Katedra za Signale i sisteme
Elektrotehnički fakultet
Univerzitet u Beogradu

Obrada i prepoznavanje govora

Domaći zadatak

Student:
Aleksandar Đorđević 2019/0086

Mentori:
prof. dr Željko Đurović
dipl. ing Natalija Đorđević

Sadržaj

1 Zadatak 1

1.1 Postavka problema

1. Korišćenjem komercijalnog mikrofona u programskom okruženju *MATLAB*, snimiti govornu sekvencu u dužini od 20-ak sekundi. Sekvencu snimiti sa frekvencijom odabiranja 8 ili 10 kHz i ona treba da se sastoji od desetak jasno segmentiranih reči

2. 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)

3. 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.

1.2 Rešenje

1.2.1 Segmentacija reči

Snimljena je govorna sekvenca od 12 reči sa pauzama između njih radi lakše segmentacije. Korišćena je frekvencija odabiranja od $8kHz$ sa kvantizacijom na 16 bita. Segmentacija je vršena pomoću kratkovremenske energije i kratkovremenske brzine prolaska kroz nulu.

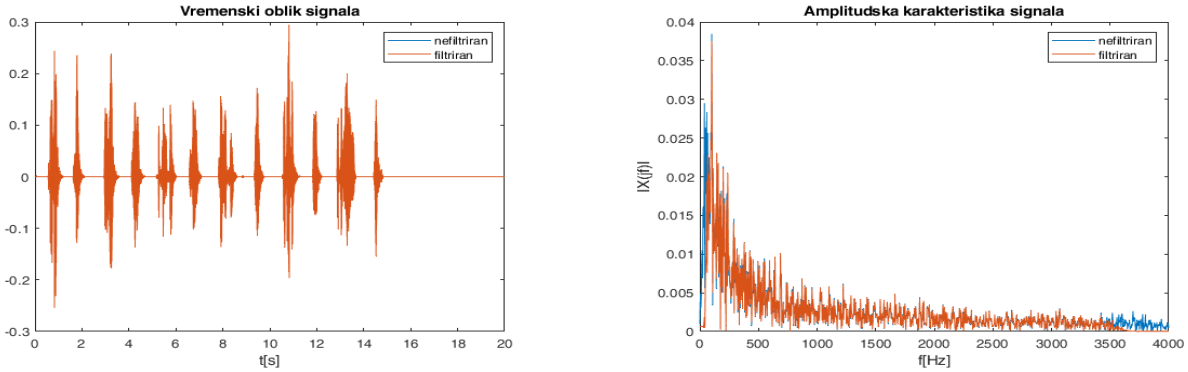
Kako je govorni signal nestacionaran on se prvo prozoruje pa se onda računaju energija i brzina prolaska kroz nulu po sledećim relacijama.

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \tag{1}$$

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

gde je x govorni signal, w prozorska funkcija (konkretno u našem slučaju pravougaona) dužine L. Za dužinu prozorske funkcije uzeto je 20 ms odnosno 160 odabiraka.

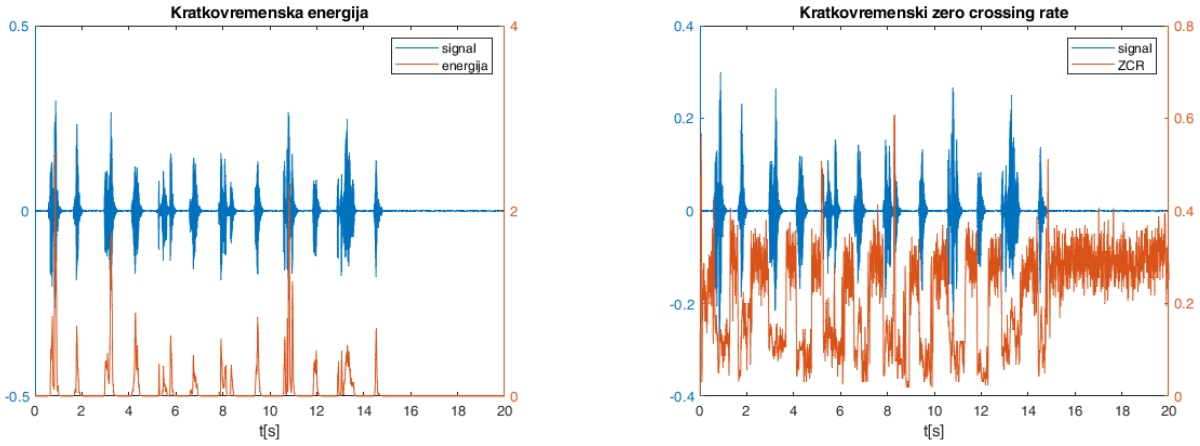
Takođe, pre računanja ovih karakteristika signal je propušten kroz Butterworth-ov filtar propusnik učestanosti od 60 do 3500 Hz kako bi se eliminisao šum, a pritom informacija sadržana u govornom signalu ostala nepromenjena. Na sledećim graficima prikazani su vremenski oblik signala i njegov amplitudski spektar pre i posle filtriranja.



Slika 1: Govorni signal pre i posle filtriranja

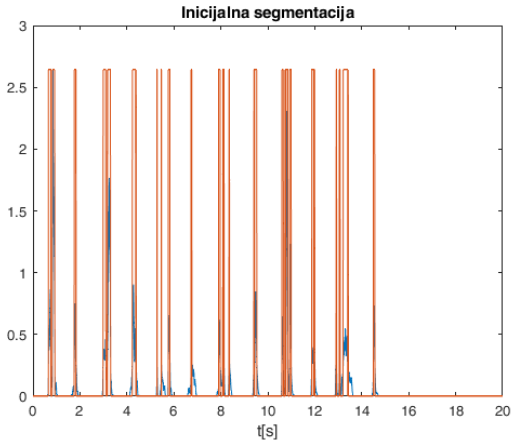
Možemo uočiti da se u predstavom signala u vremenu jasno uočavaju 12 oblasti koje predstavljaju izgovorene reči te ne bi trebalo da bude većih problema prilikom segmentacije (osim možda odsecanja krajeva reči koje se završavaju bezvučnim glasovima). Takođe, primećuje se da efekat filtriranja nije vidljiv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

Na sledećim graficima su prikazane kratkovremenska energija i kratkovremenska brzina prolaska kroz nulu filtriranog signala.



Slika 2: Kratkovremenske karakteristike govornog signala

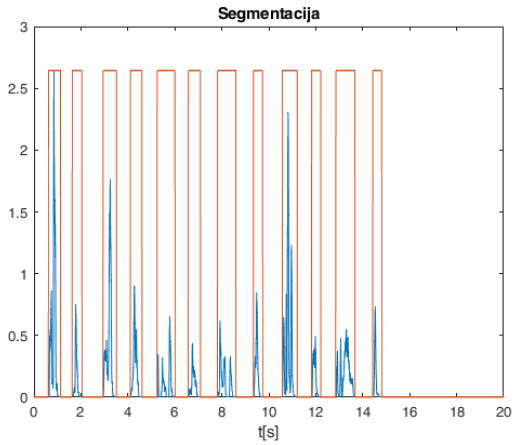
Možemo primetiti da je energija velika tamo gde ima izgovorenih reči, a mala gde je tišina. Za brzinu prolaska kroz nulu važi suprotno to jest ona je velika kada je tišina nego kada postoji govor. Koristeći ove dve osobine moguće je segmentirati reči iz govorne sekvence koristeći sledeći postupak. Prvo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



Slika 3: Segmentacija nakon poređenja sa višim pragom

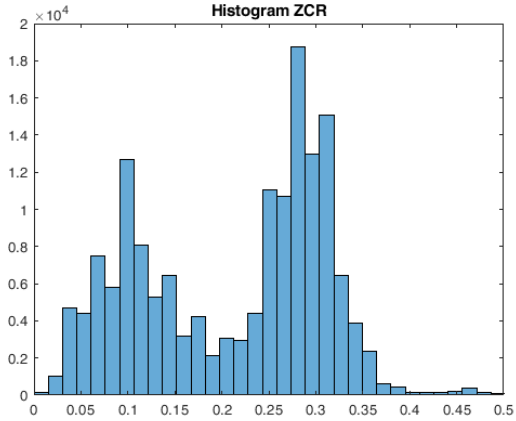
Na slici vidimo da postoji više segmenata nego što ima reči to jest izbačeni su neki delovi reči koji nisu toliko zvučni i reč na taj način podeljena na više segmenata. Zbog ovoga se uvodi takozvana niska granica ITL (takođe se određuje kao procenat maksimalne energije, konkretno ovde je uzeto 0.01%). Sada za svaku reč krenemo od njenog početka i idemo ulevo sve dok energija ne padne ispod ITL i ovo je novi početak reči. Slično i za kraj idemo udesno sve dok je energija veća od ITL i to nam predstavlja novi početak reči.

Naravno, ako se kraj i početak dva susedna segmenta preklope spajamo ih u jednu reč. Nakon poređenja sa nižim pragom segmentacija izgleda ovako:



Slika 4: Segmentacija nakon poređenja sa nižim pragom

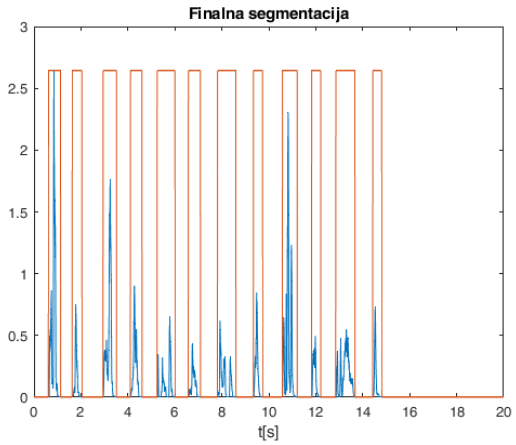
Sada možemo primetiti da su reči dobro izdvojeno iz govorne sekvence (imamo 12 segmenata, a znamo da je toliko isto i reči). Nakon slušanja zaključujemo da su rezultati zadovoljavajući, međutim na počecima i krajevima određenih reči deluje da su glasovi malo odsečeni i ovo se može popraviti pomoću kratkovremenske brzine prolaska kroz nulu. Motivacija za ovo se može uočiti na sledećoj slici.



Slika 5: Histogram kratkovremenske brzine prolazaka kroz nulu

Na histogramu se jasno vidi da je raspodela bimodalna odnosno da se može predstaviti kao zbir dve zasebne raspodele. Raspodela na nižim vrednostima potiče od dela signala gde postoji govor dok raspodela na višim vrednostima potiče od tišine. Sa ovog grafika se zaključuje da se prag treba postaviti na IZCT = 0.2.

Algoritam za ovu popravku izgleda na sledeći način. Krenemo od granice reči i gledamo 25 prozorskih funkcija ulevo odnosno udesno. Ako je u ovom opsegu ZCR barem tri puta preseca prag IZCT početak odnosno kraj reči se postavlja na poslednje mesto gde je ZCR bilo veće od IZCT. Nakon ove popravke segmentacija izgleda ovako:



Slika 6: Segmentacija nakon popravke pomoću ZCR

Na grafiku se ne vide značajnije razlike, ali se slušanjem čuju blage razlike na krajevima reči te zaključujemo da je ova obrada makar malo poboljšala rezultate.

Rezultate je moguće poboljšati korišćenjem *Teager* energije umesto obične, ali kako su oni već dobri ovo nije neophodno.

1.2.2 Procena *pitch* periode

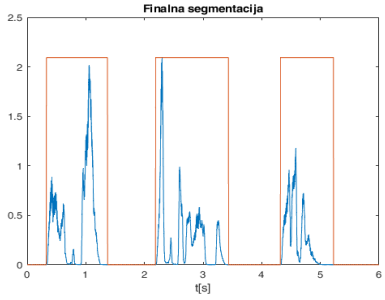
Estimacija *pitch* je veoma važan zadatak u obradi govora i koristi se u raznim sistemima za identifikaciju i verifikaciju govornika. Ona predstavlja učestanost kojom osciluju glasne žice i jedinstvena je za svaku osobu. Kod muškaraca *pitch* frekvencija se kreće u opsegu (80, 160)Hz, za žene (150,250) Hz, dok se za decu kreće oko 300 Hz.

Gold i *Rabiner* su 1969. godine predložili **metod paralelnog procesiranja** za procenu *pitch* periode. Na početku se filtrira propusnikom učestanosti u onom opsegu u kojem se očekuje da bude vrednost *pitch* periode (dakle zavisi od toga dal procenjujemo muški, ženski ili dečiji glas). Kako je u našem slučaju u pitanju muški glas opseg je bio od 70 do 200 Hz. Zatim ulazi u blok za procesiranje koji kao izlaz daje 6 signala koje dobije na sledeći način:

1. impulsi jednaki vrednostima maksimuma originalnog signala na lokacijama u kojima su se oni pojavili, inače je nula
2. impulsi jednaki razlici maksimuma i prethodnog minimuma, a na mestu gde se pojavio maksimum, inače nula
3. impulsi jednaki razlici maksimuma i prethodnog maksimuma, a na mestu gde se pojavio tekući maksimum (ako je negativna stavlja se nula), inače nula
4. impulsi jedanki negativnim vrednostima minimuma na poziciji gde se on pojavio, inače nula
5. impulsi jednaki negativnoj razlici tekućeg minimuma i prethodnog maksimuma, a na lokaciji minimuma, inače nula
6. impulsi jednaki negativnoj razlici tekućeg minimuma i prethodnog minimuma, a na lokaciji tekućeg, inače nula

Za svaki od ovih signala se posebno procenjuje pitch perioda na sledeći način. Imamo funkciju koju setujemo na vrednost prvog odabirka, i ona ovu vrednost zadržava neko predodređeno vreme τ (*blanking time*). Ovo vreme ima za ulogu da obezbedi da *pitch* perioda ne bude manja od očekivane. Zatim funkcija kreće da opada eksponencijalno sa $e^{-\lambda t}$ i *pitch* perioda se dobija kao proteklo vreme od setovanja funkcije do prvog preseka sa odabirkom. Uzeto je $\tau = 4$ ms i $\lambda = 120s^{-1}$ Na kraju se na osnovu procena na svakom od signala i konačne procene iz prethodne iteracije konačna procena za trenutnu iteraciju dobija kao medijana ovih sedam vrednosti.

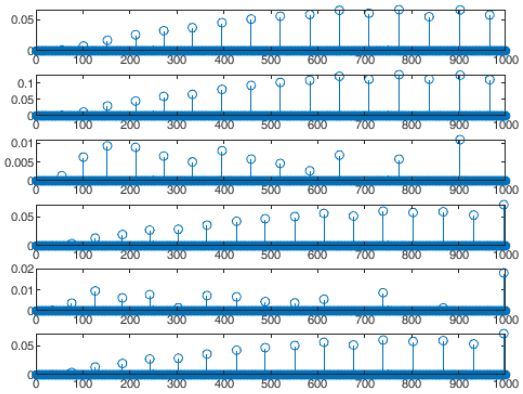
U cilju određivanja *pitch* periode sopstvenog glasa snimljena je sekvenca od 3 reči bogate samoglasnicima (onomatopeja, autoperionica i aerodinamika). Kako periodi tišine ne bi uticali na estimaciju reči su segmentirane korišćenjem metode iz prethodne tačke, zatim je na svakoj reči posebno procenjena izvršena estimacija. Segmentacija reči izgleda ovako:



Slika 7: Finalna segmentacija reči

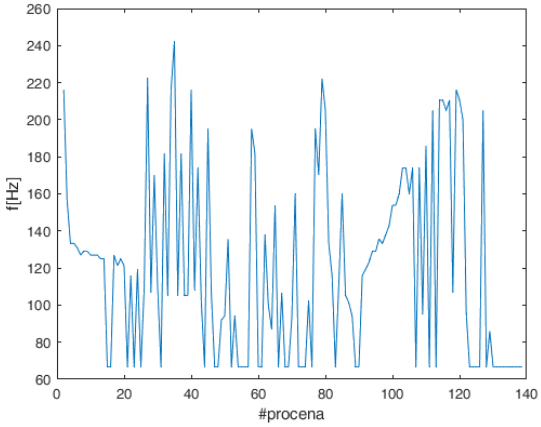
Kao što vidimo segmentacija je urađena na zadovoljavajući način.

Signali na izlazu bloka za procesiranje za prvu reč su prikazani na sledećoj slici.



Slika 8: Izlaz bloka za procesiranje

Procena *pitch* periode za prvu reč kroz vreme prikazana je na sledećem grafiku.



Slika 9: Procena *pitch* periode

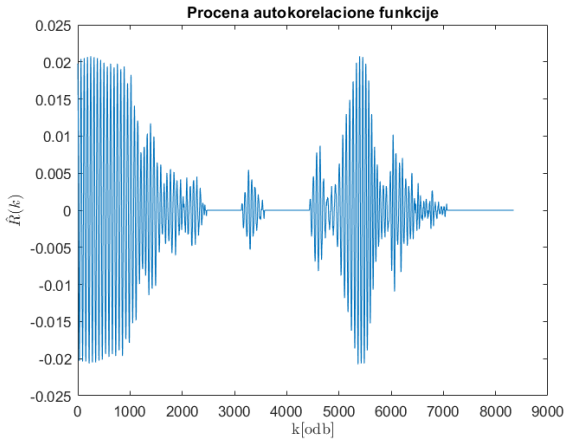
Medijane ovih procena za sve 3 reči su redom: $\hat{f}_{p1} = 111.11$ Hz, $\hat{f}_{p2} = 106.67$ Hz i $\hat{f}_{p3} = 105.2632$ Hz, što jeste u opsegu očekivanih učestanosti.

Druga veoma popularna metoda za procenu *pitch* periode je **autokorelaciona metoda**. Ona se zasniva na proceni autokorelacione funkcije i njenom osobinom da za periodične signale i sama bude periodična i to sa istom periodom kao i signal. Kako izgovoreni samoglasnici u vremenskom domenu osciluju baš sa *pitch* periodom te se ona može proceniti iz dela autokorelacione funkcije koji odgovara samoglasnicima.

Pre procene autokorelacione funkcije je izvršeno filtriranje istim filtrom kao i kod paralelnog procesiranja (propusnik učestanosti od 70 do 200 Hz). Nakon toga se vrši klipovanje (pokazano je da značajno poboljšava rezultate ove metode). Postoji više vrsta klipovanja, a ovde je upotrebljen *three level clipping* odnosno odabirci signala manji od praga $-C_L$ se postavljaju na -1, oni veći od C_L se postavljaju na 1, a ostali su 0. C_L se bira kao procenat maksimalne vrednosti signala (u našem slučaju 20%). Nakon ovog procena autokorelacione funkcije se vrši pomoću sledeće relacije:

$$\hat{R}_x^m(k) = \frac{1}{2N+1} \sum_{n=-N}^{N} x(n)w(m-n)x(n+k)w(m-n-k) \tag{3}$$

gde je $2N+1$ dužina prozorske funkcije w (u našem slučaju pravougaone). Korišćena je dužina prozora od 50 ms odnosno 400 odabiraka. Kako zbog delova gde su bezvučni glasovi procena na celoj autokorelacionoj funkciji ne daje zadovoljavajuće rezultate *pitch* frekvencija je procenjena na delovima koji potiču od samoglasnika (zbog jednostavnosti procena je izvršena na prvom glasu jer sve 3 reči počinju samoglasnikom). Dobijene su sledeće *pitch* frekvencije: $f_{p1} = 125.00$ Hz, $f_{p2} = 119.40$ Hz i $f_{p3} = 108.84$ Hz. Dobijene su nešto više učestanosti nego metodom paralelnog procesiranja, ali i dalje u očekivanom opsegu. Na sledećoj slici je prikazan procenjena autokorelaciona funkcija prve reči.



Slika 10: Procena autokorelacione funkcije

2 Zadatak 2

2.1 Postavka problema

1. Korišćenjem komercijalnog mikrofona u programskom okruženju *MATLAB*, snimiti govornu sekvencu u dužini od 20-ak sekundi. Sekvencu snimiti sa frekvencijom odabiranja 8 kHz u šesnaestobitnoj (*default*) rezoluciji.

2. 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.

3. Isprojektovati Delta kvantizator za sekvencu iz tačke 1. 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.

2.2 Rešenje

2.2.1 μ-kompanding kvantizator

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

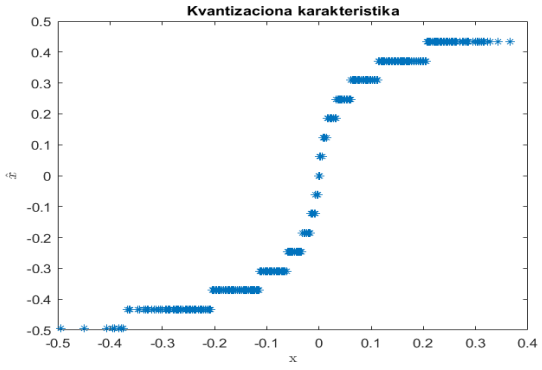
$$F(x[n]) = X_{max} \frac{log\left[1 + \mu \frac{|x[n]|}{X_{max}}\right]}{log[1+\mu]} sign(x[n]) \tag{4}$$

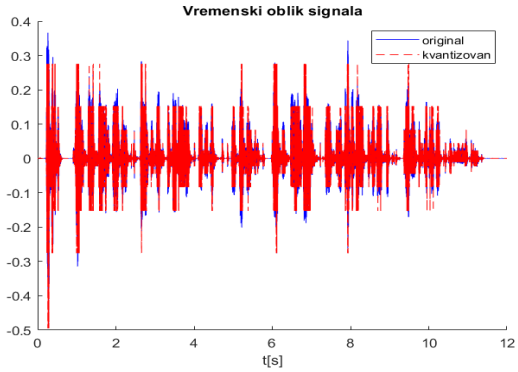
Nakon kvantizacije signla se kodira i transportuje na željenu lokaciju gde se pre dekodiranja propušta kroz funkciju $F^{-1}(x[n])$.

Na ovaj način SNR opada mnogo sporije te je kvalitet kvantizacije značajno bolji za glasne signale nego kod uniformnog kvantizatora. Odnos signal-šum za ovaj kvantizator je dat sledećim izrazom.

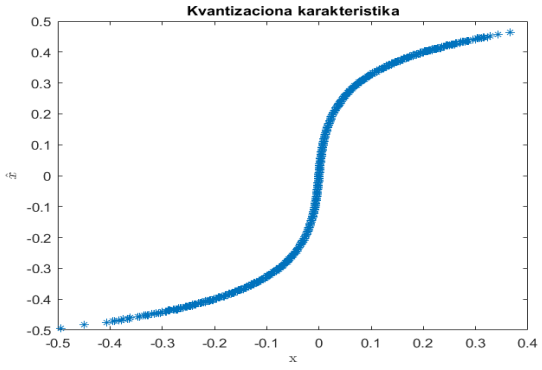
$$SNR[dB] = 6B + 4.77 - 20log_{10}(ln(1+\mu)) - 10log_{10}\left[1 + \left(\frac{X_{max}}{\mu\sigma_x}\right)^2 + \sqrt{2}\frac{X_{max}}{\mu\sigma_X}\right] \tag{5}$$

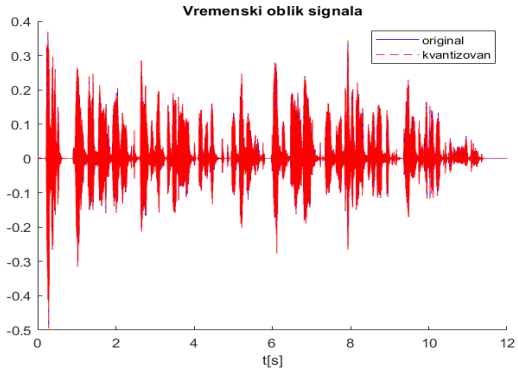
Pored ovog izraza odnos signal-šum je bio računat i eksperimentalno kao odnos varijanse govornog signala i varijanse greške kvantizacije. Na sledećim graficima su prikazane kvantizacione karakteristike kao i vremenski oblik originalnog i kvantizovanog signala za različite vrednosti μ i različite brojeve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



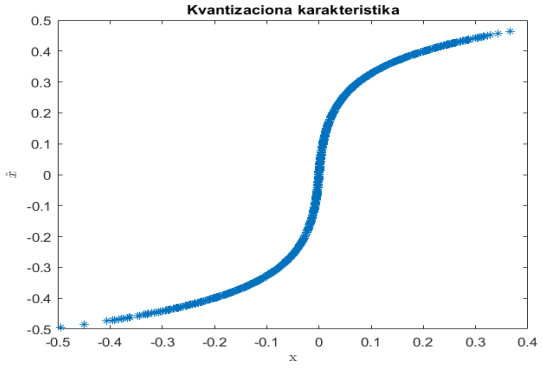


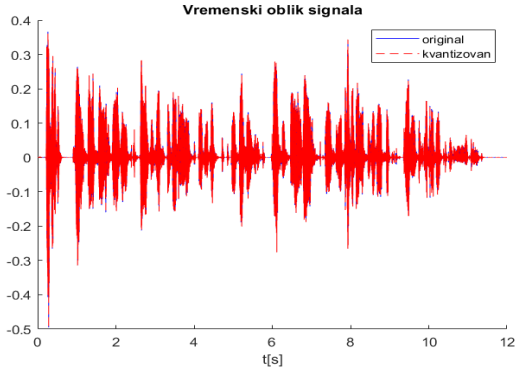
Slika 11: $\mu = 100$, $b = 4$



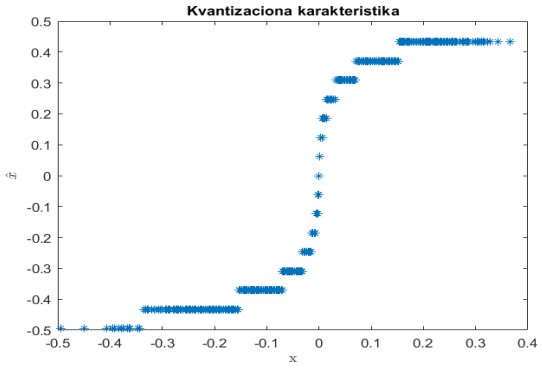


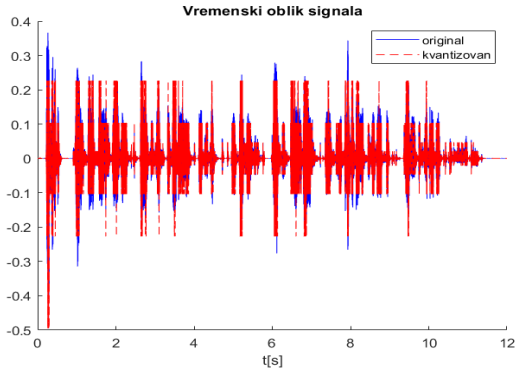
Slika 12: $\mu = 100$, $b = 8$



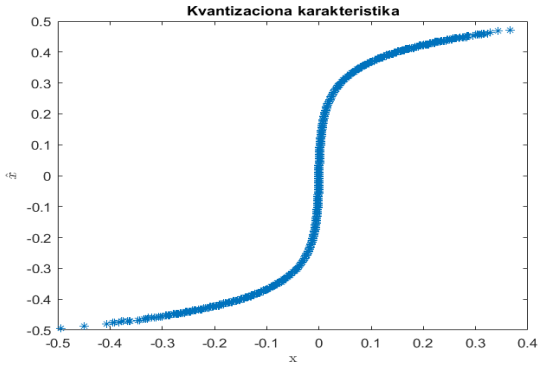


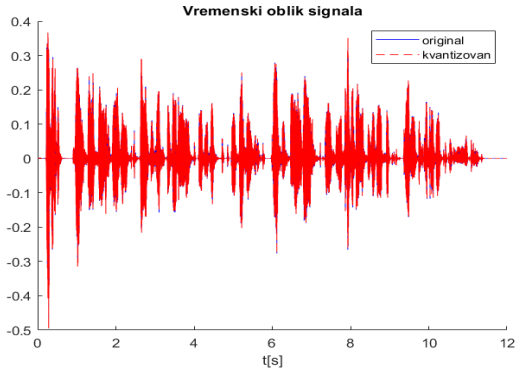
Slika 13: $\mu = 100$, $b = 12$



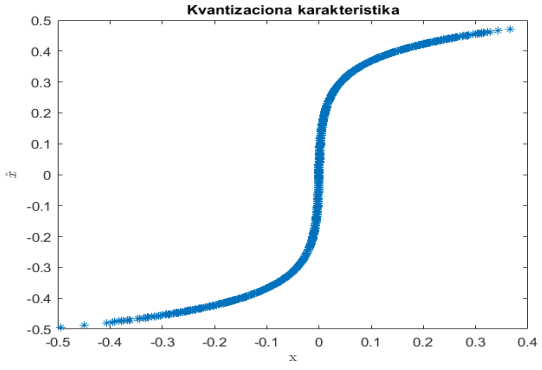


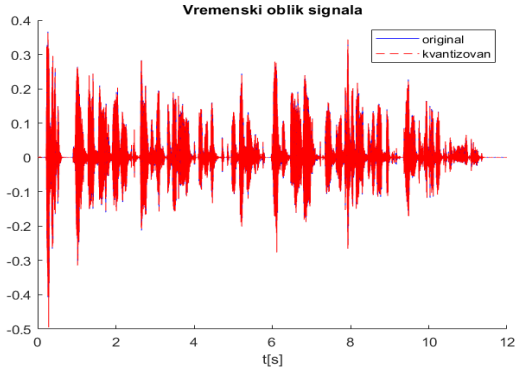
Slika 14: $\mu = 500$, $b = 4$





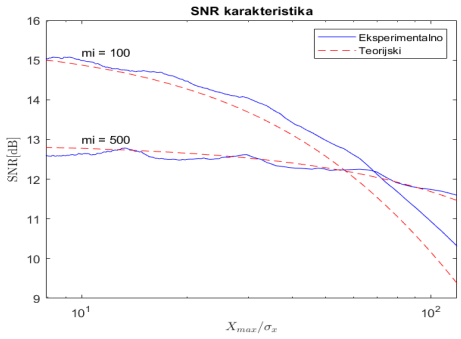
Slika 15: $\mu = 500$, $b = 8$



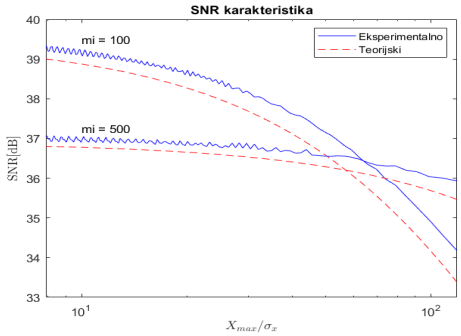


Slika 16: $\mu = 500$, $b = 12$

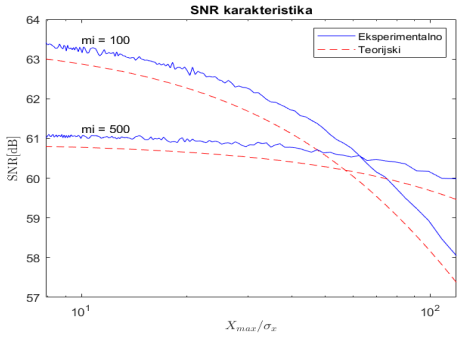
Na sledećim graficima prikazane su SNR karakteristike za različite μ i brojeve bita dobijene teorijski i eksperimentalno.



Slika 17: SNR karakteristika za 4 bita



Slika 18: SNR karakteristika za 8 bita

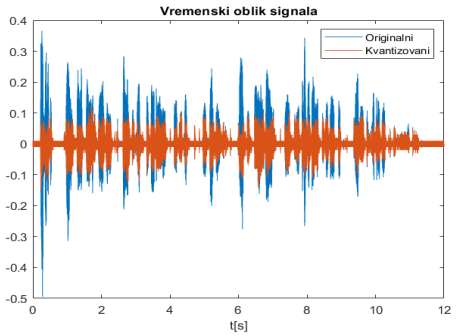


Slika 19: SNR karakteristika za 12 bita

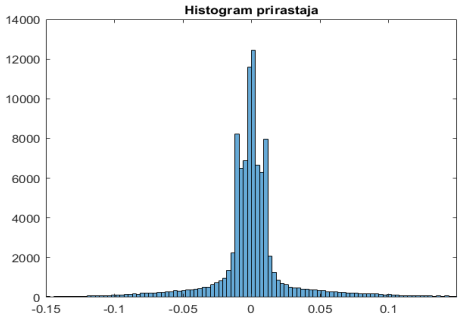
Možemo primetiti da sa porastom broja bita rastu vrednosti SNR-a, a sa porastom μ nagib se smanjuje te je kvantizacija kvalitetnija za tiše signale, međutim za manje vrednosti X_{max}/σ_x SNR opada sa porastom μ te je za glasnije signala rezultat nešto lošiji.

2.2.2 Delta kvantizator

Delta kvantizator je pojednostavljena verzija diferencijalnog kvantizatora. Kvantizuje se samo priraštaj signala odnosno razlika u odnosu na prethodni odabirak. Na ovaj način se štedi na memoriji odnosno potreban je samo jedan bit za kodiranje i zbog toga je moguće povećati brzinu. Unapred se odredi korak kvantizacije Δ koji je krucijalan za dobru kvantizaciju. Ako je premali neće moći da isprati nagib signala (mora da važi $\Delta > Tmax(x'[n])$). Ako je preveliki postojaće granularni šum odnosno javiće se značajne oscilacije. Kada smo odredili Δ sada su u svakoj iteraciji, zavisno od toga da li je odabirak signala veći ili manji od prethodnog odabirka kvantizovanog signala, prethodni odabirak kvantizovanog signala uvećava ili umanjuje za Δ i postaje novi odabirak.



Slika 20: SNR karakteristika za 12 bita



Slika 21: SNR karakteristika za 12 bita

Uzet je korak kvantizacije $\Delta = 0.01$ jer za veće korake granularni šum postaje veoma izražen, dok je za ovu vrednost govorni signal dosta slab, ali dovoljan da se razume šta se govori te je poruka uspešno preneta korišćenjem samo jednog bita.

Na drugoj slici se vidi histogram priraštaja govornog signala. Primetimo izražene pikove na vrednostima Δ i $-\Delta$.

3 Zadatak 3

3.1 Postavka problema

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

1. Napisati funkciju *preprocessing* koja prima govornu sekvencu i vraća je nakon izvršene predobrade (segmentacije i filtriranja).

2. 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).

3. 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.

4. 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.

3.2 Rešenje

U cilju projektovanja sistema za prepoznavanje izgovorenih cifara formiran je trening i test skup. Sistem radi na ciframa 1, 2 i 8. 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 zadatku 1 pa ćemo ovde preskočiti detaljnije objašnjenje. Nakon toga se računaju LPC 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**.

3.2.1 *Yule-Walker*-ova metoda

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

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \dots & r_{xx}(-p+1) \\ r_{xx}(1) & r_{xx}(0) & \dots & r_{xx}(-p+2) \\ \vdots & \vdots & & \vdots \\ r_{xx}(p-1) & r_{xx}(p-2) & \dots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = - \begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(p) \end{bmatrix} \tag{6}$$

gde je p red modela koji se za potrebe računanja LPC koeficijenata uzima običnog između 14 i 16, a u našem slučaju je $p = 15$.

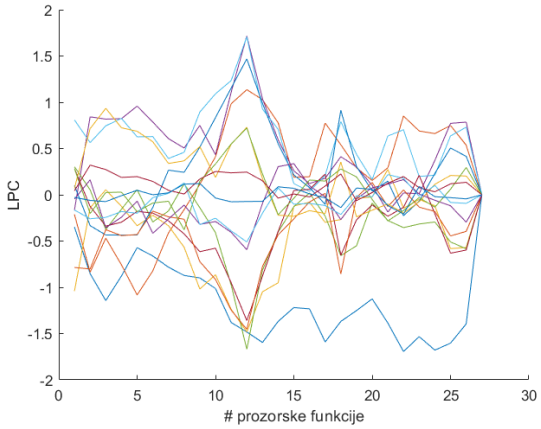
Za procenu autokorelacione funkcije se koristi pomereni estimator.

$$\hat{r}_{xx}(k) = \begin{cases} \frac{1}{N} \sum_{n=0}^{N-1-k} x^*(n)x(n+k) & ,k = 0, 1, \dots, p \\ r^*_{xx}(-k) & ,k = -1, -2, \dots, -p \end{cases} \tag{7}$$

Sada imamo $p + 1$ LPC koeficijenta za svaki prozor za svaku sekvencu. Kako broj prozora u sekvenci zavisi od dužine sekvence (a nama su naravno sekvence različite dužine) to se broj svakog od LPC koeficijenta razlikuje od sekvence do sekvence te je za obeleźje uzeta srednja vrednost po vrstama odnosno svaki

LPC koeficijenta je usrednjen po prozorima i na taj način nam je za svaku sekvencu ostalo tačno $p + 1$ koeficijenta.

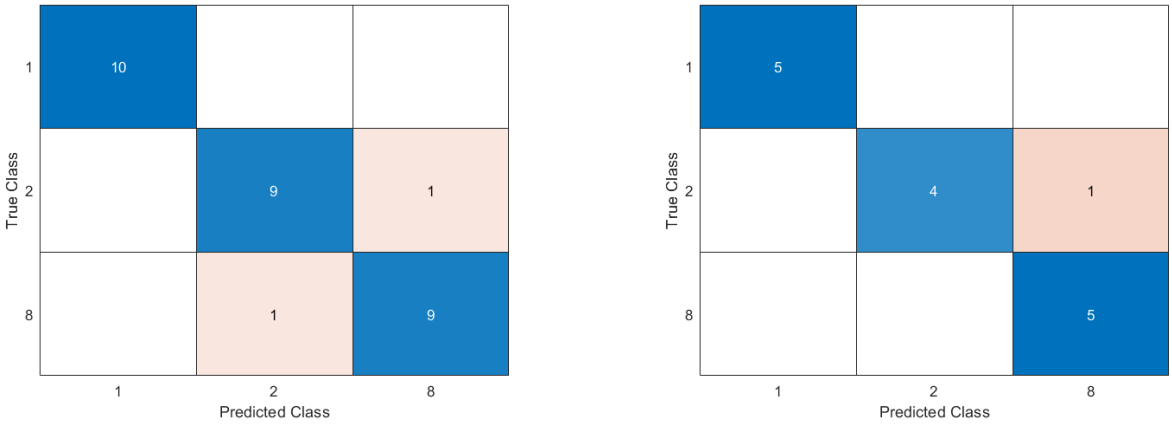
Na sledećem grafiku prikazani su LPC koeficijenti po prozorskim funkcijama za sekvencu u kojoj je izgovorena cifra 1.



Slika 22: LPC po prozorskim funkcijama

Projektovan je klasfikator distance koji test sekvencu progalašava za onu klasu koja je najmanje udaljena od centra klase koja je izračunata na trening skupu. Korišćen je Euklidova distanca.

Rezultati klasifikacije prikazani su na sledećoj slici. Možemo videti da je do greške dolazilo samo



(a) Trening skup

(b) Test skup

Slika 23: Konfuzione matrice

prilikom klasifikacije između 2 i 8, dok je 1 uvek dobro klasifikovana. I na trening i na test skupu uspešnost klasifikacije iznosi 93.33%.