slici 12 prikazane su **SNR** karakteristike za različite μ i brojeve bita b dobijene teorijski i eksperimentalno.



(a) $b = 4$ bita



(b) $b = 8$ bita



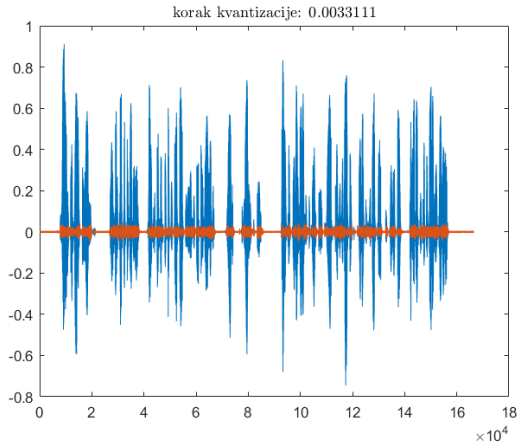
(c) $b = 12$ bitova

Slika 12: **SNR** karakteristike za različite μ i različite b

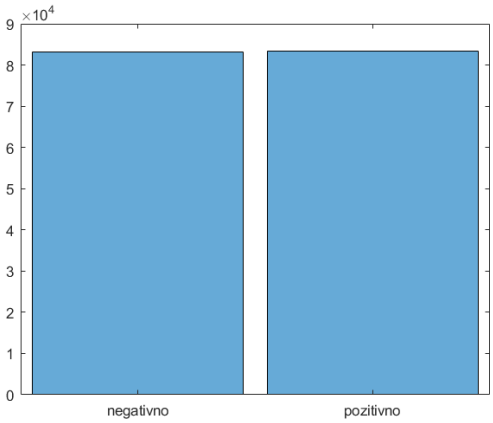
Može se uočiti da sa porastom broja bita b rastu vrednosti **SNR**-a, a sa porastom μ nagib se smanjuje, pa je kvantizacija kvalitetnija za tiše signale, međutim, za manje vrednosti $\frac{X_{max}}{\sigma_x}$ odnos **SNR** opada sa porastom μ, te je za glasnije signale rezultat nešto lošiji.

2.2 Delta kvantizator

Potrebno je isprojektovati delta (Δ) kvantizator za istu sekvencu kao u prethodnoj tački. Prvo ćemo isfiltrirati signal, a zatim odrediti neke vrednosti Δ i videti kako se ponaša sekvenca pri različitim vrednostima Δ. Glavni cilj Δ kvantizatora je da se što bolje približi stvarnom signalu koristeći samo dve vrste priraštaja ($+\Delta$ ako je razlika trenutnog i prediktovanog signala veća od 0, i suprotno, $-\Delta$, ako je razlika trenutnog i prediktovanog signala manja od 0).

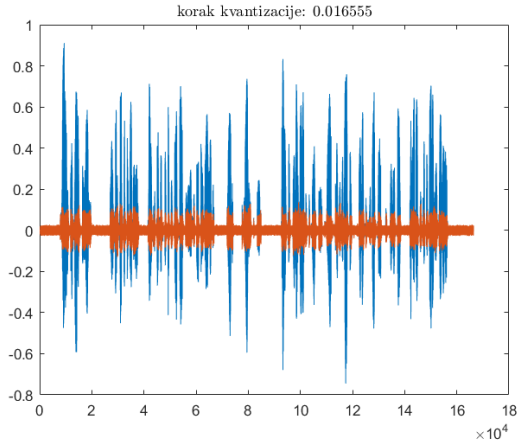


(a) Vremenski oblik signala pre i nakon kvantizacije

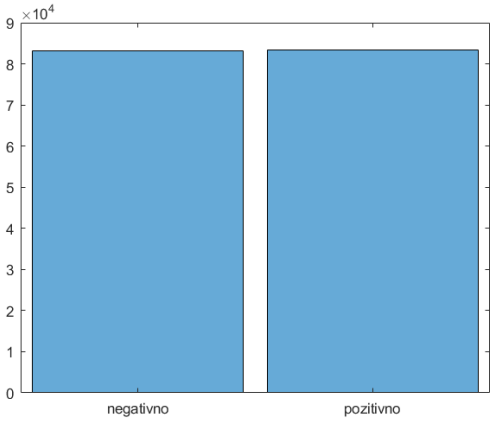


(b) Histogram priraštaja

Slika 13: $\Delta = 0.0033111$

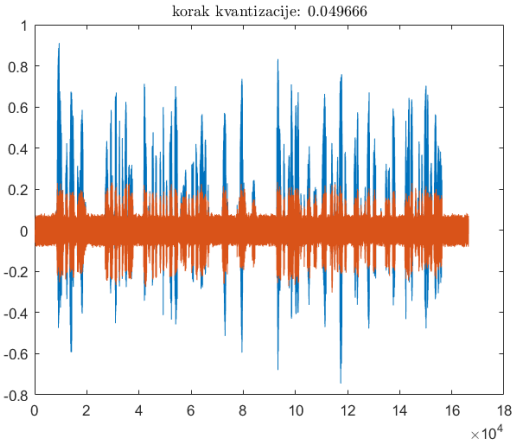


(a) Vremenski oblik signala pre i nakon kvantizacije

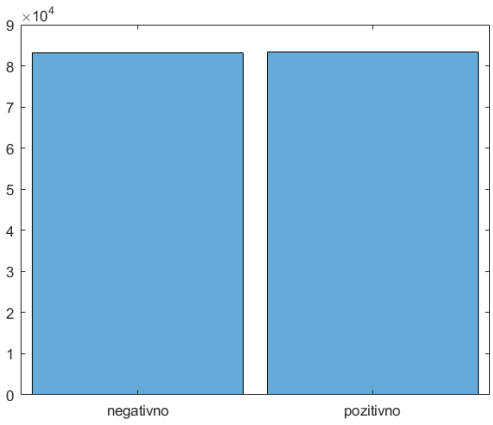


(b) Histogram priraštaja

Slika 14: $\Delta = 0.016555$

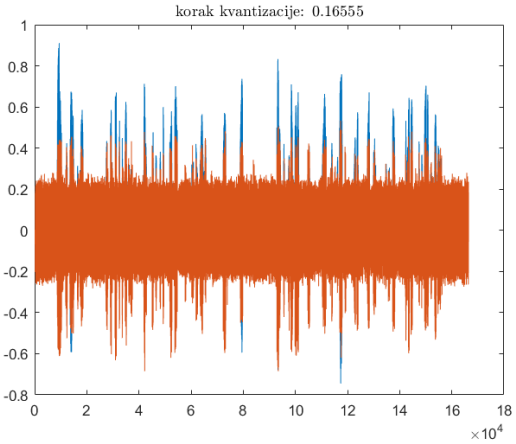


(a) Vremenski oblik signala pre i nakon kvantizacije

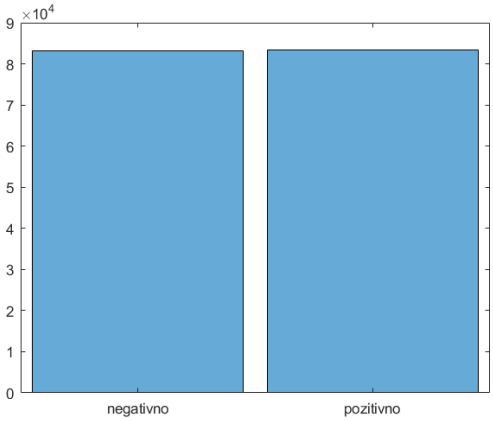


(b) Histogram priraštaja

Slika 15: $\Delta = 0.049666$



(a) Vremenski oblik signala pre i nakon kvantizacije



(b) Histogram priraštaja

Slika 16: $\Delta = 0.16555$

Na osnovu dobijenih rezultata možemo proceniti da je najbolji korak kvantizacije za datu sekvencu **0.0166555**. Ukoliko je korak kvantizacije previše mali, dobićemo tišu sekvencu na izlazu, dok u slučaju prevelikog koraka kvantizacije dobijamo glasniju sekvencu na izlazu sa izraženim izobličenjima. Kao što možemo primetiti sa histograma priraštaja, broj negativinih i broj pozitivnih priraštaja je približno isti za svako Δ, pa iz ove informacije ne možemo odrediti optimalan korak u zavisnosti od histograma.

3 Zadatak 3

Snimiti bazu za 3 izgovorene cifre, gde je svaka cifra izgovorena 10 puta od strane istog govornika (30 sekvenci u bazi).

- Napisati funkciju *preprocessing* koja prima govornu sekvencu i vraća je nakon izvršene predobrade (segmentacije i filtriranja).
- Implementirati funkciju *feature_ extraction* koja za prosleđenu sekvencu vraća obeležja zasnovana na LPC i/ili kepstralnim koeficijentima (dozvoljeno je korišćenje ugrađenih funkcija uz teorijski opis).
- Konačna funkcija *cifer_ recognition* treba da pokrene kod za snimanje govorne sekvence, i zatim da obeležja snimljene sekvence dobijena na osnovu funkcija iz tačaka 1 i 2 prosledi klasifikatoru po izboru. Kada klasifikator donese odluku, ispisati je u komandnom prozoru.
- Uspešnost klasifikacije testirati na po 5 novosnimljenih sekvenci iz svake klase i prikazati u obliku konfuzione matrice. Takođe prikazati konfuzionu matricu za trening skup. Za svaku od navedenih tačaka dati sažet pregled teorije na kojoj se zasniva, kao i detaljan opis implementacije same funkcije. Rezultate svake tačke prikazati grafički na odabranoj sekvenci i prokomentarisati uticaj izbora obeležja i klasifikatora na ishod klasifikacije. Izdvojiti i prokomentarisati primere tačno i pogrešno klasifikovanih sekvenci.

U cilju projektovanja sistema za prepoznavanje izgovorenih cifara formiran je trening i test skup. Sistem radi sa ciframa 0, 2 i 5. U trening skupu se nalazi ukupno 30 sekvenci (po 10 za svaku cifru), dok se u test skupu nalazi 15 sekvenci (po 5 za svaku cifru). Svaku sekvencu najpre filtriramo i segmentujemo na isti način kao i u prvom zadatku pa ćemo ovde preskočiti detaljnije objašnjenje implementacije funkcije *preprocessing*. U okviru funkcije *feature_ extraction* se računaju LPC koeficijenti tako što se sekvenca prozoruje pravougaonim prozorom dužine 20 ms odnosno 160 odabiraka, a zatim za svaki prozor procene parametri AR modela korišćenjem *Yule-Walker*-ove metode.

Yule-Walker-ova metoda se zasniva na tome da se procena autokorelacione funkcije iskoristi za rešavanje *Yule-Walker*-ovih jednačina koje su date sledećom relacijom:

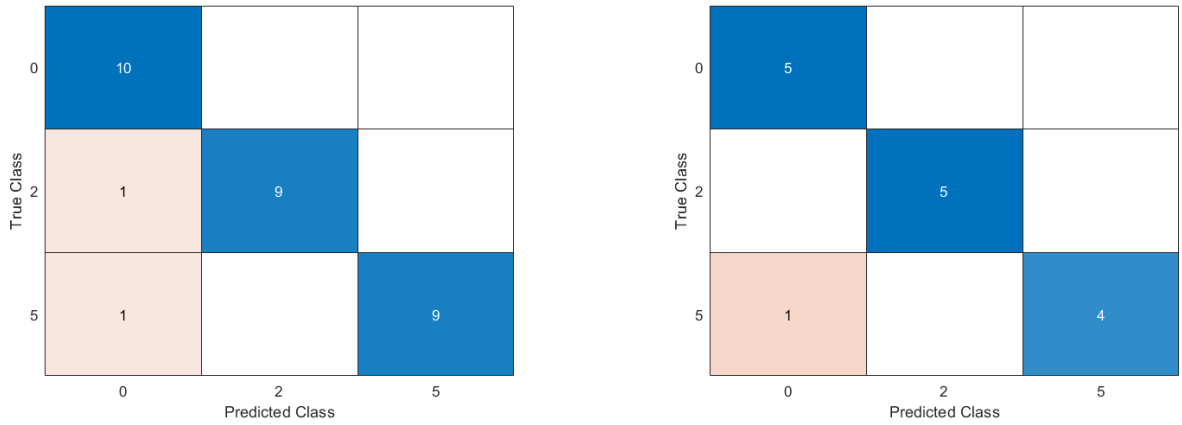
$$\begin{bmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \dots & \hat{r}_{xx}[-(p-1)] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}[0] & \dots & \hat{r}_{xx}[-(p-2)] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{xx}[p-1] & \hat{r}_{xx}[p-2] & \dots & \hat{r}_{xx}[0] \end{bmatrix} \begin{bmatrix} \hat{a}[1] \\ \hat{a}[2] \\ \vdots \\ \hat{a}[p] \end{bmatrix} = - \begin{bmatrix} \hat{r}_{xx}[1] \\ \hat{r}_{xx}[2] \\ \vdots \\ \hat{r}_{xx}[p] \end{bmatrix},$$

pri čemu je p red modela koji je u našem slučaju $p = 15$. Procena autokorelacione funkcije $\hat{r}_{xx}[k]$ je pomerena.

Sada imamo $p + 1$ LPC koeficijenta za svaki prozor svake sekvence. Kako broj prozora u sekvenci zavisi od dužine sekvence (koje su sve različite dužine), to se broj svakog od LPC koeficijenta razlikuje od sekvence do sekvence, te je za obeležje uzeta srednja vrednost po vrstama odnosno svaki LPC koeficijent je usrednjen po prozorima i na taj način nam je za svaku sekvencu preostalo tačno $p + 1$ koeficijenta.

Projektovan je klasfikator Euklidove distance koji test sekvencu proglašava za onu klasu koja je najbliža centru klase koja je izračunata na trening skupu.

Rezultati klasifikacije prikazani su na slici 17 ispod.



(a) Trening skup

(b) Test skup

Slika 17: Konfuzione matrice klasifikacije cifara

Uočavamo greške klasifikacije i u trening i u test skupu za slučaj kada izgovorenu cifru 5 klasifikator proglasi za 0. Dodatno, u trening skupu, klasifikator je pogrešno odlučio i u jednom slučaju kada je cifru 2 proglasio za 0, stoga stiče se utisak da klasifikator je naklonjen klasi 0 i možda bi trebalo drugačije projektovati klasifikator, ali i ovaj jednostavni klasifikator distance daje vrlo zadovoljavajuće rezultate. Tačnost oba skupa je ista i iznosi 93.33%.