# Domaći zadatak iz obrade i prepoznavanja govora

*Marija Brkić, 2020/0200*

June 17, 2024

## 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 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.

Određivanje početka i kraja reči je odrađeno Rabiner-ovom metodom koja se zasniva na kratkovremenskoj energiji i kratkovremenskoj brzini prolaska kroz nulu. Naime, postave se dva praga, gornji i donji prag, koji su prilagođeni datom govornom signalu. U implementaciji koda je korišćeno $ITU = 0.01max(E)$ i $ITL = 0.0004max(E)$. Nakon toga je pretpostavljeno da čitav deo signala čija se amplituda nalazi iznad gornjeg praga pripada rečima, a zatim se granice reči pomeraju ulevo i udesno do donjeg praga. Nakon toga je moguće da se granice reči dodatno pomeraju u skladu sa zero crossing rate-om, međutim u konkretnom slučaju to nije bilo potrebno, pošto je kratkovremenska energija bila dovoljna za segmentaciju reči.
Za početak se ulazni audio signal isfiltrirao kako bi se što više otklonio šum. Ulazni signal pre filtriranja izgleda na sledeći način:

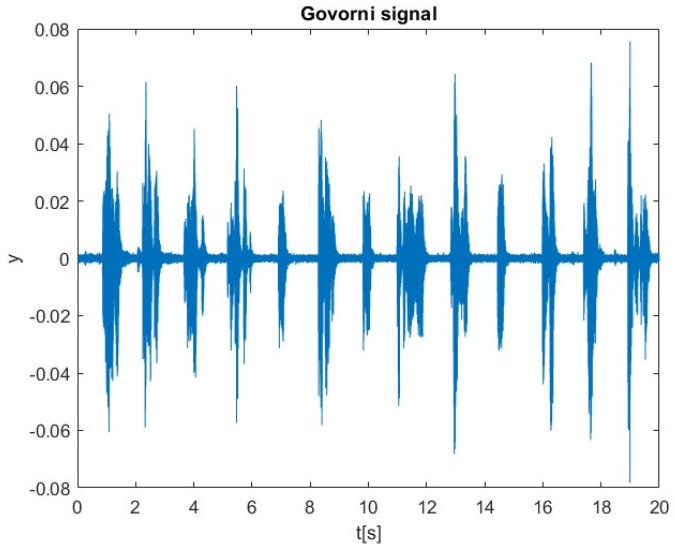


Figure 1: Ulazni audio signal

Amplitudski spektar ulaznog signala izgleda na sledeći način:

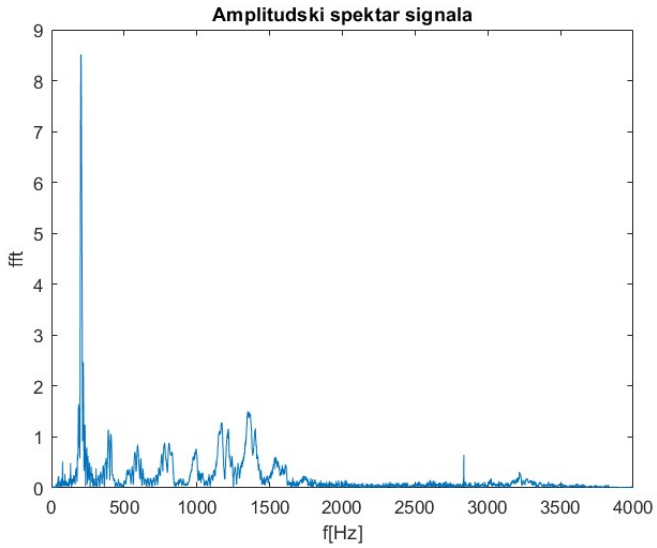


Figure 2: Spektar ulaznog signala

Nakon filtriranja signal i njegov spektar izgledaju na sledeći način:

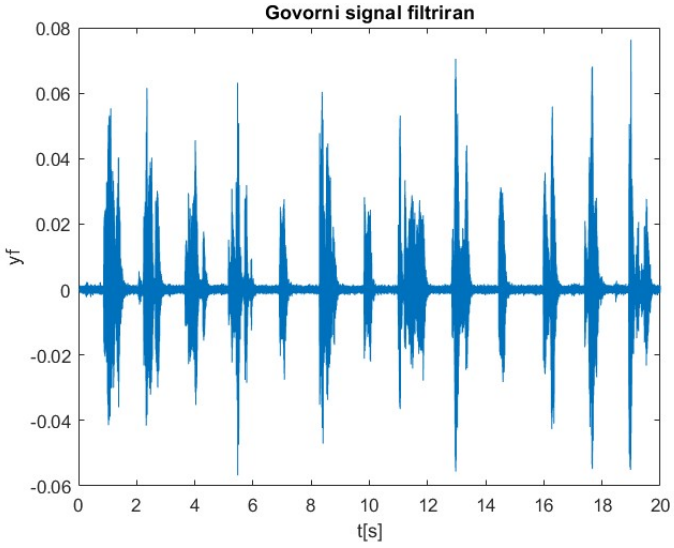


Figure 3: Ulazni audio signal nakon filtriranja

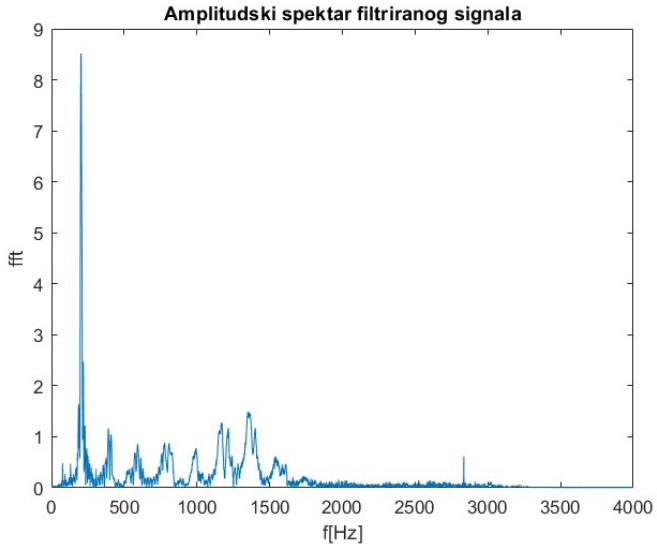


Figure 4: Spektar ulaznog signala nakon filtriranja

Kratkovremenska energija signala i kratkovremenski zero crossing rate izgledaju na sledeći način:

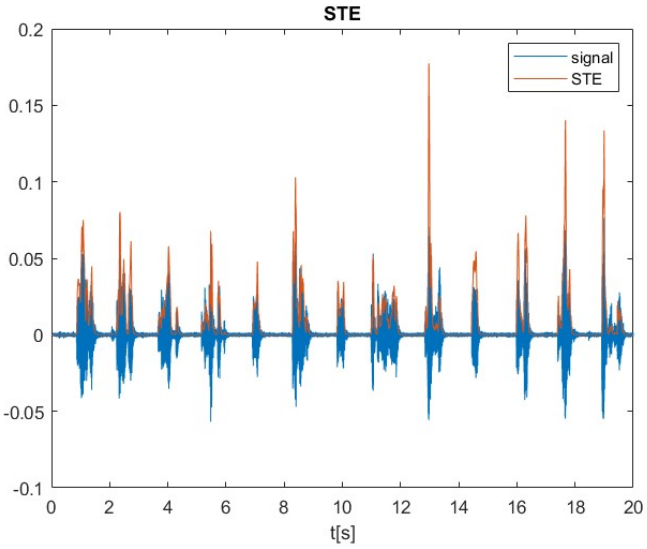


Figure 5: Kratkovremenska energija

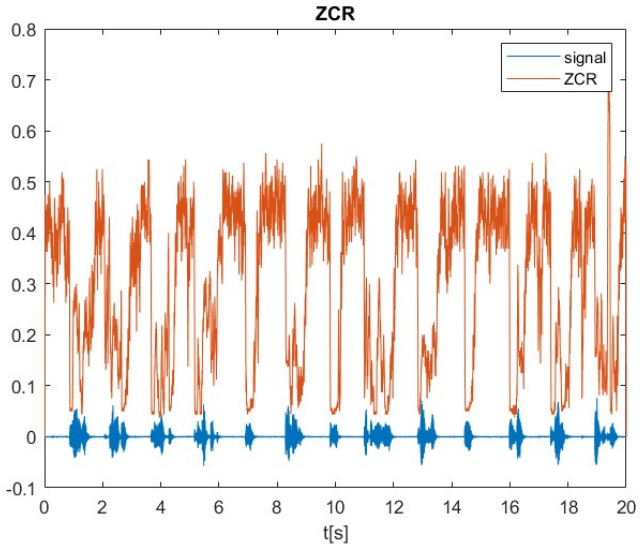


Figure 6: Kratkovremenski zero crossing rate

Na osnovu prethodna dva grafika možemo primetiti da zvučni delovi signala imaju visoku energiju, ali niži zero crossing rate, dok se bezvučni ponašaju obrnuto, što je u skladu sa očekivanjima. Konačno, signal sa izdvojenim rečima na gore objašnjen način je sledeći:

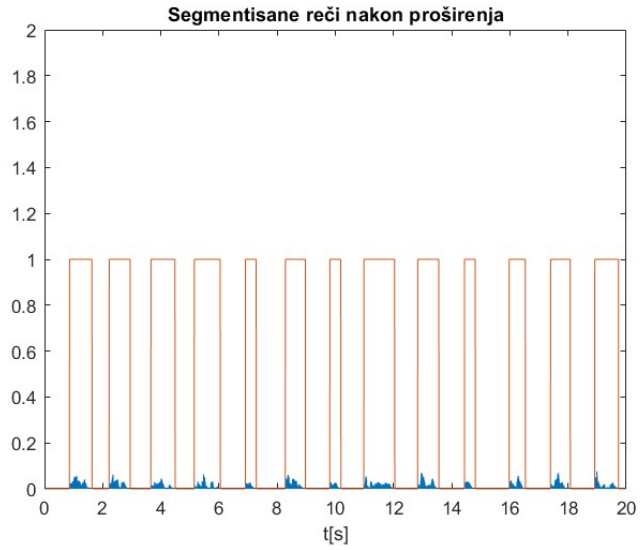


Figure 7: Segmentacija reči

Možemo primetiti da su reči potpuno pravilno segmentisane, što bi se čulo i ako se one zasebno odslušaju.

U narednoj tački zadatka je potrebno da se na osnovu ulazne sekvence nađe pitch frekvencija glasa. Za početak, potrebno je da snimljena sekvenca bude što bogatija samoglasnicima, i da ima minimalno bezvučnog signala kako bi za bilo koju metodu periodičnost signala bila dovoljno velika. Za procenu pitch frekvencije su korišćene metoda paralelnog procesiranja i procena na osnovu autokorelacione funkcije. Takođe je izvršeno i poređenje sa rezultatom dobijenim MATLAB-ovom ugrađenom funkcijom.

Prva korišćena metoda je metoda paralelnog procesiranja. Ideja te metode je da se signal za početak isfiltrira tako da ostane opseg sa mogućim vrednostima pitch frekvencije, što je u konkretnom slučaju $(60, 300)Hz$. Nakon toga se generiše šest povorki impulsa na osnovu signala, koje bi trebalo da imaju istu periodu kao pitch perioda. Te povorke impulsa se dobijaju na sledeći način:

$$\mu_1(i) = max(0, M_i) \tag{1}$$

$$\mu_2(i) = max(0, M_i - M_{i-1}) \tag{2}$$

$$\mu_3(i) = max(0, M_i - m_{i-1}) \tag{3}$$

$$\mu_4(i) = max(0, -m_i) \tag{4}$$

$$\mu_5(i) = max(0, -m_i + m_{i-1}) \tag{5}$$

$$\mu_6(i) = max(0, -m_i + M_{i-1}) \tag{6}$$

Prvih 1000 odbiraka signala i povorki impulsa izgledaju na sledeći način:

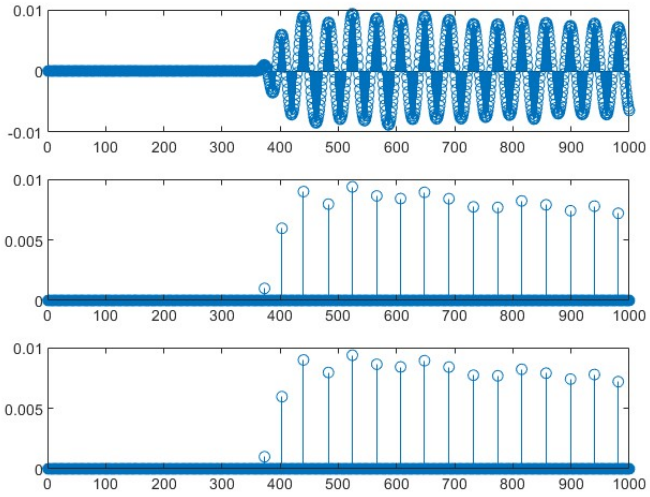


Figure 8: Povorke impulsa

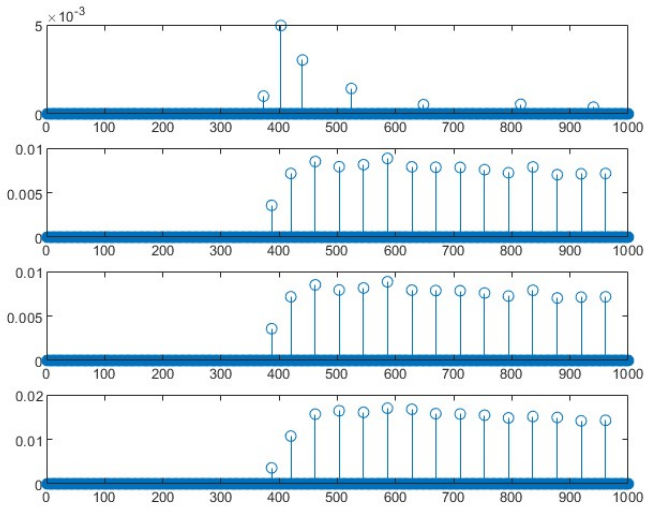

Figure 9: Povorke impulsa

Sada se na osnovu povorki impulsa pravi šest procena pitch perioda. Za početak se postavi blanking time $\tau \in (3.5, 4)ms$. To je vreme od koga je pitch perioda sigurno veća i u kome se ne vrši procena. Nakon toga se posmatra kada se prvi put naišao presek nekog od impulsa i funkcije $Ae^{\lambda(t-\tau)}$, gde $\frac{1}{\lambda} \in (7, 8)ms$. Vreme od početka perioda $\tau$ do dobijenog preseka je procena pitch periode. Konačna procena se dobija kao medijan šest dobijenih procena i u konkretnom slučaju dobijena vrednost pitch frekvencije je $186.0465Hz$, što je potpuno ista vrednost koju daje i ugrađena funkcija. Takođe dobijena vrednost pripada opsegu pitch frekvencija ženskih glasova.

Drugi metod za procenu pitch periode je na osnovu autokorelacione funkcije. Naime autokorelaciona funkcija periodičnog signala je periodična sa istom periodom. Stoga bi bilo očekivano da perioda autokorelacione funkcije odgovara pitch periodi. Pre toga je na signal primenjen 3-level clipping kako bi se otklonio dodatno šum. Na osnovu novog signala je procenjena autokorelaciona funkcija, koja izgleda na sledeći način:

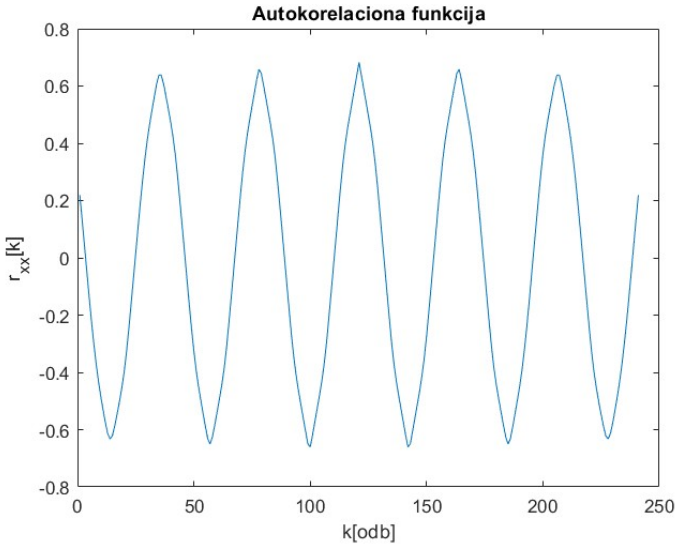


Figure 10: Procena autokorelacione funkcije

Na osnovu periodičnosti autokorelacione funkcije, ponovo je dobijeno da je pitch frekvencija govornika(mene) $186.0465Hz$.

# 2 Zadatak

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 $\mu = 100$ kompanding kvantizator sa 4, 8 i 12 bita i za njih odrediti zavisnost odnosa signal-šum za različite vrednosti odnosa $\frac{X_{max}}{\sigma_x}$. Ovaj odnos menjati promenom varijanse korisnog signala, prostim skaliranjem početne snimljene sekvence. Prikazati rezultate grafički.
3. Isprojektovati Delta kvantizator za sekvencu iz tačke 1. Adekatno podesiti parametar $\Delta$ tako da se dobije što bolji kvalitet kvantizacije. Uporediti oblike originalnog i kvantizovanog signala. Šta se dešava kada je korak kvantizacije $\Delta$ previše mali ili previše veliki? Da li se histogram priraštaja može koristiti za odred ̄ivanje adekvatnog parametra $\Delta$? Pratiti kvalitet zvuka i promene u amplitudi za svaki slučaj.

Nad dobijenom sekvencom je za početak korišćen $\mu$ companding kvantizator. Ideja tog kvantizatora je da razvuče niske vrednosti signala, i zbije visoke, i idealna funkcija za to bi bila logaritamska. Međutim logaritamsku funkciju nije moguće realizovati pošto može da teži beskonačnosti. Stoga je uneta funkcija koja pravi isti efekat kao logaritamska funkcija:

$$F(x) = X_{max} \frac{log(1 + \mu \frac{|x|}{X_{max}})}{log(1 + \mu)} sgn(x) \tag{7}$$

Glavno poboljšanje ovog kvantizatora u odnosu na uniformni je to što odnos signal-šum slabije opada, što znači da odnos signal-šum nije toliko zavisan od amplitude signala. Sa većim $\mu$ raste i konstantnost odnosa signal-šum. Takođe, što je veći broj nivoa kvantizacije to je manja varijansa šuma, a samim tim je i veći odnos signal-šum. Ovo se može videti i na sledećim graficima koji predstavljaju odnos signal-šum za različite brojeve nivoa kvantizacije(bita) i za različite vrednosti $\mu$:

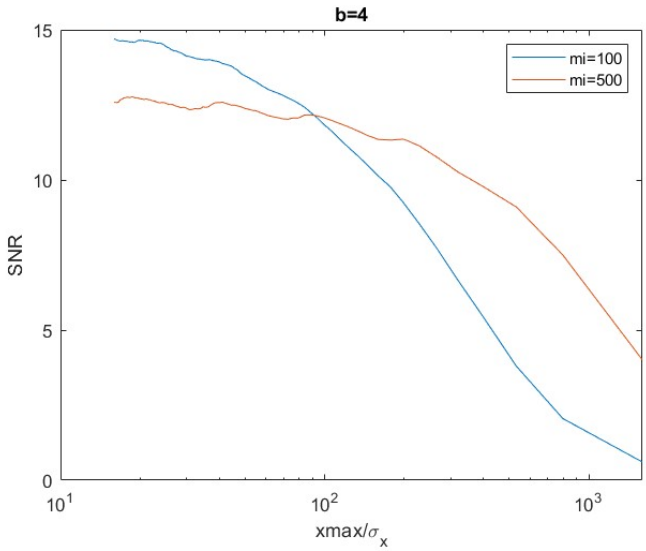


Figure 11: SNR za 4 bita

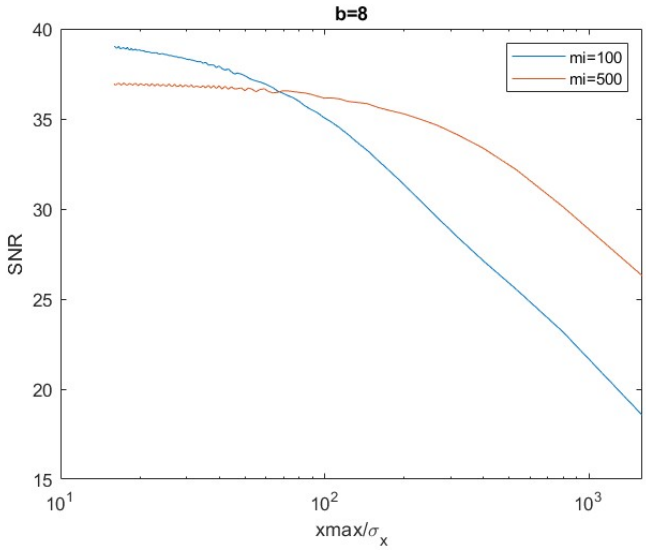


Figure 12: SNR za 8 bita

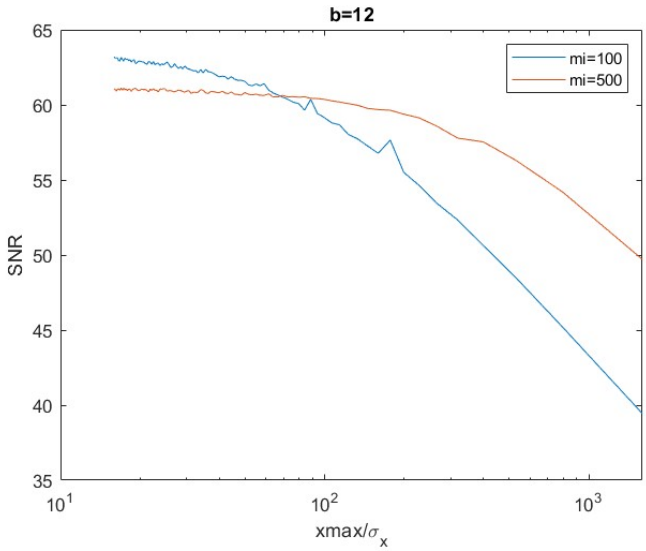


Figure 13: SNR za 12 bita

Naredni korišćeni kvantizator je $\Delta$ kvantizator koji radi po istom principu kao i diferencijalni kvantizator, s tim da radi isključivo sa dva nivoa kvantizacije, $\Delta$ i $-\Delta$, to jest za priraštaj važi:

$$Q(d) = \begin{cases} \Delta & d(n) \geq 0 \\ -\Delta & d(n) < 0 \end{cases} \tag{8}$$

Odatle sledi da binarizacija izgleda na sledeći način:

$$c[n] = \begin{cases} 0 & Q(d) = \Delta \\ 1 & Q(d) = -\Delta \end{cases} \tag{9}$$

Ovako dobijene predikcije signala kao i sam signal su dati na sledećem grafiku:

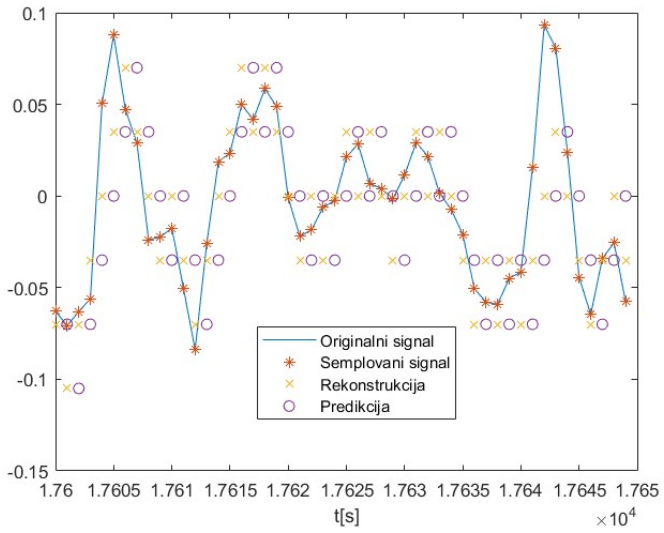


Figure 14: Rezultat delta kvantizatora

Korišćeni korak kvantizacije je 0.035 i ovaj kvantizator može imati dva problema vezana za veličinu koraka kvantizacije i oba su vidljiva na datom grafiku. Prvi mogući problem je slope overload, što znači da je korak kvantizacije premali da bi se ispratio nagib signala, i to se može viseti na grafiku na onim delovima signala sa najvišom amplitudom. Sa druge strane, na onim delovima signala gde je amplituda manja, ili pak signal nema veliki nagib, ovakav korak kvantizacije može da bude preveliki i da estimacija signala osciluje iznad i ispod tačne vrednosti, i takva pojava se naziva grain noise, što se takođe može videti na slici. Ovaj nivo kvantizacije je izabran kao kompromis kako nijedan od neželjenih efekata ne bi bio preizražen, ali bi bolje rešenje svakako bio adaptivni $\Delta$ kvantizator. Vredi napomenuti i da je izdvojeni deo signala sa slike zvučni deo signala, i da bi ovaj korak kvantizacije sigurno bio preveliki za bezvučne delove. Korak kvantizacije bi se mogao i proceniti sa histograma priraštaja kao vrednost koja bi pokrila najveći deo(90%) svih priraštaja. Histogram priraštaja je dat na sledećem grafiku: Ovako procenjeni korak kvantizacije je takođe 0.035.

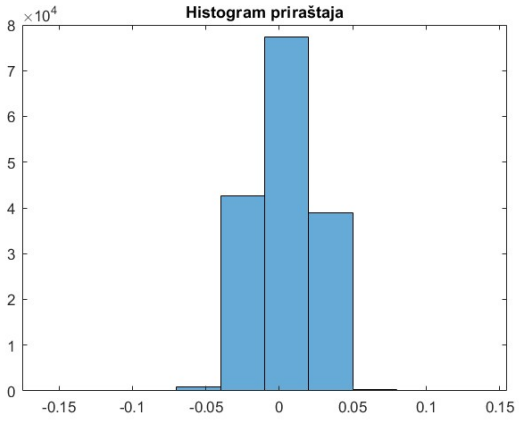


Figure 15: Histogram priraštaja

# 3 Zadatak

Snimi bazu sa 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 (segmentacija i filtriranje).
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 sekvencii prokomentarisati uticaj izbora obeležja i klasifikatora na ishod klasifikacije. Izdvojiti i prokomentarisati primere tačno i pogrešno klasifikovanih sekvenci.

U ovom zadatku su za početak snimljene i očitane sekvence izgovorenih cifara jedan, devet i pet. Nakon toga su svi signali isfiltrirani i segmentisane su reči od šuma. Metoda korišćena za segmentaciju je Rabinerova metoda i dobijena segmentacija za po jedan primer svake cifre izgleda na sledeći način:

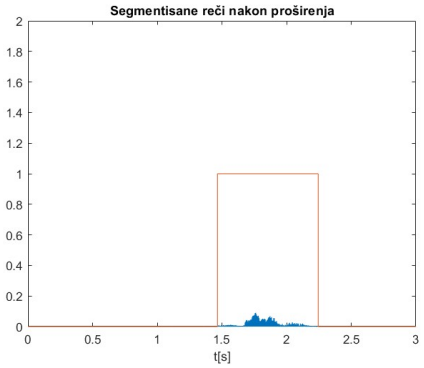


Figure 16: Segmentisana reč devet

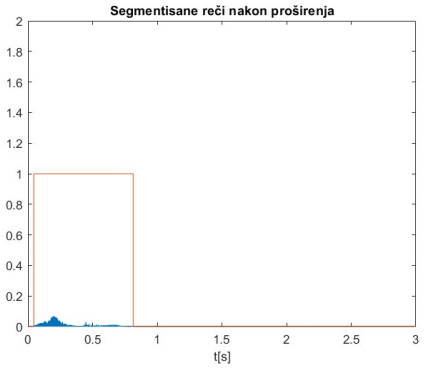


Figure 17: Segmentisana reč jedan

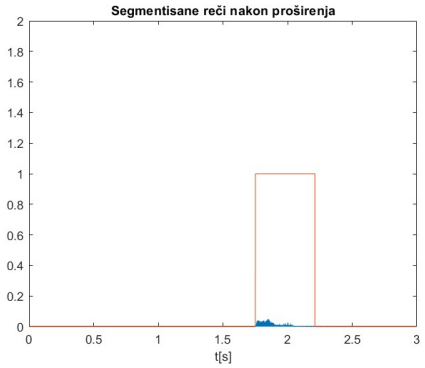


Figure 18: Segmentisana reč pet

Prethodno objašnjeni deo je spadao u pretprocesiranje signala. Nakon toga je za svaku sekvencu nađen adekvatam AR model reda 50. Metod korišćen za pronalaženje parametara AR modela je autokorelacioni metod, za koji je iskorišćena ugrađena funkcija. Naime to je urađeno za prozore veličine 20ms i dobijeni LPC parametri su iskorišćeni za pronalaženje obeležja na osnovu kojih bi mogao da se projektuje klasifikator. Medijan parametara AR modela za sve prozore je sledeći:

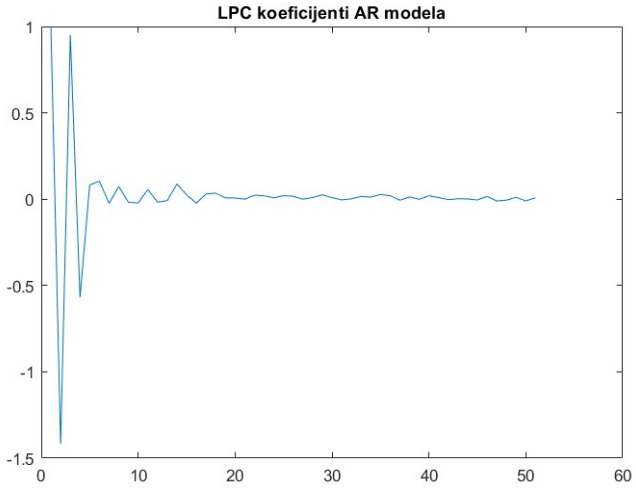


Figure 19: Medijan parametara AR modela reči devet

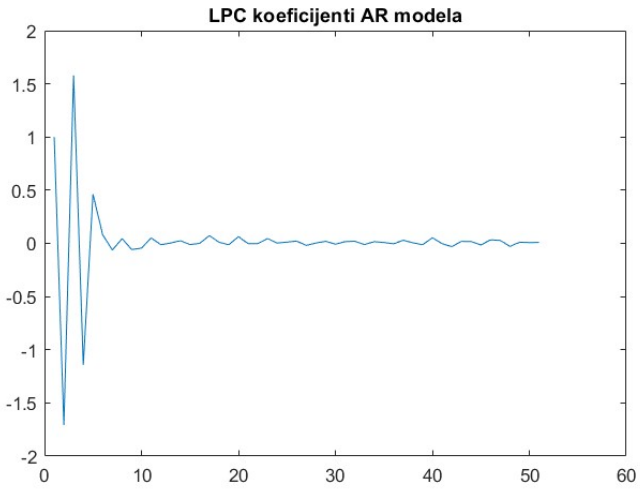


Figure 20: Medijan parametara AR modela reči jedan

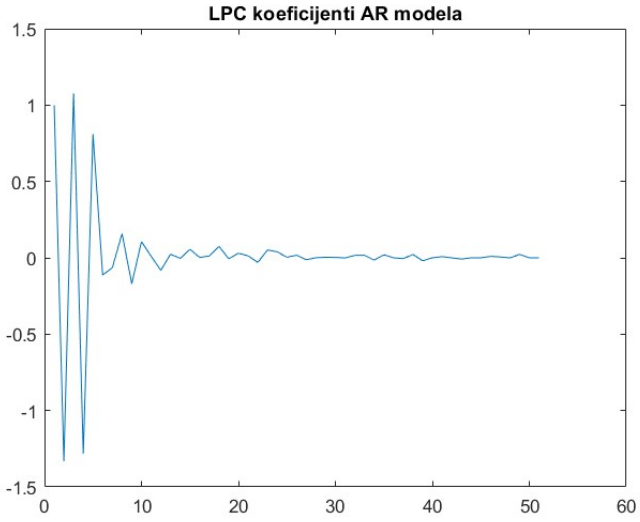


Figure 21: Medijan parametara AR modela reči pet

Na osnovu dobijenih grafika vidimo da za reč pet imamo više LPC parametara koji su po absolutnoj vrednosti veći od 0.5, pa je to korišćeno kao jedno obeležje. Drugo obeležje je 8. LPC parametar, i do toga se došlo redom proveravanjem svih parametara i posmatranjem koji daje najbolju separabilnost.

Konačno je primenjen linearni klasifikator na bazi željenog izlaza, jer su klase koliko toliko linearno separabilne. Takođe je isproban i Bayesov klasifikator i dao je lošije rezultate. Pored toga je data veća težina jednoj od klasa kako bi rezultat bio bolji. Konačan izgled obeležja kao i diskriminacione krive dobijene linearnim klasifikatorom su date na sledećoj slici:

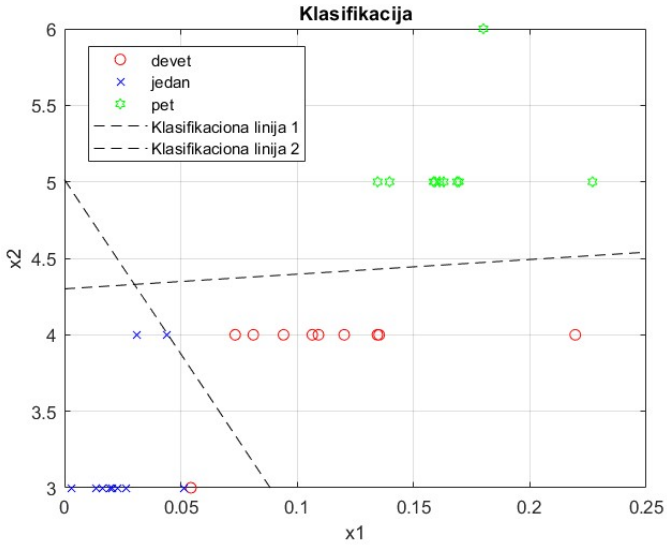


Figure 22: Klasifikacija trening skupa

Dobijena konfuziona matrica trening skupa je sledeća:

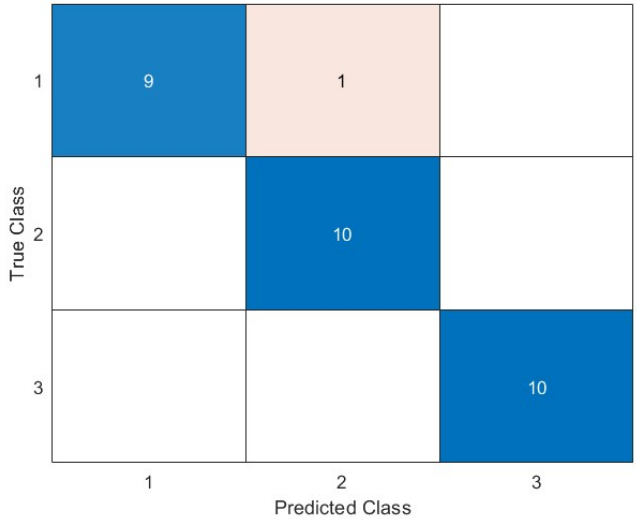


Figure 23: Konfuziona matrica trening skupa

Vidimo da je samo jedna reč devet klasifikovana kao jedan, što je svakako očekivano jer su te dve klase dosta bliže jedna drugoj i njigova segmentacija je teža.

Konačno je snimljeno po 5 novih sekvenci za svaku reč, koje su iskorišćene kao test skup. Te sekvence su prošle kroz isti proces obrade kao i reči trening skupa, i nad njima je primenjen ranije projektovani klasifikator. Dobijeni rezultat je sledeći:

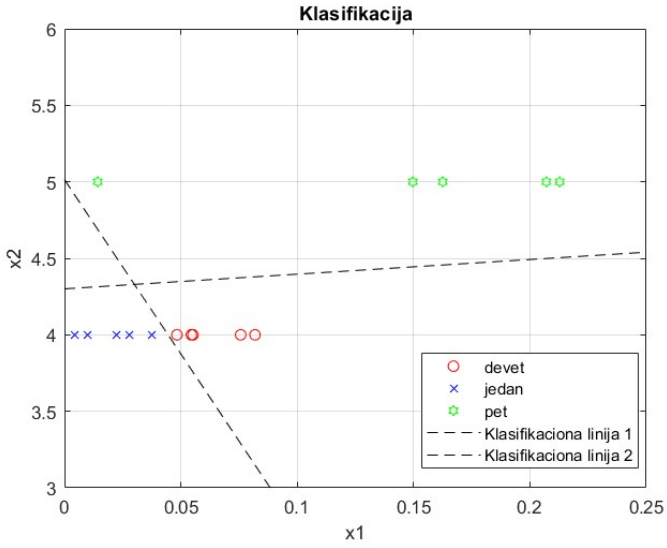


Figure 24: Klasifikacija test skupa

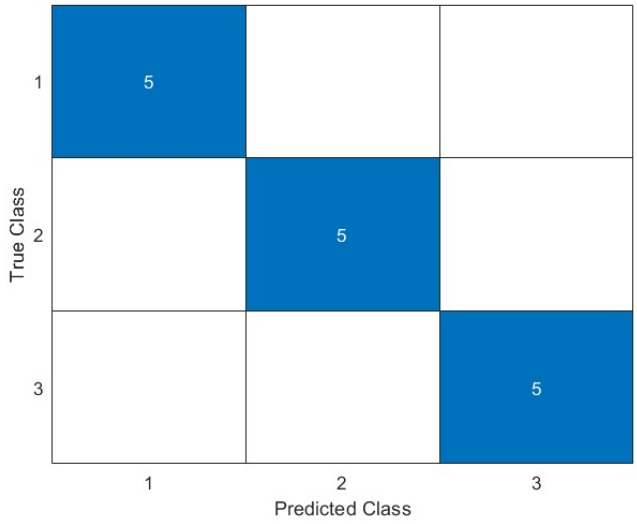


Figure 25: Konfuziona matrica test skupa

Vidmo da je na test skupu 100% tačnost klasifikacije.