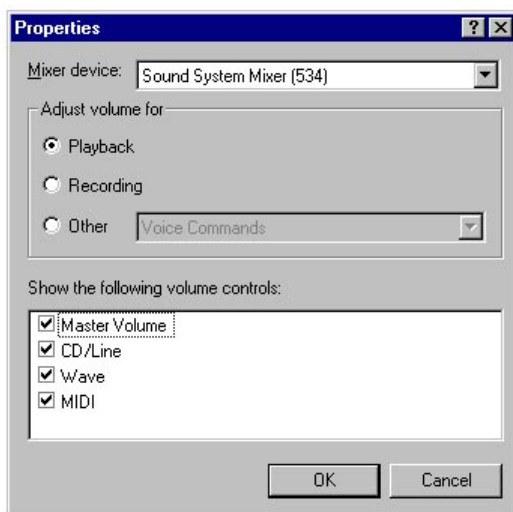
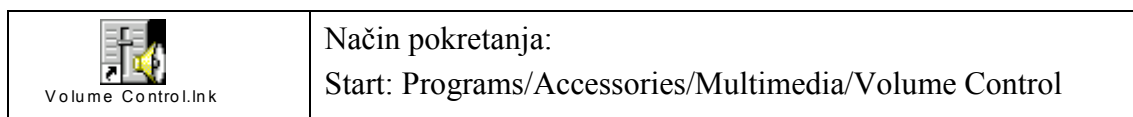


2. SNIMANJE GOVORNIH SIGNALA KORIŠTENJEM ZVUČNIH KARTICA

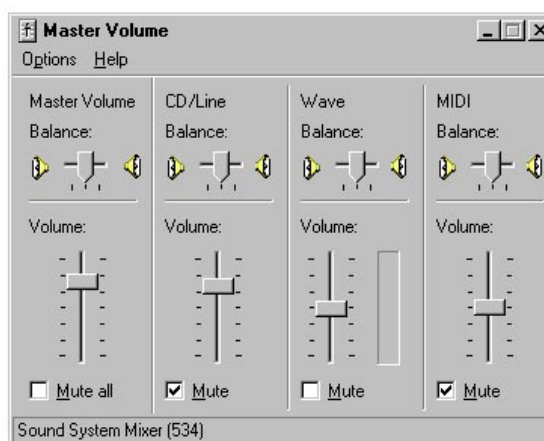
U okviru laboratorijskih vježbi iz digitalne obrade govora provoditi će se raznolike obrade govornog signala. Sve će se obrade provoditi korištenjem programskog paketa Matlab i to izvan realnog vremena. To u stvari znači da će se obrade provoditi na unaprijed snimljenim govornim signalima, a trajanje obrade će redovito biti duže od trajanja samog govornog segmenta koji se obrađuje. Ulazne datoteke u ove obrade koje sadrže snimljene govorne poruke su tipa ".WAV", a u ovoj vježbi će biti pokazano kako je moguće korištenjem standardnih programa na PC računalu opremljenom sa zvučnom karticom snimati odnosno reproducirati govorne datoteke u ".WAV" formatu. Stvarni izgledi tih programa ovise o tipu korištene zvučne kartice, a ilustracije u ovom dokumentu se odnose na karticu koja je ugrađena u PC računala u laboratoriju. U slučaju korištenja modernijih zvučnih kartica sa većim brojem ulaznih i izlaznih kanala i s većim mogućnostima upravljanja sklopovljem, izgled dialog-box-eva se može razlikovati, no osnova ideja koja će biti izložena u nastavku moći će biti primjenjena na bilo kojoj standardnoj 16-bitnoj zvučnoj kartici.

2.1 Inicijalizacija miksera za snimanje i reprodukciju

Da bi bilo moguće uspješno snimati i reproducirati govorni signal primjenom zvučne kartice u PC računalu i pripadnih zvučnika i mikrofona, potrebno je ispravno podesiti mikser koji određuje koji se izvor zvuka reproducira na zvučnicima, odnosno koji se izvor zvuka snima. Program za upravljanje audio-mikserom se naziva "**Volume Control**" i ima ikonu poput ove:



a)

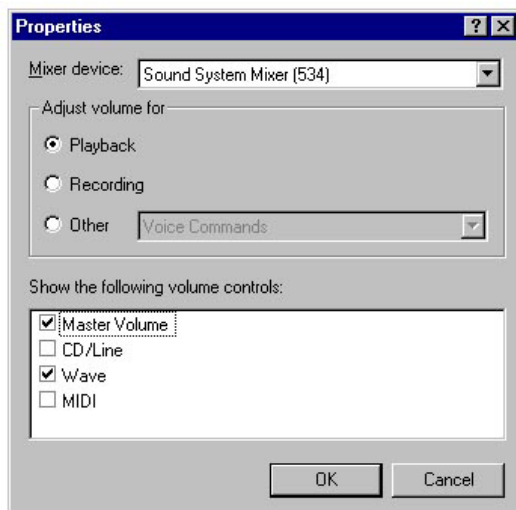


b)

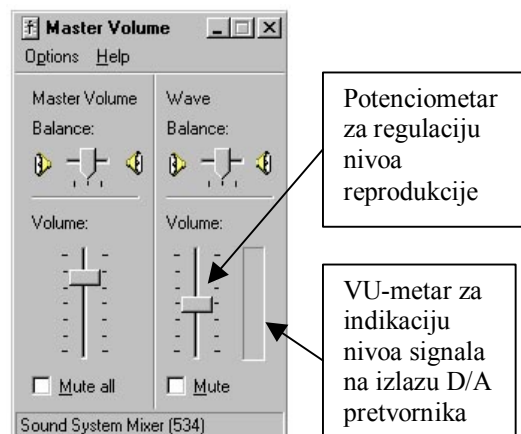
Slika 2.1-1 Dialog box za aktivaciju kontrola za upravljanje svih izvora zvuka za reprodukciju a) i sve kontrole nakon uključivanja b)

Nakon što se program pokrene potrebno je pravilno podesiti parametre. To se postiže ulaskom u Menu: "**Options/Properties**" čime se dobiva dialog-box kao što je prikazan na

slici 2.1-1 a). Potrebno je odabrati **"Adjust volume for : Playback"**, te u donjem dijelu ovog dialog box-a aktivirati prikaz kontrola za sve izvore zvuka u PC-u, kao što je prikazano na slici 2.1-1 a). Nakon što se odabrana konfiguracija potvrdi pritiskom na OK pojavljuje se mikser koji je prikazan na slici 2.1-1 b). U tom mikseru treba deaktivirati sve izvore za reprodukciju osim Wave i Master Volume, što se postiže check-iranjem Mute kontrole u svim preostalim kanalima, kao što se vidi na slici 2.1-1 b). Na taj način će na zvučniku biti reproduciran isključivo zvuk iz Wave device-a, što je u stvari 16-bitni D/A pretvornik koji se nalazi na zvučnoj kartici.



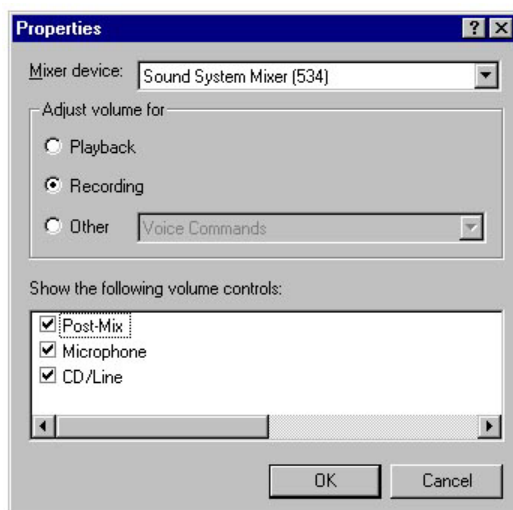
a)



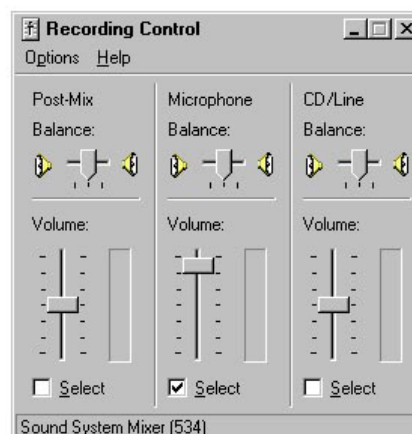
b)

Slika 2.1-2 Dialog box nakon deaktivacije nepotrebnih kontrola a) i dvije preostale kontrole potrebne za upravljanje reprodukcijom b)

Nakon što su nepotrebni izvori deaktivirani, njihova kontrola može biti i izbačena iz prikaza miksera što se postiže ponovnim ulaskom u Menu **"Options/Properties"** i izbacivanjem nepotrebnih ulaza iz liste kontrola, kao što je prikazano na slici 2.1-2 a). Nakon potvrde dobiva se mikser čiji je oblik prikazan na slici 2.1-2 b).



a)

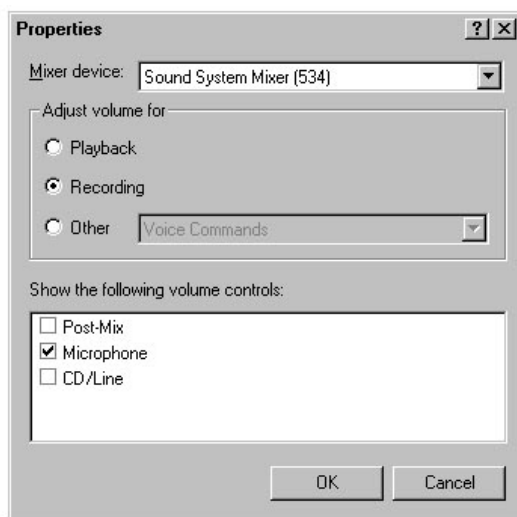


b)

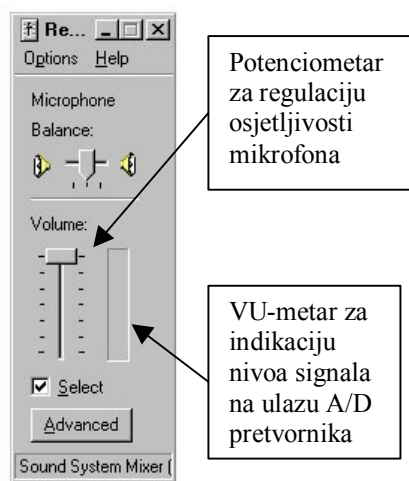
Slika 2.1-3 Dialog box za aktivaciju kontrola za upravljanje svih izvora zvuka prilikom snimanja a) i sve kontrole nakon uključjenja b)

Sličan postupak treba ponoviti i za konfiguriranje miksera koji određuje koji će izvor biti sniman. To se postiže ulaskom u isti Menu: "**Options/Properties**" ali odabirom "**Adjust volume for : Recording**", čime se dobiva dialog-box kao što je prikazan na slici 2.1-3 a).

U donjem dijelu ovog dialog box-a treba aktivirati prikaz kontrola za sve izvore zvuka u PC-u, kao što je prikazano na slici 2.1-3 a). Nakon što se odabrana konfiguracija potvrdi pritiskom na OK pojavljuje se mikser za snimanje koji je prikazan na slici 2.1-3 b). U tom mikseru treba odabrati isključivo mikrofonski ulaz kao izvor signala za snimanje, što se postiže check-iranjem "**Select**" kontrole tog kanala, dok svi ostali kanali moraju biti de-selektirani, kao što se vidi na slici 2.1-3 b). Na taj način će prilikom snimanja na ulaz u A/D pretvornik na zvučnoj kartici biti proslijeđen isključivo zvuk iz mikrofona.



a)



b)

Slika 2.1-4 Dialog box nakon deaktivacije nepotrebnih kontrola a) i preostala kontrola potrebna za upravljanje snimanjem b)



Slika 2.1-5 Dialog box za skokovitu promjenu osjetljivosti mikrofona


Nakon uspješno provedenog konfiguriranja, kontrole ulaza koji se ne koriste mogu biti izbačene iz prikaza miksera za snimanje što se postiže ponovnim ulaskom u Menu **"Options/Properties"** i izbacivanjem nepotrebnih ulaza iz liste kontrola, kao što je prikazano na slici 2.1-4 a). Nakon potvrde dobiva se mikser za snimanje čiji je oblik prikazan na slici 2.1-4 b).

Ispod potenciometra za regulaciju osjetljivosti mikrofona, postoji još jedan dodatni gumb, **"Advanced"** koji se ovdje pojavljuje samo ako je u Menu **"Options"** check-iran **"Advanced Control"**. Pritiskom na ovaj gumb, otvara se novi dialog-box prikazan na slici 2.1-5 kojim je moguće povećati osjetljivost mikrofona sa faktorom 10, tj. 20 dB.

2.2 Postupak snimanja govornog signala

2.2.1 Program za snimanje govornog signala

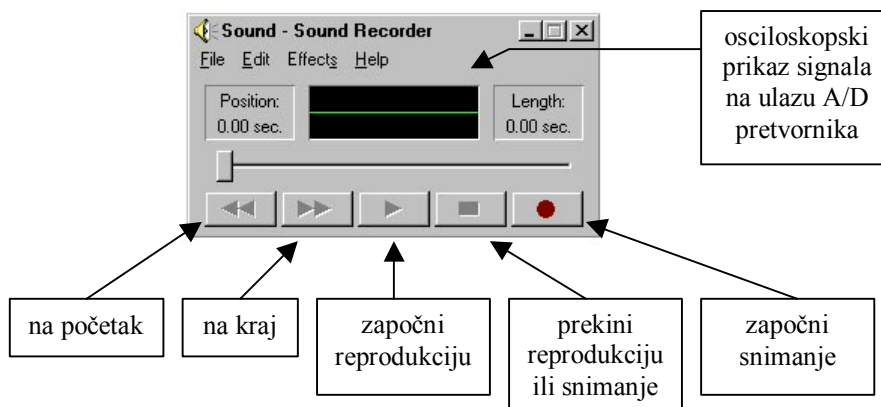
Snimanje govornog signala se provodi korištenjem programa **"Sound Recorder"** čija se ikona i način pokretanja nalaze u slijedećoj tablici:

 Sound Recorder.lnk	Način pokretanja: Start: Programs/Accessories/Multimedia/Sound Recorder
---	--

Sve zvučne datoteke snimljene ovim programom imati će ekstenziju **".WAV"** i ikonu koja je jednaka ovoj prikazanoj u prethodnoj tabeli. Snimanje se provodi digitalizacijom govornog signala primjenom A/D pretvornika na zvučnoj kartici. Digitalizirani uzorci signala pohranjuju se u sirovoj formi u izlaznoj Wav datoteci. Glavi parametri sustava za digitalizaciju su:

- broj ulaznih kanala (1=mono ili 2=stereo)
- rezolucija A/D pretvorbe (8 ili 16 bita po uzorku)
- frekvencija otipkavanja

Prije snimanja govornog signala nužno je potrebno podesiti ove parametre. Nakon pokretanja programa pojavljuje se glavni prozor **"Sound Recorder-a"** prikazan na slici 2.2-1.

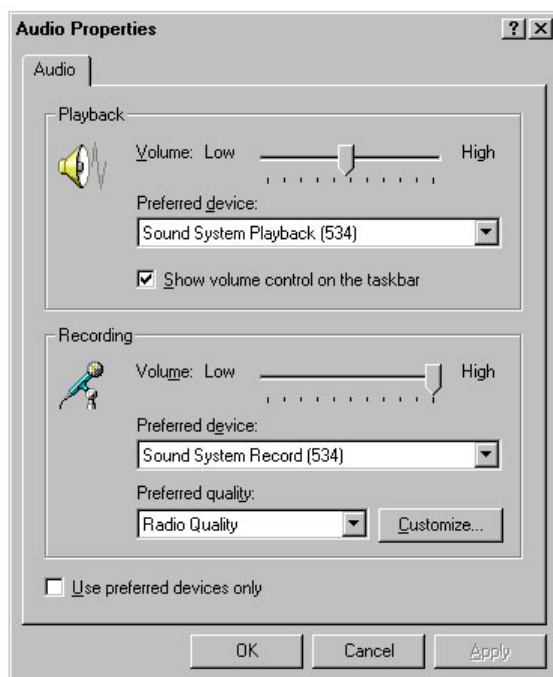


Slika 2.2-1 Izgled programa za snimanje/reprodukciju Wav datoteka

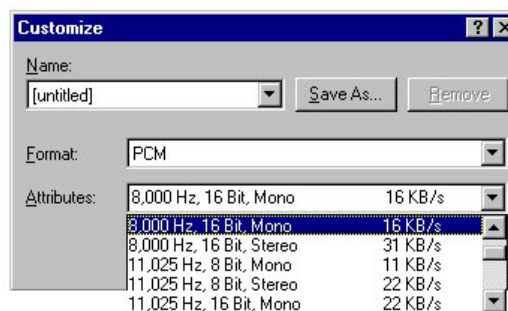
Navedeni glavni parametri sustava za snimanje se podešavaju ulaskom u Menu **"Edit/Audio Properties"** nakon čega se otvara dialog-box prikazan na slici 2.2-2 a). U ovom dialog-box-u je potrebno odabrati podešavanje parametara snimanja pritiskom na tipku

"Customize", nakon čega se ponovno otvara novi dialog box koji je prikazan na slici 2.2-2 b). U njemu se odabire format snimanja "**PCM**", nakon čega je potrebno izabrati i jedan od mogućih "**Atributa**" (konfiguracija parametara), pritiskom na padajući izbornik.

Frekvencija otipkavanja koja će biti korištena prilikom obrade govornog signala u gotovo svim vježbama u okviru ovog predmeta je $f_s=8\text{KHz}$. Obzirom da postoji samo jedan ulazni mikrofoni nema potrebe snimati stereo signal, nego se odabire mono konfiguracija. Rezolucija otipkavanja se podešava na najveću moguću, tj. na 16 bita. Ovaj odabir je prikazan na slici 2.2-2 b).



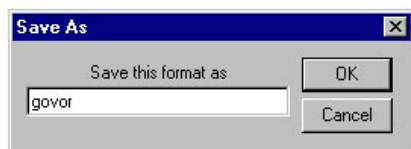
a)



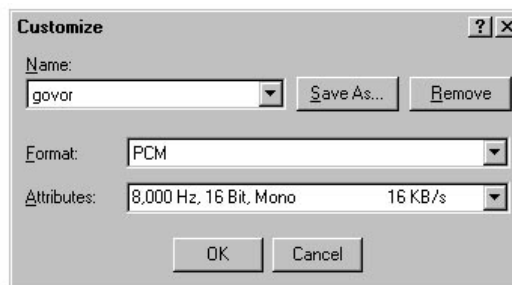
b)

Slika 2.2-2 Podešavanje parametara A/D pretvornika prilikom snimanja

Odabranu konfiguraciju je moguće i pohraniti tako da se naknadno ne moraju ponovno definirati detaljni parametri nego se samo odabere željena konfiguracija. To se postiže pritiskom na tipku "**Save as**" nakon čega program otvara dialog box prikazan na slici 2.2-3 a) u kojem traži ime pod kojim će biti pohranjena ova konfiguracija.



a)

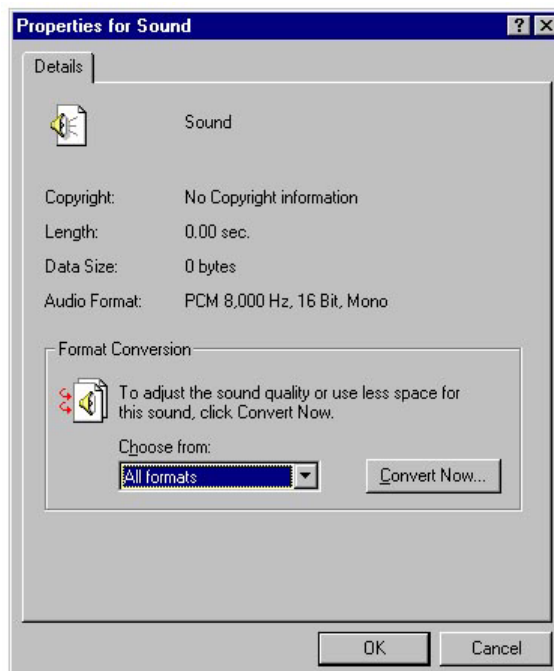


b)

Slika 2.2-3 Pohrana konfiguracije za snimanje govora

Kao naziv ove konfiguracije upišite "**govor**", nakon čega dialog-box sa parametrima snimanja izgleda kao što je prikazano na slici 2.2-3 b). Potvrdite ove promjene pritiskom na OK tipku u dialog-box-u iz slike 2.2-3 b), pa zatim u dialog-box-u iz slike 2.2-2 a). Opisanim

postupkom su podešeni parametri za snimanje govornog signala. Da bi program "Sound Recorder" ove parametre uzeo kao relevantne za prvo slijedeće snimanje, dovoljno je iz glavnog prozora odabrati "File/New". Postavljenu konfiguraciju je moguće provjeriti tako da iz glavnog prozora odaberete "File/Properties" nakon čega se otvara statusni prozor koji je prikazan na slici 2.2-4.



Slika 2.2-4 Statusni prozor s podešenim parametrima snimanja

2.2.2 Ulazna osjetljivost mikrofona

Vrlo je značajno pravilno podesiti osjetljivost mikrofona, odnosno glasnoću izgovora prilikom snimanja. U slučaju da je osjetljivost premala ili da se govori previše tiho, biti će iskorišten samo mali dio ulazne dinamike A/D pretvornika. Tako npr. ako je nominalna ulazna dinamika signala 16-bitnog A/D pretvornika $\pm 1V$ i ako to odgovara digitalnim kodovima od -2^{15} do $2^{15}-1$, tada će uz ulazni napon od $\pm 0.125V$ biti iskorištena samo jedna osmina ukupne dinamike. To znači da će se na izlazu A/D pretvornika pojavljivati samo kodovi od -2^{12} do $2^{12}-1$, tj. kao da se radi o 13 bitnom pretvorniku. Za takav ulazni signal će relativni odnos između energije signala i energije šuma kvantizacije biti $\approx 13 \cdot 6dB = 78 dB$, umjesto očekivanih 96 dB koliko bi bilo za slučaj potpunog iskorištenja ulazne dinamike. Kvantizacijska pogreška nije jedini izvor degradacije snimljenog signala. Drugi izvor su smetnje koje postoje na ulazu u A/D pretvornik, a većinom dolaze od kapacitivnih preslušavanja između zvučne kartice i drugih dijelova u PC računalu. Značajni izvor smetnji je i gradska mreža, zbog koje će se u snimljenom signalu pojaviti signal smetnje frekvencije 50 Hz, ali i svih frekvencija koje odgovaraju njezinim cjelobrojnim višekratnicima. Nivo ovih smetnji je neovisan o nivou signala koji dolazi s mikrofona, tako da je vrlo značajno osigurati što je moguće veći iznos korisnog signala, da bi u odnosu na nivo signala smetnji bio dominantan.

Iz navedenog je vidljivo da je pogodno podesiti osjetljivost na što je moguće veću, no međutim ni to nije potpuna istina. Ako bi zbog prevelike osjetljivosti, ili pak prevelike glasnoće izgovora ulazni napon A/D pretvornika premašio maksimalnu dozvoljenu ulaznu

dinamiku, dolazi do pojave zasićenja (odrezivanja). U slučaju da ulazni napon premašuje naveći dozvoljeni pozitivni ulazni napon (npr. +1V, kojem odgovara digitalni kod $2^{15}-1$), otipkani uzorak bit će predstavljen upravo s tim maksimalnim pozitivnim kodom, neovisno o stvarnoj vrijednosti ulaznog napona. Analogna priča vrijedi i za negativne napone koji prelaze najnegativniju dozvoljenu vrijednost (npr. -1V). Takva operacija zasićenja uzrokuje nelinearno izobličenje govornog signala (odrezivanje vrhova), koje je neželjeno i nepovoljno prilikom analiza, a uzrokuje i čujnu degradaciju u reproduciranom govoru. Zbog toga je prilikom snimanja potrebno osigurati da ulazni signal A/D pretvornika nikad ne premaši maksimalnu dozvoljenu ulaznu dinamiku, tj. da ne dođe do pojave odrezivanja.

2.2.3 Kontrola nivoa signala na ulazu A/D pretvornika prilikom snimanja

U svrhu kontrole nivoa signala na ulazu u A/D pretvornik moguće je koristiti dva indikatora. Jedan se nalazi u crnom prozorčiću programa **"Sound Recorder"** (vidi sliku 2.2-1) i ima ulogu minijaturnog osciloskopa koji je aktivan za vrijeme snimanja. Ako valni oblik signala prilikom snimanja poprima vrijednosti od donjeg do gornjeg ruba ovog prikaza, to znači da je nivo signala prevelik. Drugi indikator je tipa VU-metra i nalazi se kraj potencijometra za podešavanje osjetljivosti mikrofonskog ulaza na mikseru za snimanje (slika 2.1-4 b)). Ovaj VU-metar oponaša mjerač nivoa signala koji se sastoji od više LED dioda. Što je nivo signala viši, pali se sve više i više LED dioda odozdo prema gore. Najniže LED diode su tamno zelene boje, zatim svjetlo zelene, pa zatim žute boje, dok su najviše diode crvene boje.

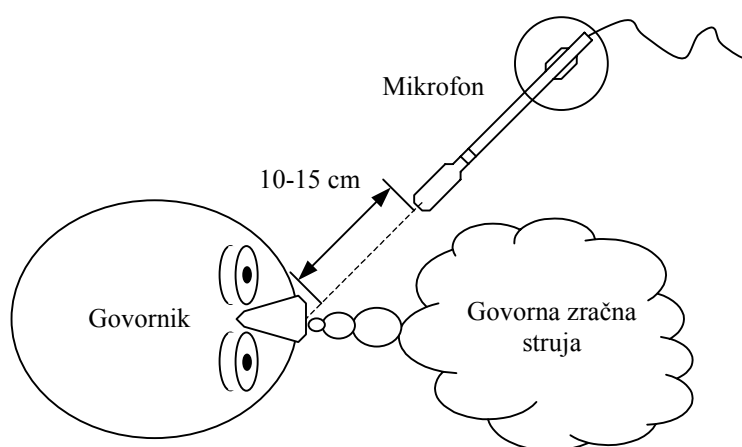
Osjetljivost mikrofona treba tako podesiti da uz normalnu jačinu govora sve zelene, a ponekad i žute diode budu upaljene, ali da se nikad ne pale crvene. Ako je osjetljivost premala, neće se upaliti niti jedna od dioda, a u osciloskopskom prikazu **"Sound Recorder-a"** vidjet će se samo ravna crta. Ovi indikatora su aktivni samo kada je snimanje pokrenuto, što se provodi pritiskom na tipku sa crvenom točkom u glavnom prozoru **"Sound Recorder-a"**. Snimanje se prekida pritiskom na tipku sa kvadratićem. Osjetljivost mikrofona se podešava regulacijom potencijometra za mikrofonski ulaz u mikseru za snimanje (slika 2.1-4 b)), a moguće ju je skokovito podignuti za 20 dB check-iranjem box-a u **"Advanced Controls for Microphone"** dialog box-u (slika 2.1-5).

2.2.4 Postupak reprodukcije snimljenog govornog signala

Nakon što je govor snimljen, on se može reproducirati pritiskom na tipku sa trokutićem. Da bi bilo moguće čuti snimljeni zvuk potrebno je provjeriti da su potencijometri za regulaciju nivoa reprodukcije na pravom iznosu (negdje oko sredine). Ti se potencijometri nalaze u mikseru za reprodukciju (slika 2.1-2 b)) koji je vidljiv tako da se iz menu-a programa **"Volume Control"** odabere **"Options/Playback"**. Oba potencijometra (Master Volume i Wave) moraju biti na pravilnom iznosu, jer inače zvuk iz D/A pretvornika neće doći do zvučnika. Nivo signala na izlazu D/A pretvornika prilikom reprodukcije je također vidljiv na dva indikatora: na osciloskopskom prikazu **"Sound Recorder-a"** (slika 2.2-1) i na VU-metar prikazu koji se nalazi pored potencijometra za regulaciju glasnoće **"Wave"** izlaza u mikseru za reprodukciju (slika 2.1-2 b)). Pored ovih programskih podešavanja potrebno je i provjeriti da su vanjski zvučnici za reprodukciju upaljeni (pod napajanjem), i da potencijometar za regulaciju glasnoće na zvučniku nije utišan na nulu.

2.2.5 Opće preporuke za postupak snimanja

Prije samog snimanja potrebno je mikrofonski usmjeriti prema ustima govornika i postaviti ga na razmak od cca.10-15 cm kao što je prikazano na slici 2.2-5. Za vrijeme snimanja mikrofonski mora stajati mirno na podlozi i ne treba ga držati ili tresti. Prilikom snimanja potrebno je obratiti pažnju na položaj mikrofona u odnosu na usta govornika. Pogodno je da mikrofonski bude što je moguće bliži da bi tako i uz manju podešenu osjetljivost mikrofona bilo postignut dovoljni nivo signala na ulazu A/D pretvornika. Ako je podešena osjetljivost prevelika, tada će mikrofonski osim govora "pokupiti" i sve moguće neželjene akustičke i električke smetnje. Nepoželjno je da se mikrofonski nalazi direktno u struji zraka koja izlazi iz usta, jer će tada mikrofonski "pokupiti" i neželjene "zračne udare", koji se manifestiraju kao neugodni udari u reproduciranom signalu. Zbog toga mikrofonski mora biti postavljen pod kutem u odnosu na centralnu os usta kao što je prikazano na slici 2.2-5, tako da glavna struja zraka prođe mimo njega. Jačina izgovora prilikom snimanja mora biti normala, da bi se dobio što prirodniji govorni signal.



Slika 2.2-5 Pravilan položaj mikrofona prilikom snimanja

Opisani postupak regulacije osjetljivosti mikrofona i njegovog položaja, te podešavanja pravilne jačine izgovora potrebno je provesti višestrukim ponavljanjem snimanja i reprodukcije, dok se ne dobije pravilni nivo signala. Između svaka dva snimanja, stara snimka se treba izbrisati, tako da se iz glavnog prozora "**Sound Recorder-a**" odabere "**File/New**". Snimljeni signal treba biti što je moguće čišći od neželjenih smetnji (šuma, bruma glasa drugih govornika, zvuka iz prostorije itd.)

2.2.6 Završna obrada i pohranjivanje snimljenog govora

Nakon što je uspješno snimljena željena govorna poruka i nakon što je utvrđeno da prilikom reprodukcije ne dolazi do pojave zasićenja (ne pali se crvena LED-dioda u VU-metru miksera za reprodukciju), moguće je pohraniti snimljeni govorni signal u datoteku. Prije ovog pohranjivanja pogodno je iz snimljene poruke izbaciti pauze koje se nalaze na samom početku i na samom kraju, da bi datoteka bila kraća, te da bi njena obrada u Matlab-u bila brža. Da bi odredili točan vremenski trenutak početka aktivnog dijela govorne poruke polagano pomičite scroll-bar "**Sound Recorder-a**" u desno tako dugo dok se u osciloskopskom prikazu signala ne pojavi prva vrijednost različita od nule. Zapamtite vremenski trenutak koji odgovara ovoj poziciji scroll-bar-a (očitajte vrijeme sa brojača lijevo gore od scroll-bar-a) i pokrenite reprodukciju od ove pozicije na dalje. Ako je poruka reproducirana u cijelosti (ne nedostaje ništa na početku), tada je moguće taj početni dio poruke do nađenog vremenskog trenutka izbrisati. Brisanje se provodi tako da se scroll-bar

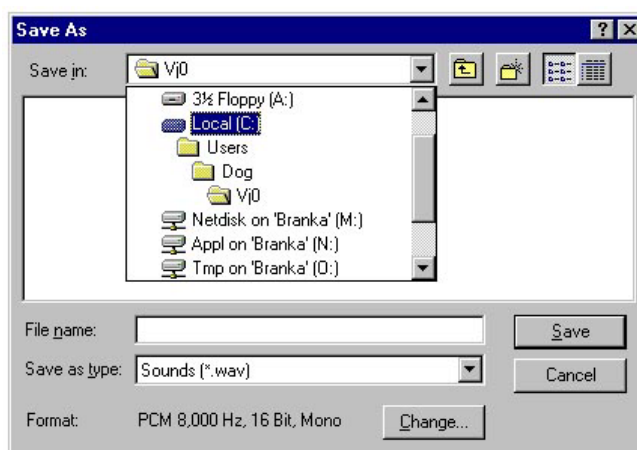
ponovno dovede u nađenu vremensku poziciju početka poruke, te da se iz glavnog prozora odabere **"Edit/Delete Before Current Position"**.

Potpuno analogna operacija se provodi i za brisanje pauze na kraju govorne poruke. Pomaknite scroll-bar na kraj poruke, te ga zatim polagano vraćajte u lijevo tako dugo dok se u prikazu ne pojavi signal različit od nule. Na brojaču očitajte i zapamtite ovaj vremenski trenutak, te pokrenite reprodukciju od nađenog trenutka na dalje. Ako se prilikom reprodukcije čuje isključivo šum, a ne i zadnji glas poruke, tada je ovaj trenutak pravilno odabran i taj dio poruke također može biti izbrisan. Dovedite scroll-bar u zapamćenu poziciju i iz glavnog prozora odabere **"Edit/Delete After Current Position"**.

Na kraju još jednom reproducirajte cijelu poruku i uvjerite se da poruka nije narušena, odnosno da su uklonjene isključivo pauze s početka i kraja.

Snimljena poruka se pohranjuje u datoteku tako da se odabere **"File/Save As"**. Pojavljuje se dialog-box prikazan na slici 2.2-6 u kojem treba odabrati directory i naziv datoteke. Za potrebe laboratorijskih vježbi iz DOG-a biti će korišteni directory C:\USERS\DOG\VJXX, gdje je XX broj koji označava redni broj vježbe. Strogo je zabranjeno pohranjivanje datoteka bilo gdje van ovih directory-ja.

Kao naziv datoteke pogodno je odabrati vlastito prezime, (ili inicijale imena i prezimena) da bi naknadno sve pohranjene datoteke mogle biti korištene u analizama.



Slika 2.2-6 Spremanje snimljene govorne datoteke na disk

3. ANALIZA SVOJSTAVA GOVORNIH SIGNALA U VREMENSKOJ DOMENI

U okviru ove vježbe biti će analizirana svojstva govornog signala u vremenskoj domeni. Biti će pokazano kako se snimljena ".WAV" datoteka učitava u Matlab, tj. prevodi u format u kojem je moguće provesti obrade signala. Biti će ilustrirani i osnovni postupci predobrade govornog signala koji su potrebni radi otklanjanja raznovrsnih smetnji. Biti će pokazano kako izgleda valni oblik govornog signala za različite glasove, te kako je moguće odrediti osnovni period titranja glasnica kod zvučnih glasova.

3.1 Projektiranje filtra za predobradu govornog signala

Prije provođenja analize potrebno je govorni signal filtrirati kako bi se uklonila istosmjerna komponenta signala koja nije zanimljiva i kako bi se frekvencijski pojas signala kojeg analiziramo ograničio na pojas od približno 300 Hz pa do 3500 Hz. Frekvencijske komponente signala na frekvencijama ispod 300 Hz nisu značajne za razumljivost govora, pa se mogu ukloniti. Kod A/D pretvornika koji otipkavaju govorni signal s frekvencijom otipkavanja 8kHz, često u najvišem dijelu frekvencijskog pojasa (3.5kHz do 4kHz) dolazi do pojave aliasing-a (preklapanja spektra), tj. signal u tom frekvencijskom pojasu više nije vjerna slika ulaznog signala. Zbog tog razloga je kao interesantni pojas govornog signala koji se koristi u telekomunikacijskim sustavima odabran pojas od 300Hz do 3.5kHz. U tom pojasu je sadržana gotovo sva informacija koja osigurava potpunu razumljivost govorne poruke i većina informacije o identitetu govornika. U visoko-frekvencijskom pojasu od 3.5kHz do cca. 10kHz, sadržana je dodatna informacija o govorniku, kao i visokofrekvencijske komponente govornog signala koje su posebno izražene kod frikativa (npr. s,š,č ..).

Zahtjevi koji se postavljaju na digitalni filter za pred-filtraciju govora su:

- da propušta pojas od 300Hz do 3.5kHz bez prigušenja (atenuacija <1dB),
- da gušenje u području gušenja filtra bude barem 40dB (100 puta),
- da ima jednu nul-točku na jediničnoj kružnici na frekvenciji 50Hz,
- da prijelazna područja budu čim uža

Nul-točka na frekvenciji 50Hz osigurava maksimalnu attenuaciju za signal smetnje uzrokovan gradskom mrežom. Filter će biti projektiran kao kaskada dva filtra:

- visokopropusnog koji propušta pojas od 300Hz do $f_s/2 = 4\text{kHz}$ i
- niskopropusnog koji propušta pojas od 0Hz do 3.5kHz.

Oba filtra će biti rekurzivnog tipa (engl. Infinite Impulse Response, IIR) i biti će projektirani korištenje Chebyshev II aproksimacije koja osigurava jednoliku valovitost u području gušenja i glatku karakteristiku u području propuštanja. Uz fiksirani zahtijev na iznos gušenja u području gušenja filtera, širina prijelaznih područja zavisiti će o njihovom redu. Pokazuje se da se šestim redom postiže dovoljno usko prijelazno područje.

Funkcija kojom se u Matlabu projektira niskopropusni filter tipa Chebyshev II je:

```
[b, a] = cheby2 (N, R, Wn)
```

pri čemu je N red filtra, R gušenje u dB, a Wn granična frekvencija **područja gušenja** (a ne propuštanja kao što bi se očekivalo) normirana sa polovicom frekvencije otipkavanja. Za visokopropusni filter poziv funkcije je:

```
[b, a] = cheby2 (N, R, Wn, 'high')
```

3.1.1 Projektiranje visokopropusnog filtra

Visokopropusni filter se projektira unošenjem argumenata u Matlab funkciju `cheby2`, a kao rezultat se dobivaju koeficijenti polinoma brojnika i nazivnika IIR filtra:

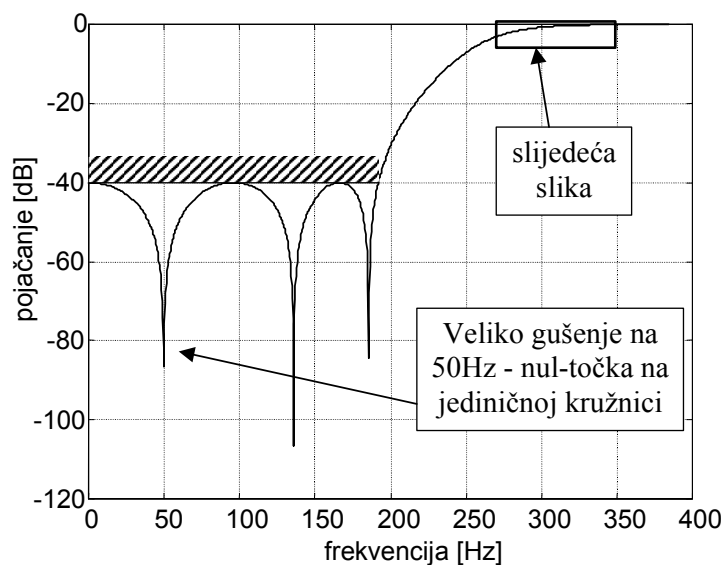
```
fs=8000; % Frekvencija otipkavanja signala
fd=192.84; % Gornja granicna frekvencija podrucja gus. visoko-prop. filtra

ORD=6; % Red oba filtra
R=40; % Iznos gusenja u podrucju gusenja filtra

[bh,ah]=cheby2(ORD,R,fd/(fs/2),'high'); % Projektiraj visokopropusni filt.
```

pri čemu su bh koeficijenti polinoma brojnika, a ah koeficijenti polinoma nazivnika (svaki redak ima 6+1 element). Važno je uočiti da je fd u stvari gornja granična frekvencija područja gušenja, tj. najviša frekvencija kod koje je gušenje filtra još uvijek traženih 40dB. Amplitudno frekvencijska karakteristika filtra u području f_{min} do f_{max} određuje se funkcijom `freqz` i prikazuje u logaritamskom mjerilu korištenjem sljedećih naredbi:

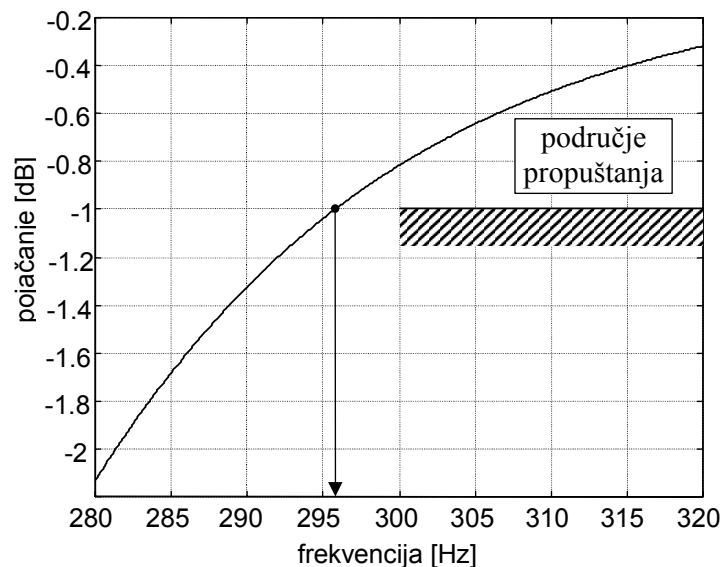
```
f_min=0; % Pocetna frekvencija pojasa
f_max=2*fd; % Zavrсна frekvencija pojasa
[h,om]=freqz(bh,ah,linspace(f_min,f_max,1024)/fs*2*pi);
plot(om/pi*fs/2,20*log10(abs(h)));
title('Podrucje gusenja visoko-prop. filtra');
xlabel('frekvencija [Hz]');
ylabel('pojacanje [dB]');
```



Slika 3.1-1 Frekvencijska karakteristika visokopropusnog filtra 6. reda u području gušenja

Na slici 3.1-1 prikazana je amplitudno frekvencijska karakteristika projektiranog visokopropusnog filtra. Granična frekvencija područja gušenja $fd=192.84$ Hz odabrana je tako da filter ima jedan konjugirano kompleksni par nul-točaka na jediničnoj kružnici na frekvenciji od 50 Hz čime se postiže notch-karakteristika (usko-pojasna brana) vidljiva na slici 3.1-1. To znači da će gušenje osnovnog harmonika smetnji gradske mreže biti gotovo beskonačno. Drugi zahtjev koji mora biti ispunjen je da atenuacija svih frekvencijskih komponenti iznad 300 Hz bude manja od 1 dB. Ispunjenje ovog drugog zahtjeva moguće je

provjeriti ponovnim izračunavanjem karakteristike filtra u pojasu $f_{\min}=280$ Hz do $f_{\max}=320$ Hz korištenjem istih naredbi, čime se dobiva amplitudno frekvencijska karakterisitka na rubu područja propuštanja prikazana na slici 3.1-2.



Slika 3.1-2 Donja granična frekvencija područja propuštanja

Iz slike se može očitati da je donja granična frekvencija područja propuštanja 295.5 Hz što je samo malo niže od traženih 300 Hz. Može se rezimirati da je projektiran visokopropusni filter sa slijedećim pojasima:

- područje gušenja 0 - 192.84 Hz s gušenjem od barem 40 dB,
- prijelazno područje 192.84 do 295.5 Hz,
- područje propuštanja 295.5 do 4000 Hz s gušenjem manjim od 1dB

čime su zadovoljeni svi traženi zahtijevi na visokopropusni filter.

Zadatak 1.: Odrediti frekvencijsku karakteristiku filtra u pojasu 49 do 51 Hz i prikazati je grafički.

3.1.2 Projektiranje niskopropusnog filtra

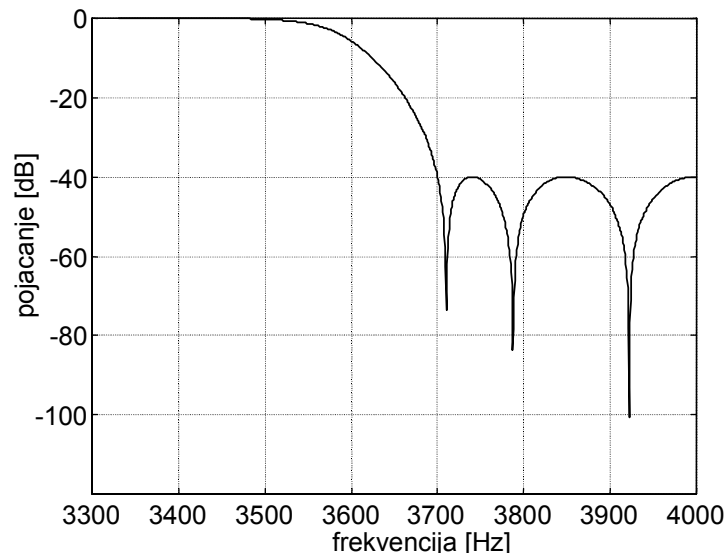
Potpuno analogno opisanome postupku projektira se i nisko-propusni filter čija je uloga uklanjanje visoko-frekvencijskih komponenti spektra. Gušenje u području gušenja filtra također treba iznositi 40 dB.

```
fg=3700; % Donja granicna frekvencija podrucja gusenja nisko-prop. filtra
[b1,a1]=cheby2(ORD,R,fg/(fs/2)); % Projektiraj niskopropusni filter
```

Oprez !! Koeficijenti niskopropusnog filtra su pohranjeni u vektore `b1` i `a1`, tako da prilikom izračunavanja frekvencijske karakteristike filtra treba dotične vektore koristiti kao argumente u funkciji `freqz`. I u ovom slučaju definirana je donja granična frekvencija područja gušenja $f_g=3700$ Hz, a ne kao što bi bilo očekivano, gornja granična frekvencija područja propuštanja. Frekvencijska karakteristika projektiranog niskopropusnog filtra u području gušenja prikazana je na slici 3.1-3 za frekvencijski pojas od $f_{\min}=3300$ Hz do $f_{\max}=4000$ Hz.

Zadatak 2.: Variranjem f_{\min} i f_{\max} , te izračunavanjem frekvencijske karakteristike filtra u tom pojasu odrediti gornju graničnu frekvenciju područja propuštanja, tj. krajnju frekvenciju na kojoj gušenje filtra još uvijek nije veće od 1dB. Na osnovu određene granične frekvencije definirati područje propuštanja, prijelazno područje i područje gušenja filtra, kao što je to bilo učinjeno za visokopropusni filter.

Zadatak 3.: Variranjem donje granične frekvencije f_g , pokušati odrediti iznos f_g uz koji će gornja granična frekvencija područja propuštanja biti točno jednaka 3500 Hz



Slika 3.1-3 Frekvencijska karakteristika niskopropusnog filtra 6. reda s gušenjem od 40dB

3.1.3 Kaskadiranje NP i VP filtra

Filtri dobiveni na prije opisan način mogu se spojiti u kaskadu čime se ostvaruje željeni pojasnopropusni filter. Koeficijenti tog kaskadiranog filtra dobivaju se tako da se pomnože polinomi u brojniku oba filtra, kao i polinomi u nazivniku. Koeficijenti produkta dva polinoma odgovaraju linearnoj konvoluciji koeficijenata svakog pojedinog filtra. U Matlabu se to obavlja funkcijom konvolucije (`conv`):

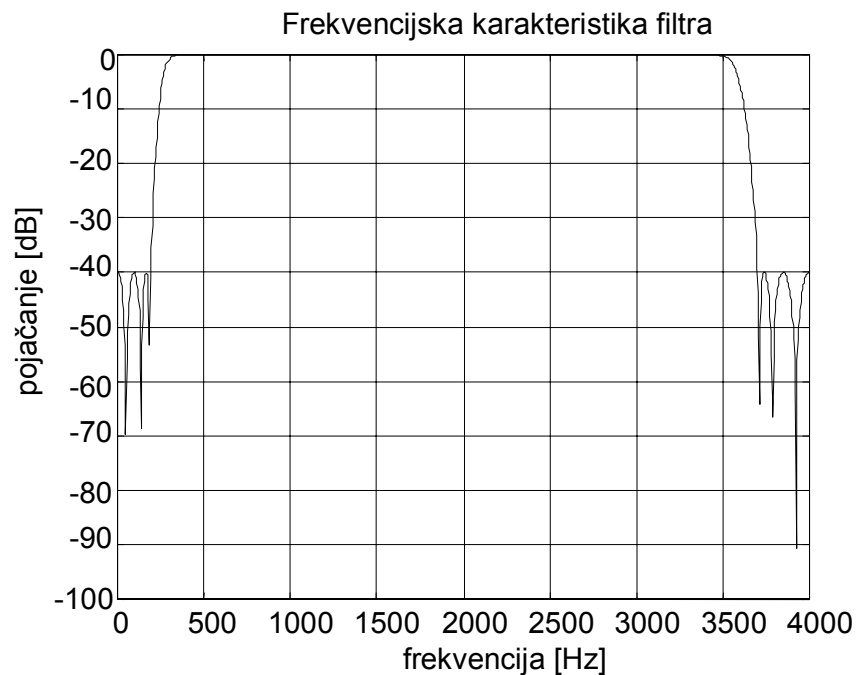
```
b=conv(bh,b1); % Koeficijenti brojnika kaskade
a=conv(ah,a1); % Koeficijenti nazivnika kaskade

% Frekvencijska karakteristika kaskade
[h,om]=freqz(b,a,1024);

plot(om/pi*fs/2,20*log10(abs(h)));
title('Frekvencijska karakteristika filtra');
xlabel('frekvencija [Hz]');
ylabel('pojacanje [dB]');
grid;
pause;
```

pri čemu su b_h i b_1 brojnici, a a_h i a_1 nazivnici VP i NP filtra. Slika 3.1-4 prikazuje konačni izgled pojasnopropusnog filtra 6. reda s gušenjem od 40dB.

Zadatak 4.: Odrediti frekvencijske karakteristike kaskade u prijelaznim područjima i usporediti ih s pripadnim karakteristikama svakog pojedinog filtra (NP i VP). Zbog čega su prijelazne karakteristike ostale nepromijenjene ?



Slika 3.1-4 Frekvencijska karakteristika pojasnopropusnog filtra 12. reda s gušenjem od 40dB dobiven kaskadiraњem NP i VP filtra

3.1.4 Učitavanje snimljene Wav datoteke u Matlab, te njena reprodukcija

U svrhu obrade, snimljenu govornu sekvencu u Wav formatu potrebno je učitati u memoriju, tako da se u Matlab-u pojavi kao nova varijabla. To se provodi funkcijom "wavread" kao što je prikazano u slijedećem primjeru:

```
[y,fs,bits]=wavread('subota');
```

Argument funkciji je naziv Wav datoteke. Nije potrebno dodati ekstenziju ".wav", jer se inherentno pretpostavlja da se radi o datotekama upravo tog tipa. Naziv datoteke mora biti pod jednostrukim navodnicima i može uključivati kompleti put do dotične datoteke. Uzorci signala se pohranjuju u stupčasti vektor y čija je dimenzija određena ukupnim brojem uzoraka. U slučaju da je format snimanja signala bio stereo, tada će varijabla y biti dvostupčani vektor. Ova funkcija provodi automatsku normalizaciju, tj. podaci će biti skalirani tako da maksimalna dinamika A/D pretvornika odgovara rasponu realnih brojeva od -1 do 1 . U varijabli fs će se nalaziti informacija o frekvenciji otipkavanja, dok će u varijabli $bits$ biti informacija o rezoluciji A/D pretvorbe, tj. br. bita po uzorku signala. Učitani signal se može prikazati naredbom plot, dok je njegova reprodukcija na audio kartici moguća primjenom naredbe soundsc, kao što je prikazano u slijedećem primjeru:

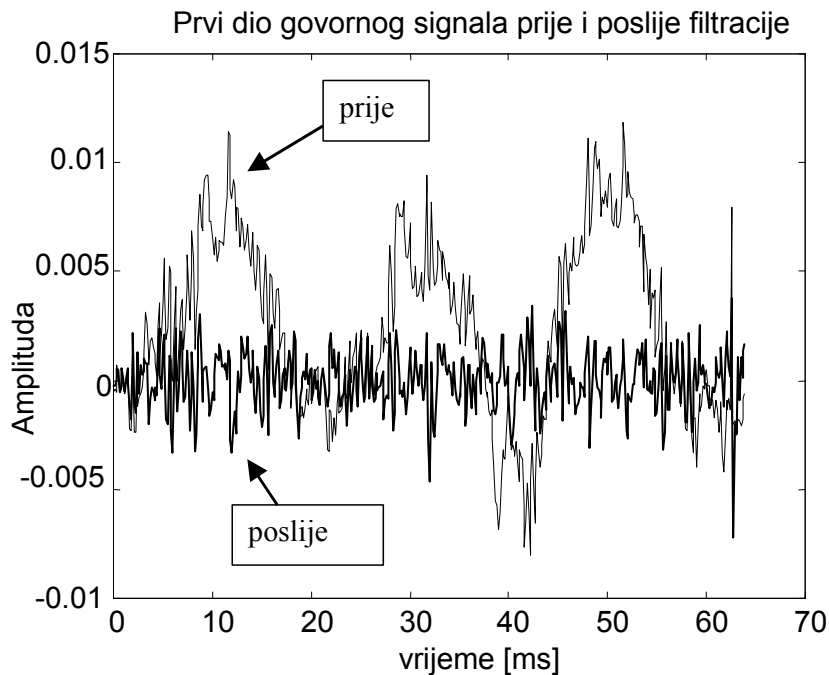
```
soundsc(y,fs)
```

3.1.5 Filtriranje govornog signala

U nastavku će učitani signal biti filtriran primjenom upravo projektiranog pojasnopropusnog filtra. U tu svrhu koristi se Matlabova funkcija za filtraciju:

```
yf=filter(b,a,y);
```

pri čemu su b i a koeficijenti pojasnopropusnog filtra, a y originalni nefiltrirani govorni signal. Izlazni filtrirani signal biti će pohranjen u varijabli yf . Na slici 3.1-5 vidi se rezultat filtracije.



Slika 3.1-5 Utjecaj filtracije na govorni signal

Primjetno je da je gušenje filtra na 50Hz vrlo veliko, pa je time odstranjena smetnja nastala zbog utjecaja gradske mreže na snimljeni govorni signal. Na slici 3.1-5 je istovremeno prikazan filtrirani i nefiltrirani signal, pa se lijepo vidi taj utjecaj.

Zadatak 5.: Naizmjenice primjenom funkcije `soundsc` reproducirati originalni signal y i filtrirani signal yf . Da li čujete kakvu razliku između ova dva signala?

3.2 Vremenska analiza filtriranog signala

U nastavku vježbe biti će provedena analiza valnog oblika govornog signala za neke karakteristične glasove. U tu svrhu koristimo se Matlab programom "pri.m" koji se nalazi u radnom directory-ju ove vježbe. Ovaj program omogućava izdvajanje pojedinih segmenata (glasova) iz snimljene Wav datoteke, njihovu reprodukciju, njihov detaljan prikaz u vremenskoj i spektralnoj domeni, kao i identifikaciju osnovnog perioda glasa kod zvučnih glasova. U prvom dijelu ovog programa provodi se projektiranje pojasno-propusnog filtra, zatim učitavanje wav datoteke "subota.wav", te konačno njena filtracija primjenom projektiranog filtra, kao što je to već detaljno objašnjeno u prvom dijelu vježbe. Pokrenuti ovaj program, unašanjem njegovog imena u komandnom prozoru Matlaba.

Zadatak 6.: Nakon što program usporedno prikaže originalni i filtrirani signal u vremenskoj domeni, na sljedećem prikazu su ta dva signala prikazana u frekvencijskoj domeni. Pohraniti ovu sliku i diskutirati područja gušenja filtra, te kako se ona manifestiraju u prikazanim spektrima.

3.2.1 Odabir glasa za analizu

Ispitivati periodičnost glasa ima smisla samo za zvučne glasove. Ta periodičnost je posebno izražena kod samoglasnika koji pored periodičnosti imaju i relativno veliku energiju. Pri izgovoru samoglasnika glasnice titraju nekom, manje-više, konstantnom frekvencijom koja se naziva osnovnom frekvencijom titranja glasnica i koja značajno ovisi o govorniku. Period osnovne frekvencije titranja glasnica se u engleskoj literaturi naziva pitch period i izražava se ili u milisekundama ili pak u broju uzoraka otipkanog signala.

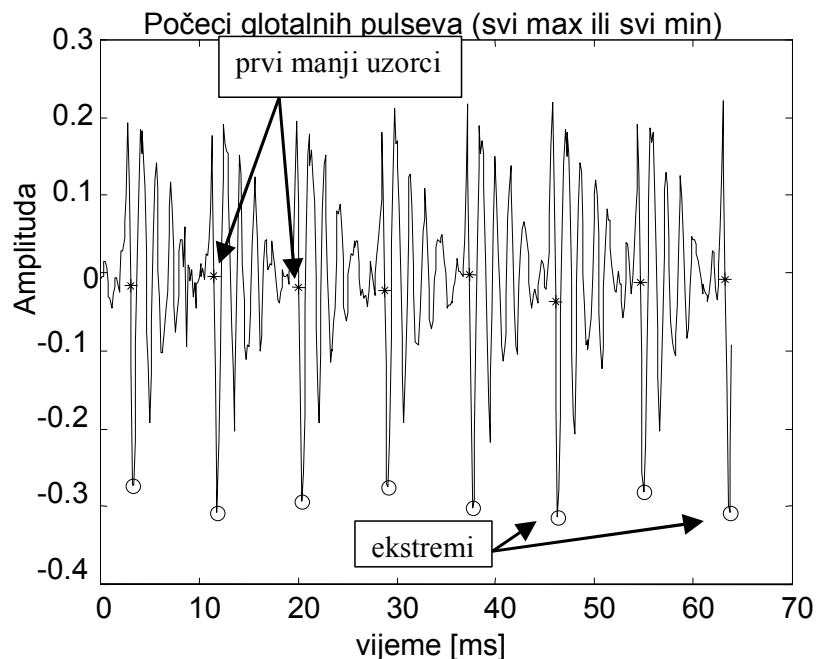
Tokom izvođenja programa "pri.m", se nakon filtracije ulaznog signala bira glas koji će se analizirati. U tu svrhu se pojavljuju se dva prozora (**jedan ispod drugog, pa se drugi na prvi pogled ne uočava!**), te ih je potrebno razmaknuti. Odabir željenog mjesta (segmenta) analize se provodi tako da se mišem klikne na grafu u prozoru broj 1. Odabrani segment signala će automatski biti reproduciran na audio kartici i zvučnicima i to nekoliko puta za redom što služi za slušnu identifikaciju odabranog segmenta. Svaki put kada kliknemo na željeno mjesto pored reprodukcije na zvučnicima će u prozoru broj 2 će biti prikazan odabrani segment. Veličina promatranog segmenta definirana je varijablom N u samom programu "pri.m". Na opisani način moguće je u prozoru broj 2 dobiti detaljniji prikaz odabranog segmenta. Konačni odabir "zaključujemo" klikom na GOTOVO u prozoru broj 1.

Zadatak 7.: Gore opisanim postupkom odrediti segment govornog signala koji odgovara glasu "a" na kraju izgovora riječi "subota". Prikazani valni oblik odabranog segmenta pohraniti u Word datoteku sa rezultatima vježbe.

U nastavku će biti opisano kako se za odabrani segment glasa "a" provodi identifikacija pojedinih perioda titranja glasnica. Nakon finalnog odabira segmenta, u prozoru broj 2 je prikazan odabrani segment i program čeka da se mišem označe počeci pojedinih glotalnih pulseva. U tu svrhu potrebno je prvo vizualno ocijeniti karakteristične točke u valnom obliku signala koje dobro označavaju rub perioda. Za neke segmente i glasove će maksimumi biti izraženiji, dok za neke druge minimumi. Radi veće pouzdanosti izdvajanja osnovnog perioda, potrebno je odabrati one ekstreme koji su izraženiji u odabranom segmentu. **VRLO JE VAŽNO ODABRATI ILI SAMO MAKSIMUME ILI SAMO MINIMUME, tj. NIKAKO NE MJEŠATI MALO JEDNE; PA MALO DRUGE UNUTAR ISTOG SEGMENTA.** Odabir pozicije ekstrema se vrši klikom miša u blizini ekstrema. Pri tome ne treba biti izrazito precizan, jer će program automatski odrediti lokalni maksimum (ili minimum) u okolini točke gdje se kliknuli. Važno je međutim da te ekstreme označavate uvijek s lijeva na desno, tj. u rastućem smjeru vremenske osi. Nakon što ste označili sve ekstreme u odabranom segmentu pritisne se *Enter* i pojaviti će se u istom prozoru ekstremi označeni zelenim kružićima, kao i pozicija prvog uzorka prije tog ekstrema koji ima isti predznak kao i sam ekstrem (označen crvenom zvjezdicom). Primjer tako izoliranih glotalnih pulseva prikazan je na slici 3.2-1 za slučaj kada su označeni minimumi signala. U tom slučaju * označava poziciju prvog uzorka signala prije ekstrema koji ima negativni predznak. Jednim glotalnim pulsom podrazumjevat ćemo dio signala između takve dvije zvijezdice.

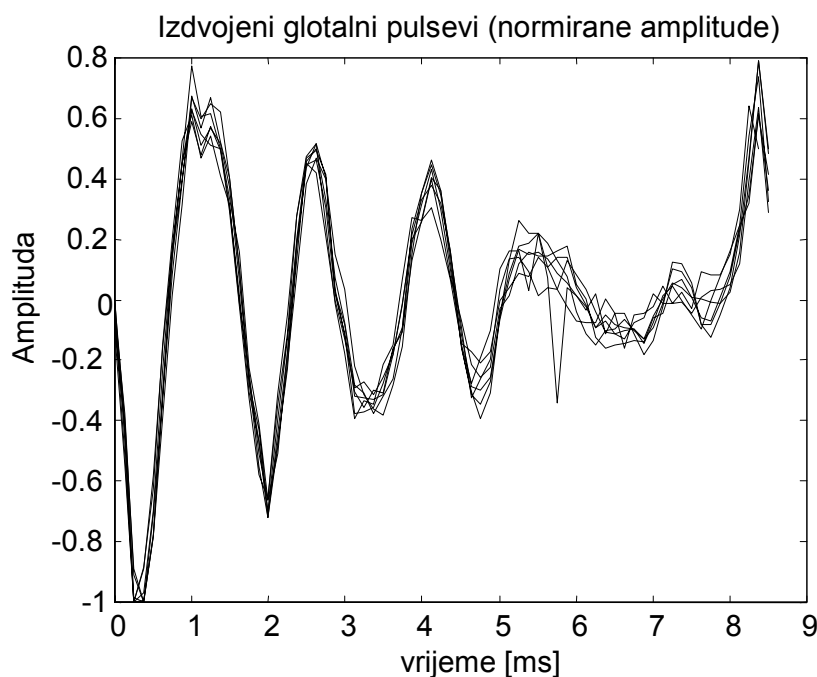
Interesantno je izdvojene periode prikazati usporedno (jedan preko drugog), da bi se uistinu mogla ocijeniti njihova periodičnost. Obzirom da se amplituda signala mijenja od perioda do perioda, svi će periodi prije tog usporednog prikaza biti normalizirani po

amplitudi, tako da odabarni ekstrem kojeg ste kliknuli prilikom izdvajanja perioda ima jediničnu amplitudu za sve odabrane periode.



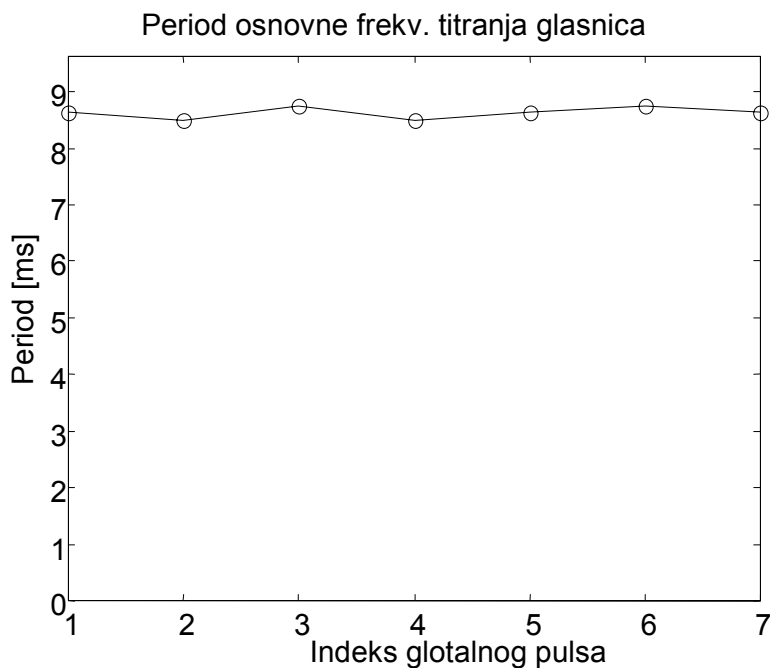
Slika 3.2-1 Označeni počeci glotalnih pulseva

U nastavku izvođenja programa "pri.m", svi izdvojeni glotalni pulsevi odnosno njihovi periodi prikazuju se u prozoru broj 3, pri čemu im se normira amplituda kao što je diskutirano (slika 3.2-2). Primjetno je da se pulsevi vrlo dobro poklapaju osim pri kraju perioda gdje postoje varijacije prvenstveno zbog varijacija u periodu (trajanju) samih pulseva.

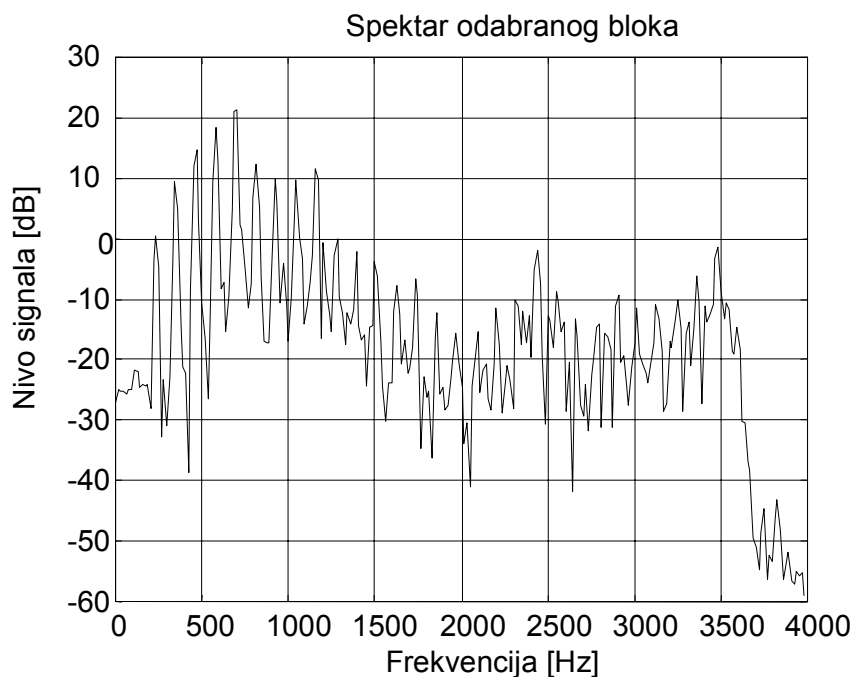


Slika 3.2-2 Izdvojeni glotalni pulsevi za glas "a"

Na slici 3.2-2 se vidjela prilično dobra periodičnost za glas “a”, pa bi bilo zanimljivo vidjeti i koliki je točni period tih glotalnih pulseva i da li su oni uistinu jednoliki. Period je određen kao obična razlika vremenskih indeksa dvije susjedne zvjezdice, tj. razmak dvaju početaka susjednih glotalnih pulseva. Na slici 3.2-3 prikazana je navedena razlika za sve susjedne parove zvjezdica u odabranom segmetu signala.



Slika 3.2-3 Period osnovne frekv. titranja glasnica (za svaki glotalni puls u analiziranom segmetu)



Slika 3.2-4 Spektar govornog signala

Iz slike 3.2-3 je vidljivo da su varijacije pitch perioda male, a da prosječni period iznosi oko 8.6 ms. Tome odgovara osnovna frekvencija titranja glasnica od 116 Hz.

Kao uvod u slijedeću vježbu, na kraju će program "pri.m" prikazati spektar odabranog segmenta govornog signala. Prije primjene diskretne Fourier-ove transformacije, odabrani segment signala je pomnožen s Hamming-ovim vremenskim otvorom u svrhu točnije spektralne estimacije. Prime spektra glasa "a" prikazan je na slici 3.2-4. Primjetno je da se neke frekvencije izdižu u odnosu na druge i da će one karakterizirati pojedini vokal.

Zadatak 8.

Opisanu analizu primjenom programa "pri.m" ponoviti na Wav datoteci sa samoglasnicima koju ste snimili tokom prve vježbe. Analzu provesi za samoglasnik "a".

Da bi se mogao analizirati vlastiti primjer, potrebno je izvorni M-file promjeniti da bi se učitao novi wave-file. U tu svrhu potrebno je samo promjeniti jedan redak u "pri.m" primjenom notepad-a ili nekog drugog tekstualog editora:

```
[y,fs,bits]=wavread('subota'); % Originalna komanda
```

u oblik:

```
[y,fs,bits]=wavread('samo_ime'); % Snimljeni file s vasim samoglasnicima
```

pri čemu je 'samo_ime' ime osobnog snimljenog primjera u vježbi 1 (uključiti i put do navedene datoteke). Nakon promjene pokreće se M-file utipkavanjem pri (bez točke).

Prilikom analize treba snimati dobivane grafove na način kako je već opisano i to poput onih na slikama 3.2-1, 3.2-2, 3.2-3 i 3.2-4. Snimljene grafove treba spremi (prebaciti) u Word datoteku sa rezultatima ove laboratorijske vježbe gdje ih se malo i smanji kako bi stala dva grafa na jednu stranicu. Potrebno dodajte i kratke komentare uz slike.

3.2.2 Utjecaj varijacija osnovne frekvencije glasnica**Zadatak 9.**

Za primjer treba snimiti glas "a", ali ovaj put sa silaznom intonacijom (visina glasa se spušta prema kraju izgovora). Kakva će biti periodičnost osnovne frekvencije titranja glasnica ako se mijenja visina glasa? Koliki je period titranja?

Postupak je isti kao i u prethodnom zadatku.

4. SPEKTRALNA ANALIZA GOVORA I SPEKTROGRAMI

4.1 Uvod u spektralnu analizu

Govorni signal je kvazi-stacionarni signal, tj. signal kod kojeg postoji stacionarnost ali samo na dovoljno kratkim odsječcima signala. Jedno od posebno važnih statističkih svojstava signala su momenti drugog reda, koji su direktno vezani sa spektrom snage signala. Spektar snage je moguće estimirati primjenom periodograma, no da bi ta estimacija imala smisla i dala rješenje koje ima fizikalno značenje, statistička svojstva analiziranog signala moraju biti stalna. Radi toga, umjesto standardne Fourierove transformacije koja transformira vremenski diskretni signal beskonačnog trajanja u pripadni periodični spektar, kod obrade govora se uobičajeno koristi vremenski kratkotrajna Fourierova transformacija. Njena glavna značajka jest da se spektralna analiza provodi nad kratkim odsječcima signala kod kojih je zadovoljen uvjet stacionarnosti. Izdvajanje takovih odsječaka se provodi primjenom vremenskih otvora, koji su različiti od nule samo za konačan skup indeksa. Množenjem ulaznog signala s takvim vremenskim otvorom pomaknutim na željeno mjesto u signalu, dobiva se novi signal također beskonačnog trajanja, ali čiji su uzorci jednaki nuli za sve indekse signala koji se nalaze izvan primijenjenog vremenskog otvora.

$$x_n(m) = x(n+m) \cdot w(m) \quad \begin{cases} w(m) \neq 0 & \text{za } 0 \leq m < N \\ w(m) = 0 & \text{za } m < 0 \text{ ili } m \geq N \end{cases} \quad (4.1-1)$$

U gornjem izrazu se primjenom otvora $w(m)$ koji ima N uzoraka izdvaja segment ulaznog signala od indeksa n do indeksa $n+N-1$. Izdvojenom segmentu signala $x_n(m)$ dodajemo indeks n da naglasimo da se radi o segmentu čiji se prvi uzorak nalazi na indeksu n . Nakon toga se na tom segmentu primjenjuje Fourierova transformacija prema izrazu:

$$X_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} x_n(m) \cdot e^{-j\omega m} = \sum_{m=0}^{N-1} x_n(m) \cdot e^{-j\omega m} \quad (4.1-2)$$

Važno je naglasiti da je Fourierova transformacija $X_n(e^{j\omega})$ izdvojenog segmenta $x_n(m)$ kontinuirana funkcija varijable frekvencije ω , koja je zbog vremenske diskretizacije signala periodična s periodom od 2π . Međutim iz praktičnih razloga ova se funkcija izračunava samo za konačan skup diskretnih frekvencija ω_k koje su jednoliko raspoređene između 0 i π , tj. za ω_k definiran kao:

$$\omega_k = \frac{2\pi}{M}k, \quad k = 0, 1, 2, \dots, M/2 - 1 \quad (4.1-3)$$

Fourierovu transformaciju za preostale indekse k nije potrebno računati, jer ako je ulazni signal realan, tada je spektar između π i 2π jednoznačno određen sa spektrom između 0 i π kao njegova kompleksno konjugirana vrijednost. U slučaju kada je varijabla ω diskretizirana kao što je to opisano gornjim izrazom, tada se u stvari dobiva Diskretna Fourierova Transformacija vremenski diskretnog signala (kratica je DFT), koja postaje funkcija cjelobrojnog indeksa k , $X_n(k)$, a ne više realne varijable ω . Prema tome kombiniranjem sva tri gore navedena izraza slijedi da se DFT segmenta signala na indeksu n nalazi kao:

$$X_n(k) = \sum_{m=0}^{N-1} x(n+m) \cdot w(m) \cdot e^{-j\frac{2\pi km}{M}} \quad (4.1-4)$$

Izračunavanje DFT-a $X_n(k)$ u M točaka moguće je provesti primjenom tzv. brze Fourier-ove transformacije u M točaka (kratica je FFT), koja je u stvari samo numerički učinkovit postupak izvedbe DFT-a. Širina vremenskog otvora N ne mora nužno biti jednaka broju uzoraka u frekvencijskoj domeni M , tj. može biti manja, ali ne i veća. U slučajevima kada je N manji od M , tada se izdvojeni segment signala $x_n(m)$ širine N uzoraka nadopunjuje nulama do širine od M uzoraka, te se na takvom bloku od M uzoraka primjenjuje FFT. Obzirom da se izdvojeni segment dobiva množenjem ulaznog signala i vremenskog otvora, u spektru će se to množenje manifestirati kao konvolucija između spektra signala beskonačnog trajanja i spektra vremenskog otvora. U slučaju kada je ulazni signal idealni harmonijski periodički signal beskonačnog trajanja, tada će njegov spektar biti oblika para šiljaka na frekvencijama $\pm\omega_0$, kao i na svim periodičkim slikama ovog para, tj. na $\pm(\omega_0 + 2\pi l)$, gdje je l prirodan broj. Zbog toga konvolucija takvog spektra s spektrom vremenskog otvora odgovara pomaku spektra vremenskog otvora na mjesta svih navedenih šiljaka, te njihova superpozicija. Prema tome iako je ulazni signal bio idealni harmonijski signal beskonačnog trajanja, množenje s vremenskim otvorom pretvara šiljak u niz spektralnih latica. Širina glavne laticе je određena širinom vremenskog otvora N i to tako da što je vremenski otvor širi, to će glavna latica spektra otvora biti uža. Kod pravokutnog vremenskog otvora ta širina iznosi $\pm(2\pi/N)$ dok kod npr. Hamming-ovog ili Hann-ovog otvora širina iznosi $\pm(4\pi/N)$. U slučaju da su harmonijske komponente signala relativno bliske po frekvenciji, tada vremenski otvori s širokom glavnom laticom nisu pogodni, jer može doći do preklapanja spektara takvih bliskih komponenata. Dakle može se zaključiti da je na spektralnu razlučljivost je moguće utjecati odabirom tipa vremenskog otvora i variranjem njegove širine N .

S druge strane, broj uzoraka diskretne Fourier-ove transformacije M isključivo određuje koliko će "gusto" Fourierova transformacija biti izračunata (otipkana). Što je broj uzoraka DFT-a veći, to će niz $X_n(k)$ biti bolja reprezentacija kontinuirane funkcije $X_n(e^{j\omega})$.

Vrlo često nas prilikom analize govora interesira samo spektar snage signala, tj. njegova faza na početku okvira analize u većini primjena nije toliko bitna. Spektar snage se uobičajeno prikazuje u logaritamskom mjerilu, tj. u dB, a izračunava se na slijedeći način:

$$\begin{aligned} S_n(k) &= 10 \cdot \log_{10}(|X_n(k)|^2) \\ &= 20 \cdot \log_{10}(|X_n(k)|), \text{ [dB]} \end{aligned} \quad (4.1-5)$$

Zadnji parametar o kojem do sada nije bilo riječi jest korak analize. Iz dosadašnjih izraza je vidljivo da je Fourierovu transformaciju moguće izračunati za bilo koji početni indeks segmenta signala n . To znači da je na osnovu uzoraka $x(n)$ do $x(n+N-1)$ moguće odrediti $X_n(k)$. Prikupljanjem jednog novog uzorka na indeksu $x(n+N)$, cijeli postupak je moguće ponoviti za novi segment koji počinje na indeksu $n+1$, tj. na osnovu uzoraka $x(n+1)$ do $x(n+N)$ čime se dobiva $X_{n+1}(k)$. Najčešće ubacivanje samo jednog novog uzorka neće značajno promijeniti spektar snage signala koji se promatra, pa stoga tako "fini" pomak nema praktičnog smisla. S druge strane najveći smisleni pomak jest upravo N uzoraka koliko je i širina vremenskog otvora, jer bi inače uz veće korake analize pojedini dijelovi signala biti potpuno preskočeni. U slučaju da se prilikom analize koriste vremenski otvori koji prigušuju signal prema rubovima otvora (kao npr. Hamming-ov, Hann-ov ili Blackman-ov otvor), tada se za korak tipično odabire $N/2$. Uz takav odabir će uzorci koji su se u trenutnom okviru analize nalazili točno na desnom rubu otvora, u slijedećem okviru analize biti točno u njegovoj sredini. Korak analize se nekad izražava preko recipročne vrijednosti, tj. preko učestalosti analize. Dakle ako se DFT izračunava svakih K koraka ulaznog signala otipkanog frekvencijom otipkavanja f_s , tada je učestalost analize f_s/K okvira u sekundi

Obzirom da je spektrogram $S_n(k)$ funkcija dvije varijable: početnog indeksa okvira analize n i frekvencijskog indeksa k , moguće ga je prikazati isključivo u trodimenzionalnom prikazu. Radi jednostavnosti se obično treća dimenzija (iznos spektra snage) prikazuje bojom ili nijansama sive boje, pa je tako spektrogram moguće prikazati u obliku dvo-dimenzionalnog prikaza, gdje je na x osi pomak analize, a na y osi frekvencija. Dakle spektrogram prikazuje raspodjelu energije signala istovremeno duž vremenske osi i duž frekvencijske osi.

4.2 Specifičnosti spektralne analize govora

U svrhu spektralne analize govornog signala potrebno je podesiti glavne parametre analize, a to su:

- tip i širina vremenskog otvora, N
- broj uzoraka u spektralnoj domeni, M
- korak (učestalost) analize, K

Zavisno o tipu primjene, tj. koja svojstva govornog signala se želi analizirati, razlikuju se dva tipa spektrograma: uskopojasni i širokopojasni spektrogram. U povijesti, tj. u vrijeme dok nisu postojala računala, se spektralna analiza provodila nizom paralelno spojenih selektivnih filtara, od kojih je svaki propuštao jedan frekvencijski pojas govornog signala. Na izlazima tih filtara su se nalazili punovalni ispravljači i detektori vršne vrijednosti, čija je uloga bila ocjena (mjerjenje) energije signala u svakom od pojasa. U slučaju da je broj takvih filtara velik, tada je širina pojasa propuštanja svakog pojedinog filtra relativno mala, čime se postiže velika frekvencijska razlučljivost (selektivnost), tj. moguće je detektirati harmonijske komponente čije se frekvencije vrlo malo razlikuju. Na žalost uski frekvencijski pojas za posljedicu ima malu vremensku razlučljivost, tj. takvi "uski" filtri se sporo utitravaju i istitravaju, pa je vrlo teško detektirati brze promjene u signalu. Takav tip analize s velikim brojem uskih filtara se naziva uskopojasni spektrogram. Druga krajnost je analiza primjenom manjeg broja filtara sa relativno širokim pojaskima, ali dobrim vremenskim svojstvima, tj. brzim utitravanjem i istitravanjem. Takav tip analize se naziva širokopojasni spektrogram.

Uskopojasni spektrogram se najčešće koristi kada je potrebno odrediti parametre pobudnog signala vokalnog trakta i to posebno kada se radi o zvučnim glasovima. Osnovna frekvencija titranja glasnica se kreće u rasponu od 60 do 300 Hz, a obzirom na harmonijsku strukturu tog signala, susjedne spektralne komponente mogu biti vrlo bliske. Radi toga je za točnu analizu potreba visoka frekvencijska razlučljivost koja se postiže uskopojasnim spektrogramom. S druge strane prilikom analize frekvencijske karakteristike vokalnog trakta, tj. prilikom analize formantne strukture govora, poželjno je koristiti širokopojasni spektrogram, jer se time automatski uklanjaju varijacije uzrokovane periodičnosti pobudnog signala i dobiva se spektralna ovojnica.

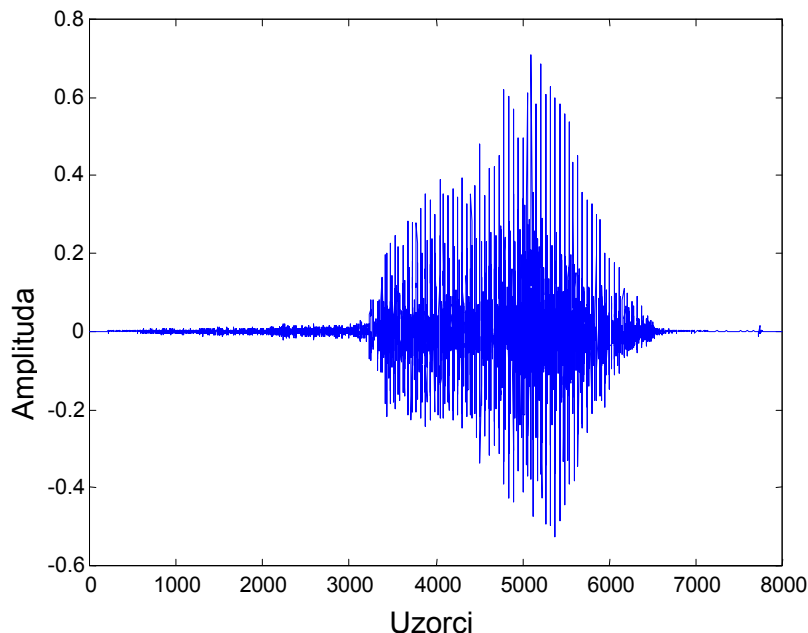
U slučaju spektralne analize govornog signala primjenom brze Fourier-ove transformacije (umjesto niza selektivnih filtara), na tip spektrograma se utječe odabirom parametara analize N , M i K . Uz frekvenciju otipkavanja od $f_s=8\text{kHz}$, tipično se za uskopojasni spektrogram odabiru otvori širine $T_a=25\text{--}30\text{ ms}$, tj. broj uzoraka otvora od $N=200$ do $N=240$ uzoraka. Vrlo često se koristi Hammingov vremenski otvor kod kojeg širina glave laticе iznosi $\pm(4\pi/N)$. Uz takav odabir otvora nije teško izračunati da minimalni razmak spektralnih komponenata signala uz koji se glavne laticе spektra otvora te dvije komponente tek dodiruju (još ne preklapaju), iznosi $4/T_a$. Uz gore odabrane širine otvora, taj razmak iznosi od 160 do 133 Hz. Taj uvjet je zadovoljen za većinu ženskih glasova, dok će kod muških ipak doći do preklapanja glavnih laticа, tj. bilo bi poželjno koristiti još šire

otvore. Na žalost širina otvora analize se rijetko povećava u odnosu na gore dane vrijednosti, jer stacionarnost signala uobičajeno ne prelazi navedene periode, pa bi time bila narušena točnost spektralne estimacije. Uz tako odabrane širine otvora, tipično se odabire spektralna rezolucija DFT-a od $M=512$ uzoraka, čime se kontinuirana funkcija $X_n(e^{j\omega})$ otipkava svakih $f_s/M = 8000/512=15.6$ Hz. Uz takav odabir će glavna latica spektra otvora biti otipkana sa $4M/N$ uzoraka, tj. sa 8 do 10 uzoraka zavisno o N . Kao što je već prije diskutirano, za korak analize dovoljno je odabrati $K=N/2$.

Kod širokopojasnog spektrograma, širina otvora analize je često uža od jedne periode glotalnog pulsa, čime se totalno gubi periodičnost u spektru. S druge strane, pošto vremenski otvor čas zahvaća segmente signala na početku glotalnog pulsa koji su velike energije, a u drugom okviru analize segmente na kraju, male energije, u širokopojasnom spektrogramu će se pojaviti vertikalne pruge.

4.3 Praktična vježba

U okviru početnog dijela vježbe, spektralna analiza će se provoditi na "Wav" datoteci "sa.wav" u kojoj se nalazi snimka izgovora riječi "sa", kao karakteristični primjer bezvučnog i zvučnog glasa. Kompletan analiza se provodi Matlab programom "specgrm.m", a radi razumijevanja primijenjenih postupaka u ovom dijelu vježbe će svi koraci analize biti detaljnije objašnjeni. Navedena datoteka se učitava u Matlab te se filtrira pojasno propusnim filtrom kao što je to objašnjeno u prijašnjim vježbama, kako bi se izbjegli utjecaji komponenti spektra koji nisu interesantni za analizu. Ova početna predobrada signala se obavlja u prvoj polovici programa, koju pokrenite i izvršite tako dugo dok program ne prikaze sliku filtriranog signala. Promatrani signal "sa.wav" (izgovoreno "sa") prikazan je slikom 4.3-1, vidi se da je amplituda glasa "a" puno veća od amplitude bezvučnog glasa "s". Ova slika će poslužiti kao osnova za promatranje spektrograma, tj. kao orijentacija za vremensku os spektrograma.



Slika 4.3-1 Govorni signal u trajanju 1s, 8000 uzoraka

Nakon što program prikaže ovu sliku prekinite izvršavanje programa sa Ctrl-C i nastavite drugu polovicu programa liniju po liniju. Glavni parametri analize se definiraju u zaglavlju

drugog dijela programa "specgrm.m" i to pomoću varijabli N, kor i otv. kao što je prikazano u slijedećem primjeru za uskopojasni spektrogram :

```
%=====
% Glavni dio programa sa spektralnom analizom signala
%=====

% Glavni parametri spektralne analize
N=512;           % broj uzoraka u kojem se radi FFT
                % (spektralna rezolucija M)

% Primjer parametara za uskopojasni spektrogram
kor=80;          % korak izmedju dva susjedna bloka analize
                % (korak analize K)
otv=200;         % sirina vremenskog otvora N
```

Obzirom da se energija govornog signala mijenja kroz vrijeme neki okviri će imati veću energiju, dok će drugi imati manju. Kod nekih prikaza takve varijacije u nivou signala nisu poželjne, jer se tada dijelovi signala male energije slabo vide u spektrogramu. Radi toga je u takvim situacijama potrebno svaki okvir analize normalizirati, tako da ima približno istu energiju. Odabir da li se provodi ta normalizacija ili ne ostvaruje se binarnom varijablom norm_amp, kao što je prikazano:

```
norm_amp=1;      % binarna varijabla koja odredjuje da li se radi
                % normalizacija amplitude svakog pojedinog okvira
```

Slijedeći korak je definiranje tipa vremenskog otvora w kojim se izdvaja dio signala. Odabir otvora se provodi "komentiranjem" onih linija koje nisu potrebne (samo jedna mora biti aktivna ... bez znaka % na početku reda). Primjeri definiranja otvora su dani u nastavku:

```
w=hamming(otv);    Hammingov otvor
w=blackman(otv);   Blackmanov otvor
w=ones(otv,1);     pravokutni otvor
```

Na osnovu broja uzoraka ulaznog govornog signala i na osnovu odabranih parametara analize moguće je odrediti ukupni broj okvira analize prema izrazu:

```
br_kor=floor((length(y)-otv)/kor); % max broj blokova u snimljenom signalu
```

tj. ovisi o broju uzoraka promatranog signala y, veličini otvora otv i zadanom koraku analize kor, te uz prije odabrane parametre analize slijedi da je br_kor=97 tj. da se za dobivanje spektrograma treba izračunati 97 Fourier-ovih transformacija. Radi jednostavnosti će analiza prvo biti provedena na samo jednom okviru analize. Stoga, umjesto izvršavanja cijele petlje za izračunavanje spektrograma u nastavku programa, indeks okvira analize i će biti fiksiran na i=60, što približno odgovara sredini izgovora glasa "a".

Zadatak 1: Uz tako fiksiran indeks okvira analize, izvoditi naredbe unutar petlje jednu po jednu i prikazivati signal (odnosno spektar) nakon svake pojedine linije, te tako dobivene slike pohranjivati u vašu .Doc datoteku. Naredbe kojima se provodi obrada jednog okvira analize su:

```
xnm=y((i-1)*kor+[1:otv]); % izvoji segment signala na pocetnom
                          % indeksu (i-1)*kor+1
```

```

if (norm_amp), % ako se provodi normalizacija
    xnm=xnm/sqrt(var(xnm)); % energije svakog okvira analize
end; % tada svaki okvir podijeli s drugim
% korijenom iz varijance,
% cime se dobiva signal s jedinicom
% varijancom
xnm=xnm.*w; % pomnozi izdvojeni signal s uzorcima
% vremenskog otvora
sp=fft(xnm, N)/(otv/2); % izracunaj DFT u N tocaka
sp=20*log10(abs(sp(1:N/2))); % spektar snage u decibelima
% (samo prva polovica)
figure(1); plot(sp); axis([1 N/2 min_am max_am]); pause(0.1)

```

Variraj tip i širinu vremenskog otvora analize i diskutiraj kako to utječe na dobiveni spektar `sp`. Pohrani spektre za slijedeće kombinacije parametara:

pravokutni i Hamming-ov otvor sa `otv=200`

Hamming-ov i Blackman-ov otvor sa `otv=80`

4.4 Izračunavanje spektrograma

Nakon što je proučen sam program i značenje svake pojedine linije koda, moguće je uistinu izračunati spektrogram, tj. provesti analizu za sve okvire od 1 do `br_kor`. Spektrogram će u Matlab-u biti pohranjen u matrici, koju je prije početka petlje potrebno alocirati, što se postiže slijedećom naredbom:

```
spgr=zeros(N/2,br_kor);
```

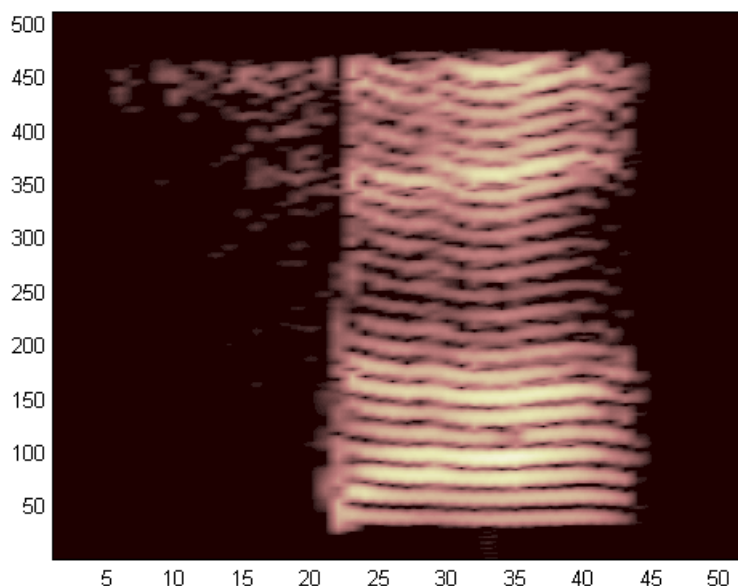
Slijedi analiza svih okvira u signalu u petlji:

```

for i=1:br_kor
    % OVDJE SU NAREDBE UNUTAR PETLJE (VIDI ZADATAK 1)
    % Naredba za pohranu spektra trenutnog okvira u matricu
    spgr(:,i)=sp;
end;

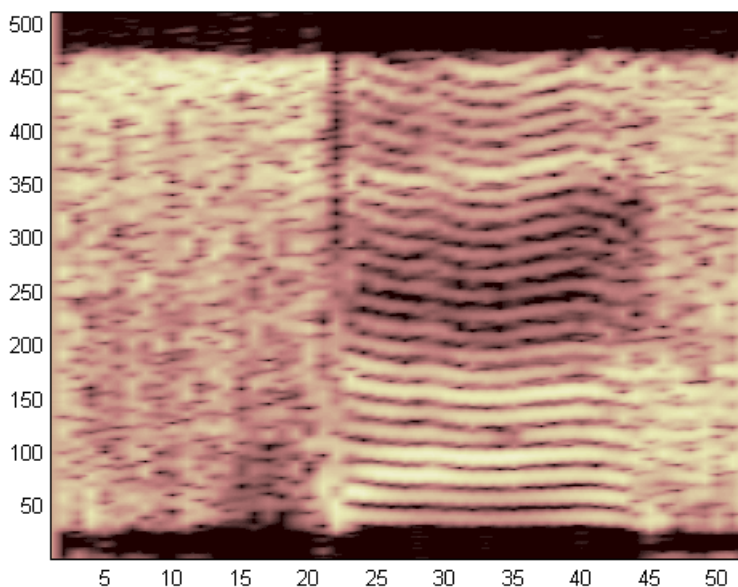
```

Primjer uskopojasnog spektrograma bez normalizacije amplitude (tj. uz `otv=200`, `kor=160`, `N=1024`, te `norm_amp=0`) prikazan je slikom 4.4-1. Vidljive su horizontalne linije koje jasno iscrtavaju pojedine harmonike za glas “a” tj. prikazuju sinusne komponente ovog periodičnog signala. Obzirom da su te komponente u harmoničkom odnosu, varijacije osnovne frekvencije titranja glasnica su očitije u višim harmonicima, nego u nižim. U prvom dijelu spektrograma se vidi da je energija glasa “s” jako mala u usporedbi sa vokalom “a” i da je spektar oblika šuma (dotična pojava je bolje vidljiva na slici 4.4-2). Na vremenskoj osi je jasno vidljiv trenutak promjene glasa “s” u glas “a”.



Slika 4.4-1 Uskopojasni spektrogram "sa"

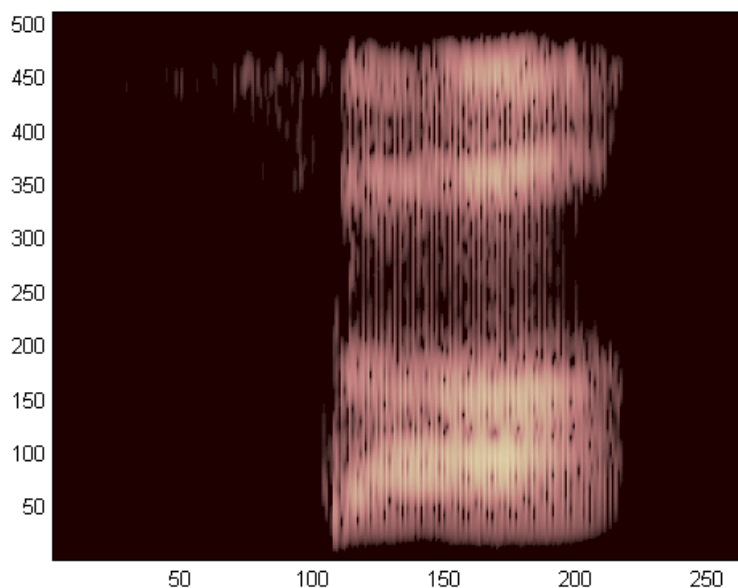
Ovaj tip prikaza je dobar za promatranje promjene apsolutnog iznosa energije signala u vremenu i frekvenciji. Nedostatak je, kao što je vidljivo na slici 4.4-1, loš prikaz spektra u vremenu trajanja glasa "s" u odnosu na glas "a". Da bi se vidio oblik spektra za sva područja primjenjuje se normalizacija za svaki izračunati okvir analize spektrograma. Rezultat je prikazan slikom 4.4-2. Sada jasno vidimo karakteristike spektra pojedinih područja, te kako se energija mijenjala u svakom od blokova (jednaka boja ne označava jednaku energiju, već jednaku normiranu energiju pojedinog bloka). Normalizacija amplitude se provodi postavljanjem varijable `norm_amp` u jedinicu.



Slika 4.4-2 Uskopojasni spektrogram "sa" sa amplitudno normiranim okvirima

Dobiveni širokopojasni spektrogram prikazan je slikom 4.4-3. Vidljive su vertikalne pruge, tj. sada se povećala razlučljivost na vremenskoj osi. Spektrogram uz vertikalne linije,

koje prikazuju kako se mijenja energija glasa u vremenu za svaki pojedini glotalni puls, jasno prikazuje formantnu strukturu govornog signala, tj. vidljive su široke horizontalne pruge koje pokazuju pozicije rezonantnih karakteristika vokalnog trakta za glas "a". U ovom prikazu amplituda svakog pojedinog okvira nije bila normalizirana.



Slika 4.4-3 Širokopojasni spektrogram "sa"

Nameće se pitanje koji je spektrogram bolji za analizu i do koje granice sa veličinom otvora i koraka ići. Na to pitanje nema jasnog odgovora, tj. jedino se može reći da svaki daje određenu korisnu informaciju, te se primjenjuje jedan ili drugi zavisno o tome koji se parametar želi analizirati.

Zadatak 2

Prije pokretanja programa `specgrm.m` potrebno je izmijeniti vrijednosti `N`, `kor` i `otv`, kao što je navedeno u tablici 4.4-1, da bi se dobio određeni spektrogram.

Potrebno je pokušati odrediti, na uskopojasnom spektrogramu, razmak među harmonicima (uzimajući u obzir da je frekvencija u rasponu 0-4kHz, uz pomoć prebrojavanja pruga i širine područja u kojem su prebrojene).

Tablica 4.4-1 Preporučene vrijednosti za `specgrm.m`

	N	kor	otv
Širokopojasni	1024	30	50
Uskopojasni	1024	150	200

Što se događa kada variramo korak između dva susjedna bloka analize `kor`? Kada je prikaz bolji?

Prikažite spektrograme za različite tipove vremenskih otvora `w` (Hamming, Blackman, pravokutni ...), ... koji je najbolji?

Pogledajte što se događa kada uzmemo mali `otv` i Blackman-ov `w`, da li je rezultat spajanja pruga dobra stvar?

Pogledajte spektrograme za svoje glasove snimljene na prethodnim vježbama.

5. ANALIZA FORMANTNE STRUKTURE GOVORA

5.1 Uvod

Formantima se nazivaju rezonantna nadvišenja u amplitudnom spektru govornog signala, a opisuju se formantnim frekvencijama na kojima dolazi do nadvišenja (pri tome se misli na centralne frekvencije) kao i širinom formanta. Formantna struktura jednoznačno karakterizira glas koji se izgovara, a najlakše ju je ilustrirati na slučaju samoglasnika zbog njihove duže stacionarnosti, što će biti ujedno i glavna tema ove vježbe. Analiza formantne strukture govornog signala provest će se pomoću LPC analize (*engl. Linear Prediction Coding*), no sama LPC tehnika bit će kasnije detaljno obrađena na predavanjima i sljedećim vježbama. Osnovna ideja LPC analize je pretpostavka da se trenutni uzorak govora može predvidjeti pomoću linearne kombinacije određenog broja prošlih uzoraka. Svi iznosi prošlih uzoraka ne doprinose jednako iznosu trenutnog uzorka, pa se moraju prije sumacije pomnožiti odgovarajućim težinskim koeficijentima. Ti težinski koeficijenti su parametri linearnog prediktora. Cilj LPC analize je odrediti parametre prediktora. Očito je da parametri prediktora ne mogu biti konstantni u duljem vremenskom intervalu, pa se oni zato računaju nad okvirima analize signala konačnog trajanja (*engl. frame*) na kojima se mogu smatrati stalnima (tipično oko 20-30 ms). Kao što će biti kasnije detaljnije objašnjeno, koeficijenti prediktora ujedno su i koeficijenti IIR filtra koji modelira spektralnu ovojniciu govornog signala, odnosno frekvencijsku karakteristiku vokalnog trakta. Taj filter ima samo polove (*engl. all-pole IIR filter*). Amplitudno-frekvencijska karakteristika takvog filtra predstavlja ovojniciu (*engl. envelope*) vremenski kratkotrajnog spektra signala za određeni okvir analize. Formantne frekvencije određene su pozicijama konjugirano kompleksnih parova polova tog filtra. U okviru ove vježbe nećemo ulaziti u sam postupak određivanja koeficijenata prediktora, nego ćemo isključivo promatrati kako izgledaju spektralne ovojnice za različite ulazne glasove.

Za potrebe ove i sljedeće vježbe potrebno je snimiti vlastitu zvučnu datoteku. Za snimanje koristite *Sound Recorder* (Start gumb, Run, u polje Open upišite `sndrec32`, OK), sa sljedećim parametrima:

- sampling rate: 8000 Hz
- broj kanala: mono
- kvantizacija: 16 bita

Za detaljnija objašnjenja pogledajte upute za prvu vježbu. Potrebno je snimiti samoglasnike točno sljedećim redom: a, e, i, o, u, 3. Pri čemu je '3' neutralni samoglasnik šva (schva), a fonetičari ga u kraćoj varijanti označavaju oznakom koja izgleda kao zrcaljeno malo slovo *e* (kao u britanskom engleskom u riječi *ago*), a u dužoj varijanti kao '3:' (kao u britanskom engleskom u riječi *her*). Prilikom snimanja, između glasova je potrebno ostaviti pauzu u trajanju oko 0.5 s. Prilikom izgovora glas treba biti što ravnomjerniji, te ne smije doći do prekoračenja dinamike (*engl. clipping*). Snimite dvije zvučne sekvence, a datoteke nazovite *prezime_broj.wav* i spremite ih u direktorij 'c:\Users\Dog\Vj5\wav\'.
'

U slučaju da su 'wav' datoteke već snimljene u okviru prve vježbe, prethodna točka se može preskočiti.

5.2 Primjer vježbe

Pokrenite *Matlab* i pozicionirajte se u radni direktorij za ovu vježbu (npr. »cd c:\Users\Dog\Vj5). Naredbom *what* provjerite da li u radnom direktoriju postoje datoteke potrebne za ovu vježbu.

```
» what
```

```
M-files in the current directory c:\Users\Dog\Vj5
```

```
dog5          lpc_robust poziv_cov2 rijec  rijec2  wavread2  wavwrit2
```

U ovoj vježbi pokrenut će te direktno programe *poziv_cov2*, *dog5*, *rijec* i *rijec2*.

5.2.1 Datoteka *poziv_cov2.m*

Ova datoteka se brine za pokretanje LPC analize na uzorcima zadane 'wav' datoteke s odgovarajućim parametrima. Rezultate nastale izvršavanjem *poziv_cov2* koristit će glavna datoteka *dog5*. Otvorite datoteku *poziv_cov2* pomoću *Matlabovog* editora:

```
»edit poziv_cov2
```

ukoliko nešto nije uredu s instalacijom *Matlaba*, pa *edit* ne radi, datoteku otvorite *Notepadom*. Program *poziv_cov2* postavlja određene parametre LPC analize u potrebne iznose i zatim poziva glavnu funkciju *lpc_robust* koja provodi LPC analizu. U ovom slučaju zanimljiva su dva parametra:

- param(4) – faktor *spektralnog gladenja* (engl. preemphasis)
- param(5) – red prediktora

Faktor spektralnog gladenja – preemphasis

Fizikalno je svojstvo spektra amplitude govornog signala da komponente nižih frekvencija imaju primjetno više amplitude od komponenti viših frekvencija. Poželjno je izglatiti spektar signala koji se obrađuje kako bi viši formanti bili uočljiviji, a diskretan sustav što manje osjetljiv na smetnje konačne preciznosti. Zato se prije LPC analize signal propušta kroz *preemphasis* filtar čiji je frekvencijska karakteristika stalna ili sporo vremenski promjenljiva. Najčešće se koristi vremenski stalan diskretan sustav prvog reda oblika kao u izrazu (5.2-1):

$$H(z) = 1 - a \cdot z^{-1}, \quad \text{gdje je } 0.9 \leq a \leq 1.0 \quad (5.2-1)$$

Uobičajen iznos koeficijenta *a* je 0.9375 ili (15/16), jer se množenje s takvom 4-bitnom frakcijom lako provodi na procesorima s cjelobrojnom aritmetikom. Ukoliko se želi izbjeći *preemphasis* dovoljno je za *param(4)* staviti vrijednost 0, čime će filtrirani signal biti jednak ulaznom

Red prediktora

Pogreška predikcije linearnog prediktora $E(z)$ reda p je u Z domeni definirana sljedećim izrazom:

$$E(z) = A(z)S(z), \quad (5.2-2)$$

gdje je $S(z)$ Z transformacija govornog signala, a $A(z)$ definiran kao što slijedi:

$$A(z) = 1 - \sum_{i=1}^p a_i \cdot z^{-i} \quad (5.2-3)$$

U slučaju otipkavanja govornog signala s frekvencijom otipkavanja $fs=8000$ Hz, red prediktora tipično iznosi oko $p=10$. Ukoliko red prediktora nije dobro odabran može doći do gubljenja (ili pojavljivanja lažnih) formanata u LPC modelu.

U vježbi je red prediktora jednak za cjelokupnu zvučnu sekvencu. Sustav $A(z)$ se ujedno naziva i inverzni filter, a njemu pripadni *all-pole* filter čija amplitudno frekvencijska karakteristika opisuje ovojnicu vremenski kratkotrajnog spektra signala dobiva se kao: $H(z) = 1/A(z)$.

Kao primjer u ovom dijelu vježbe obradit će se '.wav' datoteka '*samo_dado.wav*'. Odabran je preemphasis faktor 0.9375 i deveti red prediktora. U datoteci *poziv_cov2* podesite put do glavnog foldera i pripadnih pod-foldera gdje se nalaze ulazne '.wav' datoteke, odnosno put kamo će biti pohranjeni 'mat' file s parametrima LPC analize, npr.:

```
dire='C:\Users\Dog\Vj1\';
mat_pref='mat\LPC_params_';

wav_file=[ dire 'dado\samo_dado'];      mat_file=[mat_pref 'dado'];

.....
param(4) = 0.9375
param(5) = 9
.....
```

Snimite i pokrenite *poziv_cov2*, a nakon što je obrada završila u direktoriju *mat* nalazi se datoteka *LPC_params_dado.mat* u kojoj su pohranjene sljedeće varijable:

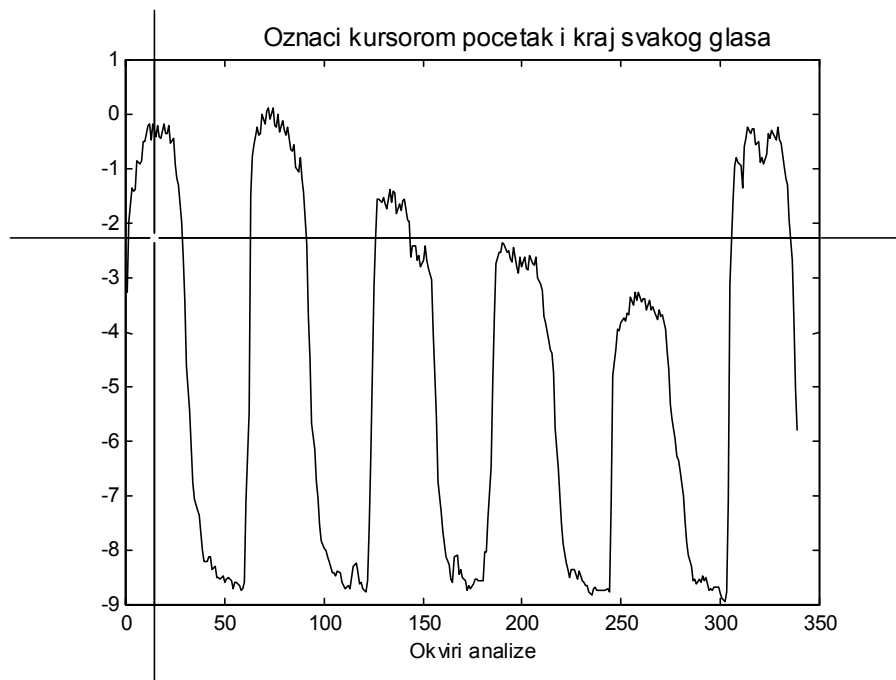
- *wavefile* string s nazivom ulazne 'wav' datoteke
- *param* vektor parametara korištenih za analizu
- *fs* frekvencija otipkavanja
- *a_m* matrica s a koeficijentima prediktora za svaki okvir analize
- *En_ma* vektor rezidualne energije nakon LPC-a

5.2.2 Datoteka *dog5.m* – početni dio

Otvorite datoteku *dog5.m* u *Matlabovom* editoru ili u *Notepadu* te pronađite liniju koja počinje s *load* (na početku programa). Promijenite tu liniju tako da učitava upravo dobivenu datoteku *lpc_params_dado.mat*.

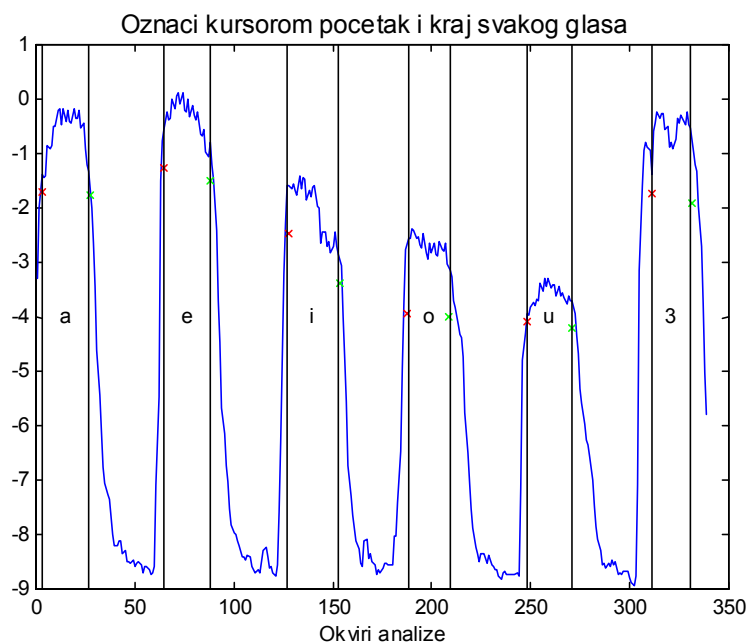
```
load mat\lpc_params_dado
```

Snimite promjenu i pokrenite *dog5*. Nakon pokretanja čut će te snimljenu sekvencu i pojaviti će se sljedeća slika:



Slika 5.2-1 Rezidualna energija snimljene sekvence *En_ma*

Budući da samoglasnici imaju veliku energiju, svaki „paket” predstavlja samoglasnik. Sada treba mišem kliknuti na početak i na kraj svakog samoglasnika, da bi program u nastavku znao točno gdje je početak/kraj svakog pojedinog glasa. Ne treba voditi računa o 'visini' gdje se klikne, već isključivo o x-poziciji. Pri tome program podrazumijeva da je redosljed slijedeći: *a, e, i, o, u, 3*, pa o tome treba voditi računa prilikom snimanja datoteke. Nakon što se označi i zadnji glas, slika izgleda kao što je prikazano na 5.2-2:



Slika 5.2-2 Rezidualna energija snimljene sekvence *En_ma* nakon označavanja svih glasova

Kao što je vidljivo na slici, početke i krajeve svakog glasa treba označavati tako da se zahvati samo dio signala velike energije (nemojte uključivati i rubove, jer se na rubovima frekvencijska karakteristika LPC modela mijenja). Sada je moguće pritiskom na kombinaciju CTRL+C prekinuti daljnje izvođenje *dog5.m* datoteke. Slijedi detaljnije objašnjenje do sad obavljenog posla. Početni i završni indeksi okvira analize svakog glasa pohranjeni su u vektorima *poo* i *zaa* :

```
» [poo'; zaa']
```

```
ans =
     3     65    127    188    248    311
    27     88    153    209    271    331
```

Ukoliko se u *Matlabovom* komandnom prozoru zada naredba:

```
»whos En_ma a_m
```

rezultat je:

```
» whos En_ma a_m
Name          Size          Bytes  Class

En_ma         339x1           2712  double array
a_m           339x10          27120  double array
```

Budući da se rezidualna energija predikcije *En_ma* (prikazana na slici 5.2-1) kao i *a_m* (matrica koeficijenata prediktora) računaju **za svaki** okvir analize, očito je da postoji 339 okvira analize. Broj okvira analize i red prediktora određuju se iz dimenzija matrice *a_m*:

```
[bf,p]=size(a_m);
p=p-1;
```

Budući da su u retku matrice *a_m* pohranjeni koeficijenti prediktora u obliku: $[1 \ a_1 \ a_2 \ \dots \ a_{p-1} \ a_p]$ potrebno je od broja stupaca oduzeti jedinicu kako bi se dobio red prediktora. Sada slijedi određivanje spektra LPC filtra u *BT* točaka za sve glasove i to za sve okvire analize koje pripadaju svakom pojedinom glasu. Radi jednostavnosti, svi rezultati (tj. frekvencijske karakteristike) biti će pohranjene u jednu veliku matricu *hm*, gdje svaki stupac odgovara jednom okviru analize pojedinog glasa.

```
BT=256; % Broj tocaka za izracunavanje
        % frekvencijske karakteristike
bfsam=zaa-poo+1; % broj okvira analize za svaki pojedini glas
```

Broj zahvaćenih okvira analize se razlikuje za svaki glas, tj. ovisi o tome gdje ste točno kliknuli prilikom označavanja glasova, a pohranjen je u vektoru *bfsam*. Za ovaj konkretan primjer on iznosi:

```
» bfsam'
```

```
ans =
    25    24    27    22    24    21
```

za glasove, a,e,i,o,u 3. U nastavku programa, izračunavamo matricu *hm*.

```

bf=sum(bfsam); % ukupni broj okvira analize
hm=zeros(BT,sum(bfsam)); % formiraj matricu za pohranu frekvencijske
% karakteristike
k=1; % brojac za pohranu u veliku matricu
for i=1:bg, % za sve glasove
    for j=poo(i):zaa(i), % preko svih indeksa okvira
        % analize koji pripadaju tom glasu
        [h,om]=freqz(1,a_m(j,:),BT); % izracunaj frek. karak. za
        % j-ti okvir LPC analize
        hm(:,k)=20*log10(abs(h)); % modul pretvori u [dB] i
        % spremi u matricu
        k=k+1; % povecaj brojac za pohranu
    end;
end;
end;

```

Obzirom da su sada svi glasovi pohranjeni jedan za drugim u matrici *hm* potrebno je na neki način znati gdje su rubovi u tu svrhu formiramo dva stupca dimenzije *bg* (broj glasova) od kojih prvi sadržava početni indeks stupca u matrici *hm*, a drugi završni indeks za svaki pojedini glas.

```

za=cumsum(bfsam); % Završni indeks
po=[1; za(1:bg-1)+1]; % Početni indeks

```

koji u našem konkretnom primjeru iznose:

```

» [po'; za']

ans =
    1    26    50    77    99   123
   25    49    76    98   122   143

```

To drugim riječima znači da će frekvencijske karakteristike za npr. glas *e* biti pohranjene u stupcima 26 do 49 matrice *hm*. U nastavku programa će izračunate frekvencijske karakteristike biti prikazane za svaki pojedini glas.

5.2.3 Datoteka dog5.m – drugi dio

Ponovno pokrenite datoteku *dog5.m* Nakon ponovnog označavanja svih glasova, *Matlab* će prvo prikazati izdvojene samoglasnike u vremenskoj domeni, tj. nacrtati će *Nt* uzoraka iz 'sredine' svakog označenog glasa. Pritiskom na tipku *ENTER* iscertavat će se ostali samoglasnici. Primjer valnog oblika samoglasnika *a* pokazuje slika 5.2-3.

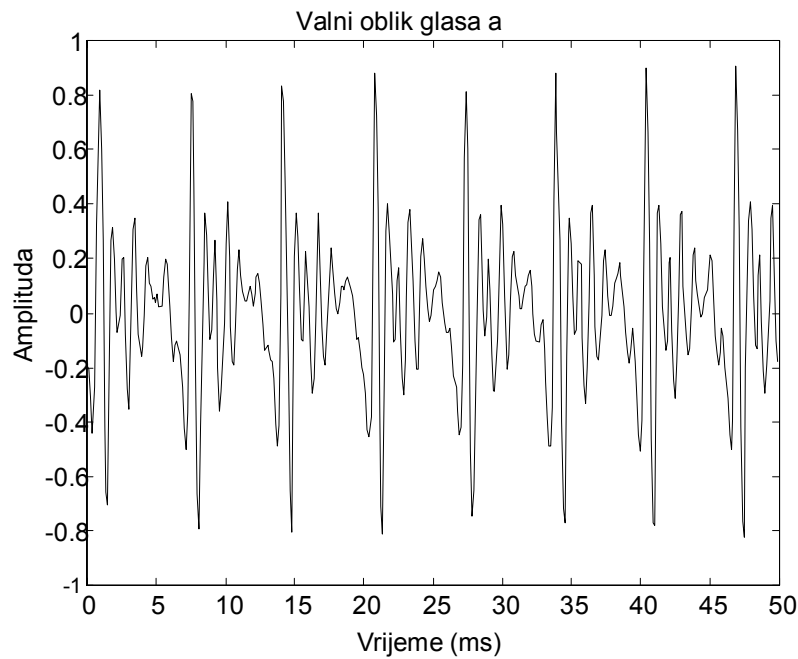
Nakon valnih oblika *Matlab* će iscrtati ovojnice kratkotrajnih spektrograma samoglasnika, odnosno module spektra *all-pole* LPC filtra i to za svaki glas nezavisno. Obzirom da svakom glasu pripada cijeli niz okvira analize (cca 20-40) frekvencijske karakteristike svih njih zajedno biti će prikazane u kumulativnom prikazu (tj. jedna preko druge) slijedećim blokom naredbi:

```

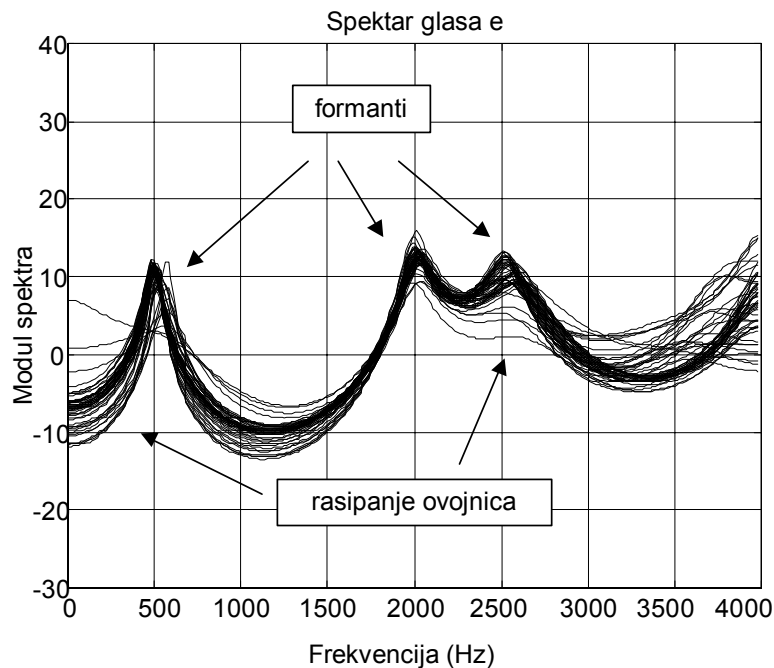
for i=1:bg,
    plot(om/pi*fs/2,hm(:,po(i):za(i)));axis([0 4000 -30 40]);grid;
    title(['Spektar glasa ' sa(i)]);
    xlabel('Frekvencija (Hz)');
    ylabel('Modul spektra (dB)');
    pause;
end;

```

Primjer takvih ovojnica za samoglasnik *e* prikazan je na slici 5.2-4.



Slika 5.2-3 Primjer valnog oblika samoglasnika *a*

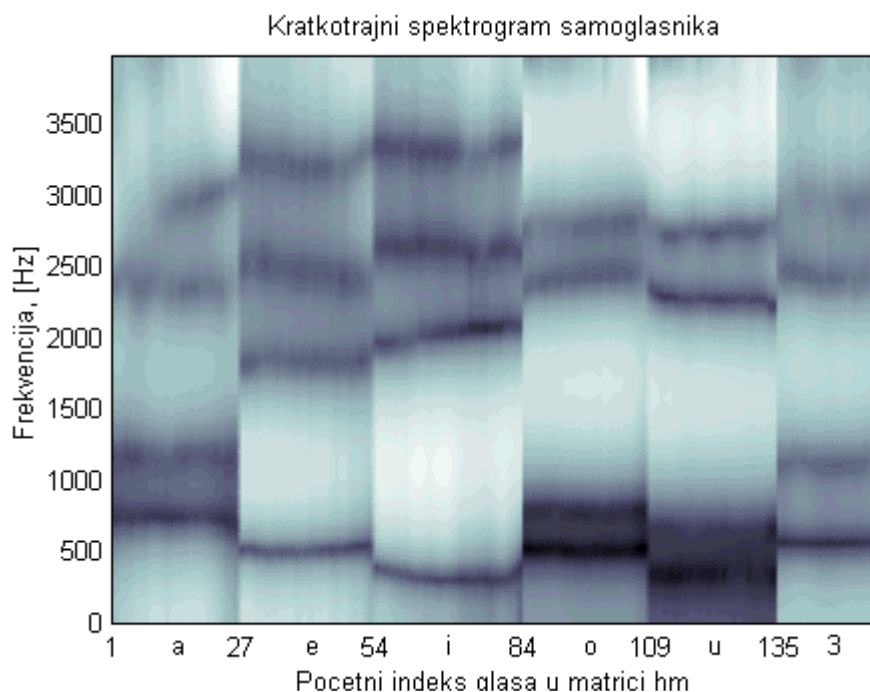


Slika 5.2-4 Primjer spektralnih ovojnica samoglasnika *e*

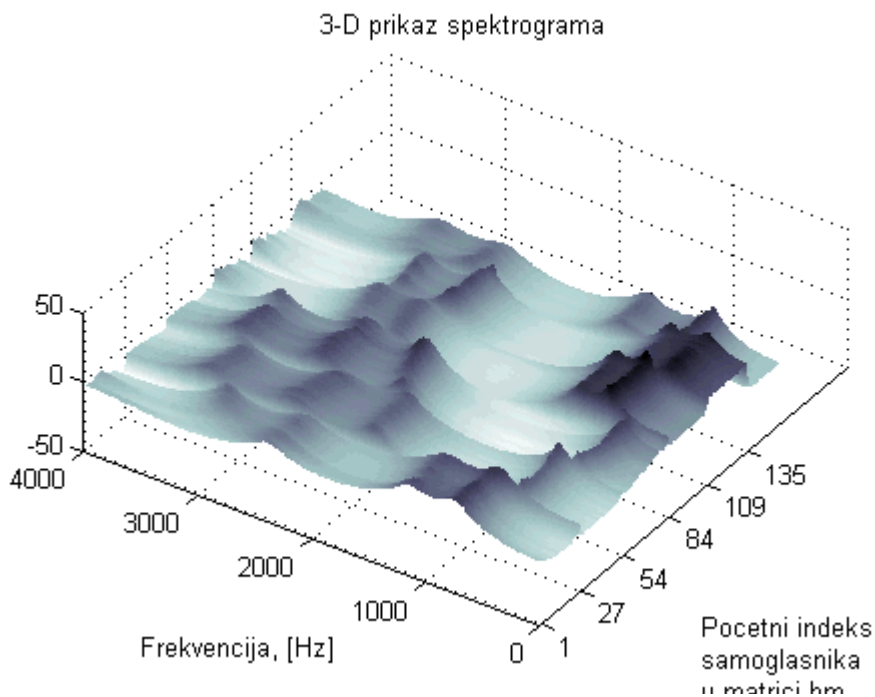
Na slici je vidljivo da može doći do „rasipanja” ovojnica, što posebno dolazi do izražaja ukoliko je samoglasnik izgovoren nejednolikom intonacijom ili ako je odabran krivi red prediktora. Nakon toga prikazat će se kratkotrajni spektrogram u 2D i 3D prikazu. Spektrogram u 3D prikazu iscrtava se naredbama:

```
figure(2);surf([1:bf],om/pi*fs/2,hm);shading interp;view(-56,66);
```

Pogled na spektrogram postavlja se naredbom `view(az,el)` gdje je *az* oznaka za azimut odnosno kut rotacije oko *z* osi, dok je *el* iznos kuta vertikalne elevacije (vertikalni nagib). Obje veličine su u stupnjevima. Pomoću naredbe `view` u komandnom prozoru *Matlaba* ugodite optimalnu kutove tako da se podjednako dobro vide i vremenska i frekvencijska promjena formantnih karakteristika. Primjeri oba spektrograma prikazani su na slikama 5.2-5 i 5.2-6.



Slika 5.2-5 Primjer LPC ovojnice kratkotrajnog spektrograma 2-D



Slika 5.2-6 Primjer LPC ovojnice kratkotrajnog spektrograma 3-D

I konačno potrebno je odrediti iznose formantnih frekvencija za svaki samoglasnik. Kao što je rečeno u uvodu, formantne se frekvencije dobivaju na osnovu pozicije konjugirano kompleksnih polova *all-pole* filtra. Određivanje formantnih frekvencija provodi se sljedećim blokom naredbi za svaki okvir analize:

```
r=roots(a_m(i,:));
r(find(abs(imag(r))<1e-12))=[];
```

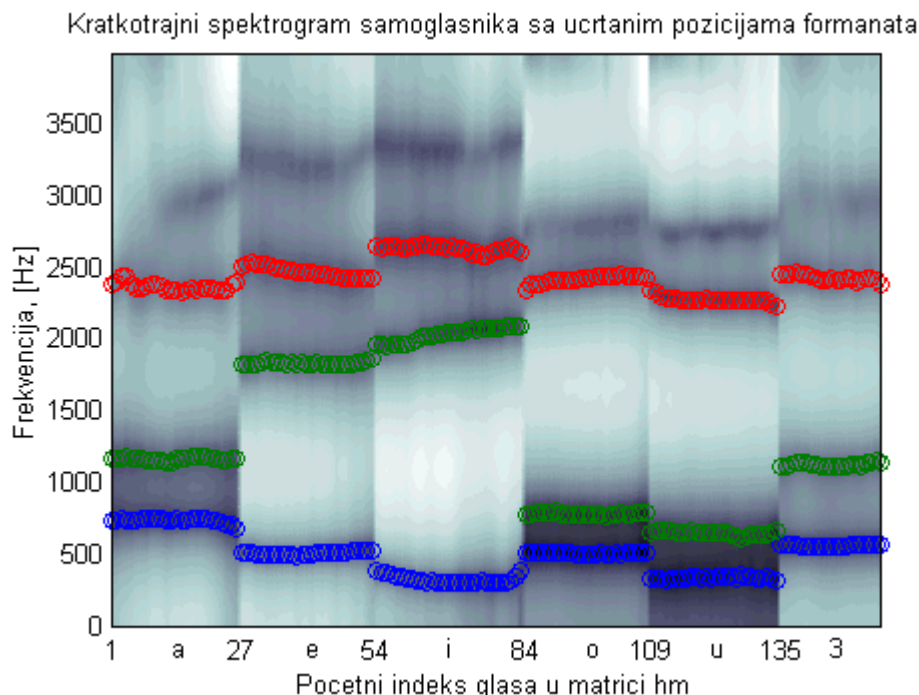
Posljednjom naredbom iz vektora *r* uklonjeni su realni korijeni (zbog pogreške u računanju i kod realnih korijena se može pojaviti određen imaginarni dio vrlo malenog iznosa). Slijedi uklanjanje konjugiranog para, kako bi u vektoru *r* ostao samo po jedan korijen za svaki par:

```
for j=1:max(size(r))/2,
    r(min(find(abs(r-r(j)')<1e-12)))=[];
end;
```

Na kraju slijedi računanje centralnih frekvencija formanta kao kuta kompleksnog pola:

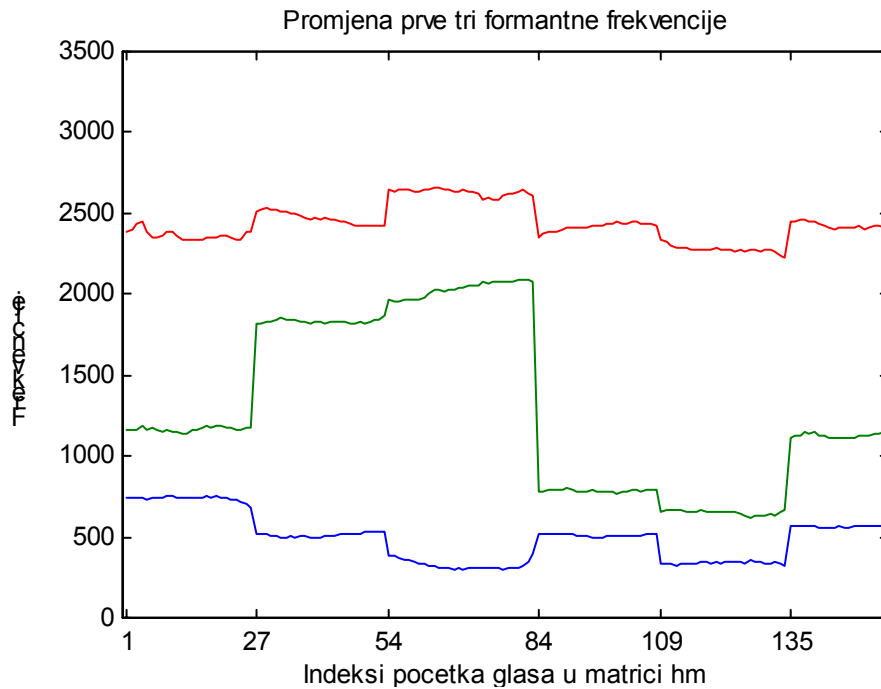
```
fr(i,1:max(size(r)))=angle(r)';
```

Nakon sortiranja po rastućim frekvencijama *Matlab* crta novi graf koji je jednak onome prikazanom na slici 5.2-5, jedino su na sliku još kružićima doctane pozicije formanta (svaki formant drugom bojom)



Slika 5.2-7 LPC ovojnica spektrograma sa ucrtanim pozicijama prva tri formanta

Na zadnjem prikazu, program prikazuje samo frekvencije formanta kao funkcije vremena (tj. indeksa okvira analize), kao što je prikazno na slici 5.2-8



Slika 5.2-8 Prve tri formantne frekvencije

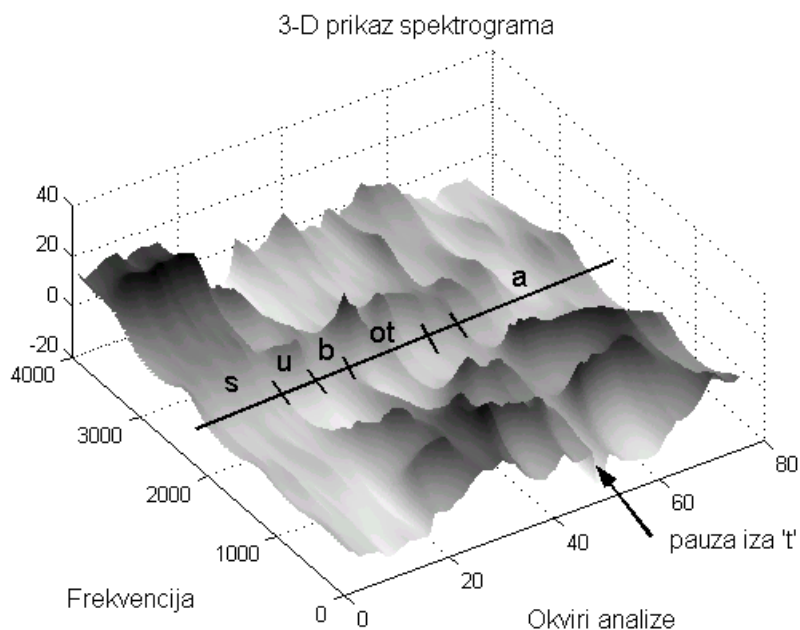
5.2.4 Program *rijec.m*

Program *rijec.m* je kombinacija programa *poziv_cov2* i *dog5*, a koristi se za 2D i 3D prikaz LPC ovojnice kratkotrajnog spektrograma jedne riječi. Ideja je vidjeti pojavu formanta u spektrogramu stvarne riječi, a ne idealno izgovorenih samoglasnika. U ovom slučaju analizirana je datoteka *subota.wav* istog govornika kao i u prošlim primjerima. Korisnik mijenja sljedeće linije, slično kao i u prošlim primjerima:

```
param(4)=0.9375; % faktor za pre-emphasis ili 0 ako je bez
param(5)=9;      % LPC predictor order

Lpc_robust('wav\\subota.wav','mat\\subota_9.mat',param,0);
load mat\\subota_9
```

Primjer 3D spektrograma je prikazan na slici 5.2-9.



Slika 5.2-9 Primjer spektrograma riječi 'subota'

5.3 Zadaci za samostalno izvođenje

Zadatak 1.

Ako još ne postoji snimiti '.wav' datoteku sa nizom samoglasnika, kao što je objašnjeno u uvodu. Obradite ih pomoću *poziv_cov2* i *dog5*. Isprobajte nekoliko redova prediktora (npr. 9., 10. i 11.) dok ne dobijete minimalno rasipanje ovojnice spektrograma i centralnih frekvencija formantata. LPC ovojnice kratkotrajnih spektara signala u 1D, 2D i 3D prikazu prenesite u *MS Word* pomoću COPY + PASTE postupka i snimite u vašu datoteku na disk. U tablicu 5.3-1 upišite početne i završne okvire analize u kojima su samoglasnici.

Tablica 5.3-1 Tablica okvira analize samoglasnika

vokal	početak	završetak
a		
e		
i		
o		
u		
3		

Zadatak 2

Provedite analizu programom *rijec2.m* koji je već prilagođen da učitava '.wav' datoteku sa snimkom vašeg imena da se odredi spektralna ovojnica za vaš glas.

Snimite kraću riječ s dovoljno samoglasnika npr. subota, utorak... (izbjegavajte riječi poput najindustrijaliziraniji, klistron ili škrob :). Obradite ju datotekom *rijec*, rezultate pohranite u *Word* datoteku. Pomoću 2D i 3D slike pokušajte pronaći samoglasnike.

6. AUTOMATSKA KLASIFIKACIJA SAMOGLASNIKA NA OSNOVU FORMANTNE STRUKTURE

Kao što smo mogli uočiti u prošloj laboratorijskoj vježbi, samoglasnici imaju vrlo karakterističnu formantnu strukturu. Radi toga, još su davno istraživači došli na ideju, da bi glasove bilo moguće automatski prepoznati, tj. klasificirati na osnovu formantne strukture. Pri tome je dovoljno promatrati centralne frekvencije prva dva formanta, pa je već na osnovu ta dva broja (frekvencije) moguće dati grubu ocjenu o kojem se od šest samoglasnika radi. Naravno da se tu kao problem nameće inherentna varijabilnost između pojedinih govornika, tj. klasifikator može raditi ispravno za jednog govornika, ali ne radi za nekog drugog. Na žalost, varijabilnost izgovora postoji čak i za istog govornika, pa radi toga čak i klasifikator specijalno projektiran baš za njega ne mora nužno uvijek raditi pravilno. Svi ovi aspekti biti će analizirani u okviru ove vježbe na primjeru automatskog prepoznavanja samoglasnika na osnovu formantne strukture. Za razliku od toga, u okviru naredne vježbe istraživati će se varijabilnost među različitim govornicima, te će se pokušati na osnovu načina izgovora šest samoglasnika provesti automatsko prepoznavanje govornika. Iako je postupak korišten u ovoj vježbi vrlo jednostavan i prilagođen opsegu znanja studenata ovog predmeta, većina problema koja se javlja u stvarnim i mnogo složenijim sustavima za automatsko prepoznavanje biti će shvatljiva i na osnovu ovog jednostavnog primjera.

6.1 Klasifikacija samoglasnika na osnovu formantnih frekvencija

Na prošlim laboratorijskim vježbama vidjeli smo kako se dobivaju formantne karakteristike govornog signala, određivanjem frekvencijskih karakteristika LPC modela. Na ovim laboratorijskim vježbama prikazat ćemo kako je moguće na osnovu parametara formantnih karakteristika provesti automatsku klasifikaciju samoglasnika.

Na početku vježbe pokrenite Matlab, te se pozicionirajte u sljedeći direktorij:

```
»cd c:\users\dog\Vj6.
```

Naredbom »what na ekranu će se pojaviti popis sljedećih datoteka:

```
» what
```

```
M-files in the current directory C:\Users\Dog\Vj6
```

```
dog6_formanti      dog6_prikaz        voroni_regions2  
dog6_lpc_analiza   lpc_robust         wavread2
```

6.1.1 Organizacija datoteka i direktorija za ovu vježbu.

Pored glavnog direktorija u kojem se nalaze izvršne '.m' datoteke, postoje i tri pod-direktorija:

- wav_files sa '.wav' datotekama sa snimljenim izgovorima samoglasnika
- mat_files sa '.mat' datotekama, koje u stvari predstavljaju LPC parametre gornjih '.wav' datoteka i koje se koriste kao ulaz u program za određivanje formantnih karakteristika
- rez_files sa '.mat' datotekama koje sadrže obrađene parametre formantne strukture, te se koriste kod konačnih prikaza.

Obzirom da će se u okviru ove i sljedeće vježbe razmatrati već prije spomenuta varijabilnost koja postoji između istog i različitih govornika, u navedenim pod-direktorijima se nalaze datoteke za oko 20 različitih govornika i to sa dva ili više izgovora za svakog

govornika. Način označavanja datoteka je jednak za sva tri pod-direktorija i to tako da naziv datoteke ima sljedeći format:

spXX_Y

gdje je xx redni broj govornika, a y redni broj izgovora tog govornika.

6.1.2 Ulazne Wav datoteke korištene u vježbi

Pored ovih već unaprijed snimljenih i obrađenih datoteka, za potrebe ove vježbe za svakog studenta potrebna su dva pravilna izgovora niza samoglasnika 'a', 'e', 'i', 'o', 'u', '3', slično kao što se to radilo i u prethodnim laboratorijskim vježbama. U tu svrhu se mogu koristiti već prije snimljene '.wav' datoteke, pod uvjetom da sadrže pravilno snimljene samoglasnike i to u pravilnom redoslijedu. Dosta je značajno da snimka bude što čišća, tj. uz što manje pozadinske buke i šuma. Radi toga je značajno prilikom snimanja koristiti što je moguće manju osjetljivost mikrofona i čim je moguće glasniji i jasniji izgovor. Kao što je objašnjeno u prvoj laboratorijskoj vježbi u kojoj se izučavao način snimanja '.wav' datoteka (vidi slike 2.1-4 b) i 2.1-5), potrebno je u Advanced izborniku programa Volume Control u kontroli osjetljivosti mikrofonskog ulaza ugasiti +20 dB Mic Gain Control (suprotno od postavke Check-box-a prikazane na 2.1-5). Time je osjetljivost ulaza 10 puta manja, pa je potrebno govoriti glasnije i iz veće blizine. Svakako prilikom snimanja paziti da se mikrofoni **ne nalazi** direktno u zračnoj struji koja izlazi iz usta !! Obzirom da često neka od snimljenih datoteka nije upotrebljiva, poželjno je snimiti tri izgovora, pa zatim nakon obrade odabrati dva bolja. Ulazne zvučne datoteke je potrebno pohraniti u direktorij 'wav_files' pod nazivima: 'sp0_1.wav', 'sp0_2.wav' i 'sp0_3.wav', tj. kao da se radi o govorniku pod rednim brojem 0.

6.1.3 LPC analiza snimljenih datoteka

Postojeće '.wav' datoteke su već obrađene, tj. nad tim '.wav' datotekama je već provedena LPC analiza, a rezultati se nalaze u direktoriju 'mat_files' u istoimenim datotekama sa ekstenzijom '.mat'. Međutim, novo snimljene datoteke treba obraditi što se radi korištenjem programa 'dog6_LPC_analiza.m'. Ovo je program koji automatski radi LPC analizu za niz govornika i to za više izgovora svakog govornika. Glavne parametre analize moguće je podesiti editiranjem izvorne datoteke 'dog6_LPC_analiza.m'. Na vrhu programa se nalaze sljedeće postavke, koje za analizu 3 izgovora vaših samoglasnika (govornik 0) moraju biti podešene kao što slijedi:

```
prvi=0;      % redni broj prvog govornika za LPC analizu
zadnji=0;    % redni broj zadnjeg govornika za LPC analizu
broj_izg=3;  % broj izgovora za svakog govornika
pref='sp';   % prefix naziva svih file-ova
```

Varijable wav_dire i mat_dire sadrže puteve do ulaznih '.wav' datoteka, odnosno do izlaznih '.mat' datoteka s rezultatima LPC analize i moraju biti postavljene kao što slijedi:

```
wav_dire='wav_files\';
mat_dire='mat_files\';
```

Preostale postavke u programu definiraju parametre same LPC analize i treba ih podesiti na sljedeći način:

```
param(1)=200;    % Broj uzoraka frame-a
param(2)=80;     % korak frame-ova izrazen u broju uzoraka
param(3)=80;     % Donja granicna frekvecija u Hz (-3 dB)
param(4)=0.9375; % faktor za pre-emphasis ili 0 ako je bez
param(5)=10;     % LPC predictor order
param(6)=1.5;    % Faktor c
param(7)=2;      % Broj iteracija
param(8)=10;     % BW exp
```

```
param(9)=20; % minimalni BW
```

Uočite da je kao red LPC analize p odabran 10. red prediktora, što je u slučaju frekvencije otipkavanja od $f_s=8$ KHz dovoljno i kod muških govornika s relativno dugačkim vokalnim traktom. Za većinu drugih govornika, ovakav odabir je i previše konzervativan, pa će se zbog činjenice da je $p/2$ veći od stvarnog broja formanata u rasponu od 0 do $f_s/2$ pojaviti i lažni formanti. Ovaj problem je moguće riješiti pravilnim odabirom reda prediktora nezavisno za svakog govornika, ili kao što će biti pokazano u nastavku ove vježbe posebnim postupcima razdvajanja pravih i lažnih formanata.

Nakon podešavanja svih parametara kao što je objašnjeno, te nakon uspješnog izvođenja programa 'dog6_LPC_analiza.m' u direktoriju 'mat_files' se moraju nalaziti tri novo obrađene '.mat' datoteke pod nazivima 'sp0_1.mat', 'sp0_2.mat' i 'sp0_3.mat', koje će biti ulaz u slijedeće programe korištene na ovoj vježbi.

6.2 Određivanje formantne strukture i ručno označavanje formanata

U okviru pripremnog dijela vježbe svi studenti će prvo napraviti formantnu analizu samoglasnika istog govornika pod rednim brojem 2 i to njegovog prvog izgovora. Obzirom da se obrađene '.mat' datoteke s parametrima LPC analize navedenog govornika već nalaze u direktoriju 'mat_files' nije potrebno ponovno provoditi LPC analizu kao što je objašnjeno u prošlom poglavlju, već treba kao ulazne datoteke iskoristiti postojeće.

Program za formantnu analizu se zove 'dog6_formanti.m' i nalazi se u glavnom direktoriju ove vježbe. Glavne postavke koje je potrebno podesiti editiranjem datoteke 'dog6_formanti.m' prije njenog izvođenja su slijedeće:

```
prvi=2; % Indeks prvog 'mat' file-a koji ce biti
        % obradjivan
zadnji=2; % Indeks zadnjeg 'mat' file-a koji ce biti
        % obradjivan
broj_izg=1; % Broj izgovora za svakog govornika
pref='sp'; % prefix file-a koji se koristi u svim folderima

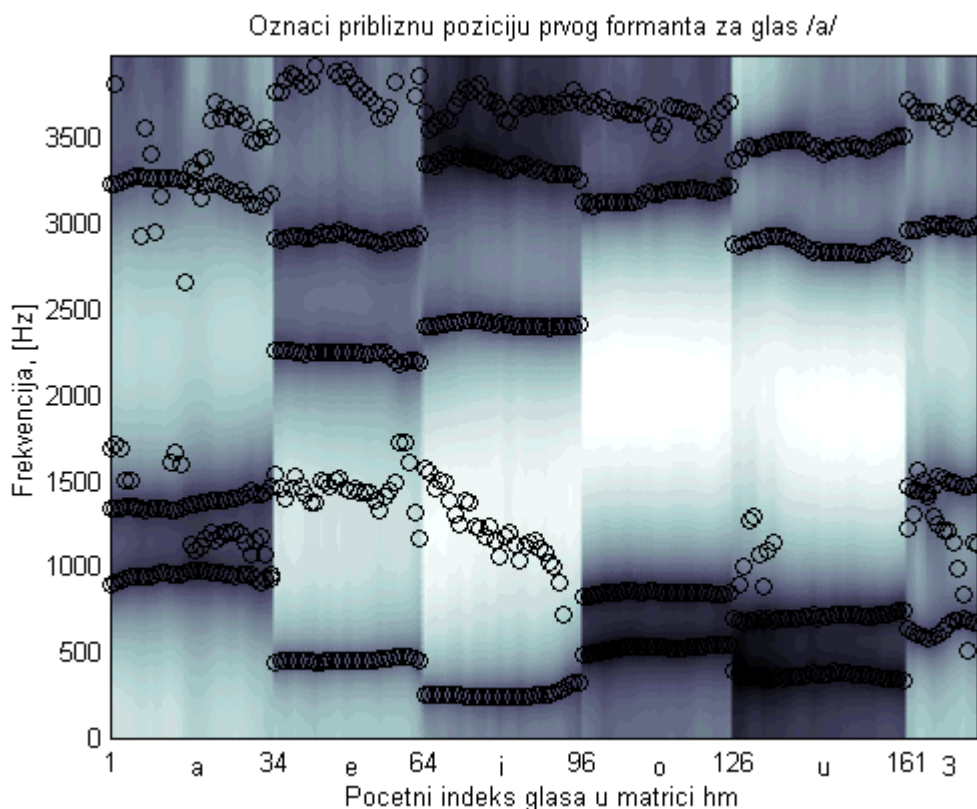
mat_dire='mat_files\'; % Folder s 'mat' file-ovima s rezultatima LPC
        % analize
rez_dire='rez_files\'; % Folder s 'mat' file-ovima s rezultatima
        % formantne analize, tj, sa frekvencijama
        % izdvojenih formanata

sel_for=3; % Broj formanata koji ce biti rucno izdvojen
faktor=2.5; % Faktor kojim se dijeli frek. drugog formanta
        % kod prikaza
```

Osim zadnje dvije, sve ostale varijable su jednake kao i u programu za LPC analizu, tj. određuju raspon govornika koji će biti obrađivan, kao i broj izgovora po svakom govorniku. Značenje zadnje dvije postavke biti će objašnjenje naknadno, a potrebno ih je podesiti kao što je gore navedeno.

Prvi dio izvođenja programa 'dog6_formanti.m' je vrlo sličan programu 'dog5.m' koji je korišten u prošloj vježbi. To znači da je potrebno ručno označiti početke i krajeve izgovora svakog od 6 samoglasnika kao što je prikazano na slici 5.2-2 za slučaj jedne druge govorne datoteke. Nakon ovog označavanja, program automatski izračunava frekvencijske karakteristike LPC modela izdvojenih samoglasnika, te određuje korijene polinoma $A(z)$ kao što je opisano u okviru prošle vježbe. Nakon toga, program će prikazati spektrogram LPC ovojnice spektra izdvojenih samoglasnika sa ucrtanim pozicijama (frekvencijama)

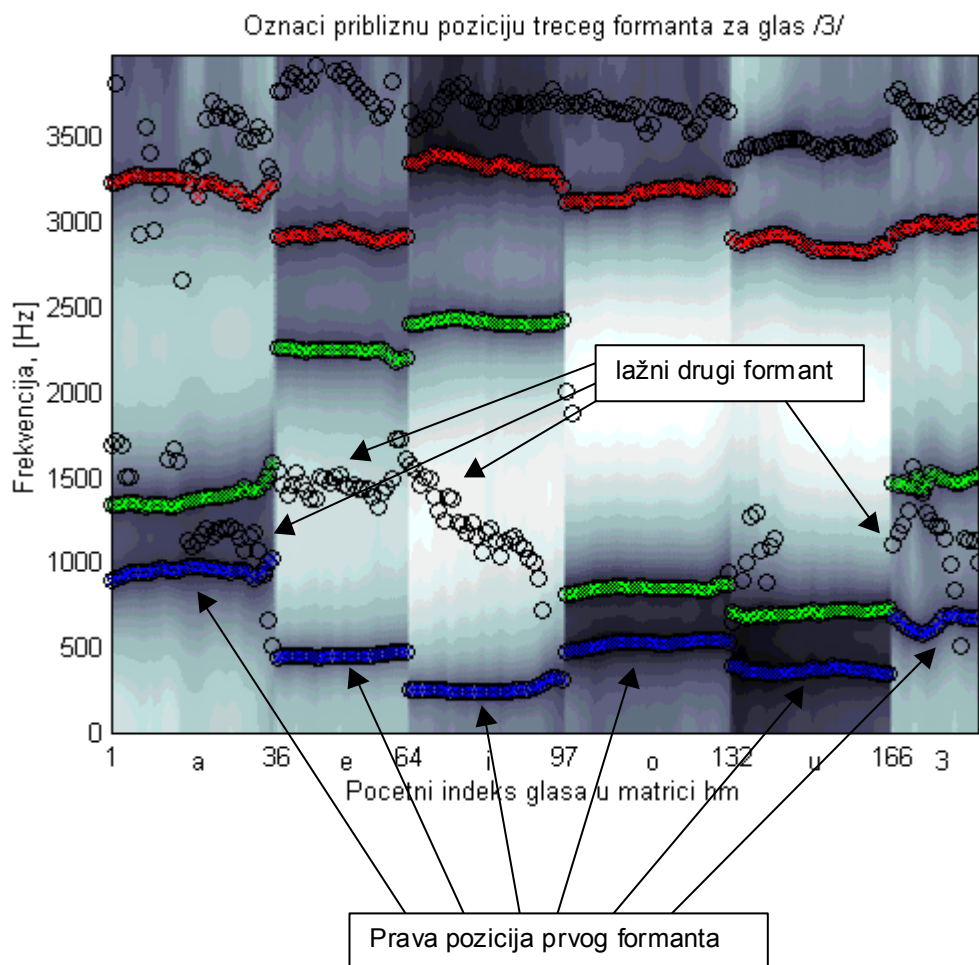
konjugirano kompleksnih korijena polinoma $A(z)$, kao što je to prikazano na slici 6.2-1 za slučaj prvog izgovora drugog govornika Samo neki od ucrtanih korijena polinoma $A(z)$ predstavljaju prave formante, dok su preostali korijeni ili realni (na 0 ili $f_s/2$) ili su lažni formanti koji su prisutni samo zato što je red modela previsok. Govornik na slici je ženski govornik koji ima između 3 i 4 formantne karakteristike u frekvencijskom pojasu između 0 i $f_s/2$, dok je prediktor 10-tog reda, pa samim time može modelirati čak do pet rezonantnih karakteristika u istom pojasu frekvencija. Tako npr. glas 'a' ima samo tri prava formanta i to na frekvencijama od približno 950 Hz, 1400 Hz i 3150 Hz. Preostali korijeni prikazani za glas 'a' su lažni i ne pripadaju niti jednom formantu. Slično tome se u glasovima 'e' i 'i' pojavljuju lažni drugi formanti na frekvencijama 1500 Hz za 'e' i između 1600 i 1000 Hz za glas 'i'. Obično sortiranje po rastućim frekvencijama bi navedene korijene proglasilo pravim drugim formantima, pa je nužno na neki način odrediti pozicije pravih formanata. Na žalost, vrlo je teško navedeni postupak napraviti automatski, tj. program koji bi to radio samostalno bi bio previše složen i ne bi bio ilustrativan za potrebe ovih vježbi. Radi toga, odabir pravih formanata biti će proveden 'ručno' sekvencijalnim označavanjem približnih pozicija svih pravih formanta za svaki glas nezavisno.



Slika 6.2-1 LPC spektralna ovojnica svih šest samoglasnika s ucrtanim frekvencijama korijena nazivnika LPC modela

U tu svrhu je dovoljno mišem kliknuti na mjestu koje odgovara približnoj poziciji pravog formanta. Nakon toga će program napraviti automatsko pretraživanje, te kao prave formante odabrati one korijene polinoma koje se nalaze najbliže frekvenciji (visini) na kojoj je korisnik kliknuo. Nakon toga, odabrani korijeni će biti označeni bojom (plavo za 1., zeleno za 2. i crveno za 3. formant). Važno je naglasiti da će korijeni označeni kao prvi formant automatski biti izbačeni iz mogućeg odabira prilikom određivanja svih viših formanata. Opisani postupak se provodi prvo za prvi formant i to za svih 6 samoglasnika. Nakon toga se isti postupak

ponavlja i za drugi pa treći formant svih samoglasnika. U naslovu slike program automatski ispisuje koji formant kojeg glasa trenutno treba označiti. Važno je napomenuti da pozicija po x -koordinati gdje treba kliknuti prilikom označavanja mora biti unutar raspona indeksa koji pripadaju tom glasu. Ako se klikne van tog raspona, program će to ignorirati i čekati tako dugo dok se ne klikne na pravo mjesto. Na slici 6.2-2 je prikazan izgled slike nakon pravilnog označavanja svih 18 formantnih krivulja. Varijabla `sel_for` određuje koliko se formantnih karakteristika izdvaja opisanim postupkom. U slučaju frekvencije otipkavanja od 8kHz, praktički za sve govornike i sve samoglasnike postoje barem tri formanta u pojasu od 0 do $f_s/2$, pa je radi toga odabrano `sel_for=3`.



Slika 6.2-2 LPC spektralna ovojnica svih šest samoglasnika nakon ručnog markiranja pozicija pravih formanata

6.2.1 Preporuke za uspješno označavanje formanata

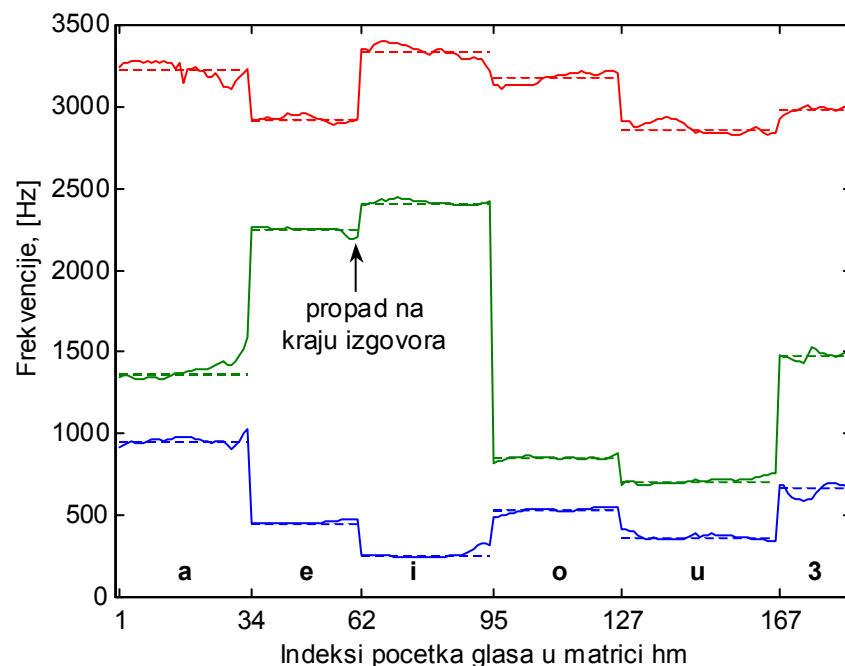
Zbog velike varijabilnosti formantnih karakteristika između pojedinih govornika, kao i zbog relativno velike varijabilnosti prilikom samog izgovora, teško je postaviti oštra pravila odabira koja bi razlučila prave od lažnih formanata. Radi toga, kao inspiraciju koristite sliku 6.2-2. Za muške govornike ova slika je komprimirana po frekvencijskoj osi, tj. očekivane pozicije formanata su na nešto nižim frekvencijama. Kao jedan od dobrih pokazatelja da se radi o pravom formantu koristite slijedeće pokazatelje:

- krivulja pravog formanta je sporo promjenjiva iz okvira u okvir, tj. javlja se kao horizontalna linija s blagim nagibom ili sporim varijacijama,
- krivulja se mora nalaziti u području spektrograma koji ima veliku amplitudu (u crnom području), a ne u spektralnoj dolini (bijelo područje)
- krivulja mora biti neprekinuta tokom cijelog izgovora dotičnog samoglasnika, tj. mora postojati od ruba do ruba
- u pravilu za više formante (3. i 4.) navedena pravila ne stoje više tako strogo, tj. može se očekivati veća varijabilnost u krivuljama
- kod nekih govornika glas 'u' može imati samo jedan formant ispod 2kHz što nije ispravno, tj. radi se o dva vrlo bliska 'prava' formanta koji se ponekad zbog svoje blizine stapaju u jedan. Radi toga, čim god je to moguće, pokušajte naći oba formanta glasa 'u' i to tako da se oba nalaze ispod frekvencije od 1kHz,
- iako vas program u tome ne sprječava, formante uvijek označujte u rastućem slijedu od nižih prema višim, tj. nakon što je krivuljom u boji označen prvi formant, drugi se može nalaziti samo iznad ove krivulje,
- ako označavate formant koji ima veću varijabilnost po frekvenciji, pokušajte mišem kliknuti na visini koja približno odgovara srednjoj frekvenciji tog formanta, jer će tada program automatski odabrati korijene koji su najbliži toj odabranoj frekvenciji, pa će i rezultirajuća krivulja biti 'najmirnija'.

Ovo su neke generalne preporuke koje će vam olakšati ispravno označavanje formanta, no na žalost vještina u ovome se postiže isključivo praksom i iskustvom.

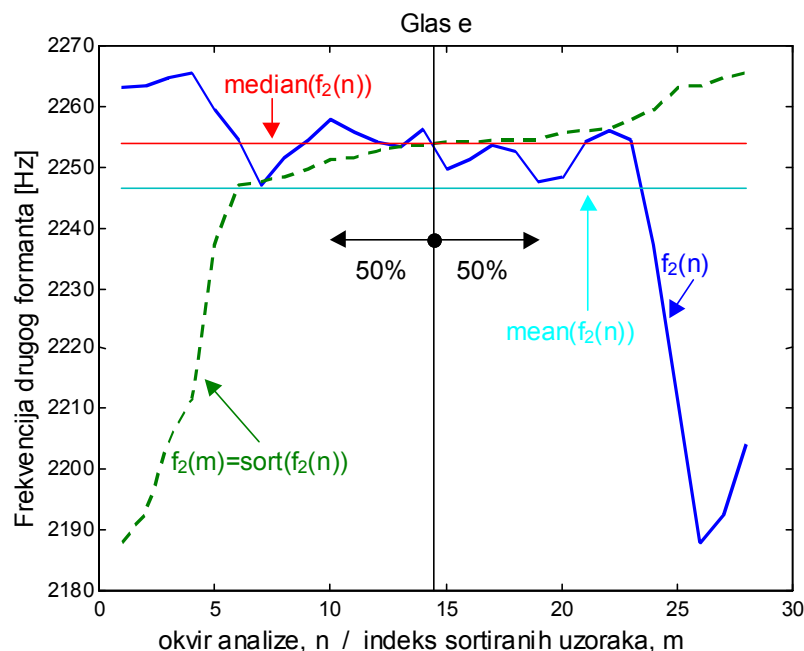
6.3 Prikaz u f_1/f_2 ravnini i klasifikacija temeljena na najbližem susjedu

Nakon uspješno provedenog označavanja formantnih frekvencija, program prikazuje određene formantne krivulje kao što je prikazano na slici 6.3-1.



Slika 6.3-1 Centralne frekvencije prva tri ručno markirana formanta i njihovi medijani (crtkanom linijom) za svih 6 samoglasnika

Obzirom da su u ovom prikazu izdvojeni samo 'aktivni' dijelovi svakog samoglasnika, tj. izbačene su pauze između pojedinih glasova, prikaz dobiva karakteristični 'stepeničasti' oblik, tj. sve tri formantne frekvencije su manje ili više konstante tokom cijelog izgovora glasa. Radi toga, interesantno je odrediti statistička očekivanja ovih frekvencija za svaki pojedini glas. U tu svrhu se može koristiti klasično definirana srednja vrijednost, ili pak bilo koji drugi postupak za estimaciju statističkih očekivanja. Problem sa srednjom vrijednosti jest u njenoj ovisnosti o iznimkama, tj. svega nekoliko iznimki s vrijednošću značajno različitom do većine preostalih uzoraka može 'pomaknuti' estimaciju očekivanja na krivo mjesto. Opisano svojstvo je ilustrirano na slici 6.3-2, na kojoj je punom plavom linijom označena promjena centralne frekvencije drugog formanta $f_2(n)$ prilikom izgovora glasa 'e'. Kao što je već naznačeno na slici 6.3-1, a još se i bolje vidi na slici 6.3-2, pred kraj izgovora ovog glasa dolazi do propada frekvencije $f_2(n)$. Obična srednja vrijednost ovih 28 uzoraka prikazana je svjetlo plavom točkastom linijom označenom sa $\text{mean}(f_2(n))$ na slici 6.3-2, koja se nalazi ispod praktički svih uzoraka signala. Već je na prvi pogled očito je da gornja (crvena točkasta) linija mnogo bolja estimacija statističkog očekivanja. Ova vrijednost označena sa $\text{median}(f_2(n))$, određena je kao *median* funkcija ulaznih uzoraka $f_2(n)$, a ta je definirana kao centralna vrijednost sortiranog niza uzoraka. Zbog toga se 50% uzoraka nalazi ispod te vrijednosti (donja polovica sortirane liste), dok se preostalih 50% uzoraka nalazi iznad vrijednosti mediana. Način određivanja mediana je prikazan na slici 6.3-2, gdje je debelom crtkanom zelenom linijom prikazan sortiran niz uzoraka $f_2(m)=\text{sort}(f_2(n))$, a sa vertikalnom točkastom crnom linijom označen središnji indeks sortiranog niza.



Slika 6.3-2 Ilustracija primjene funkcije median za određivanje reprezentanata samoglasnika

Primjenom median funkcije dobiva se robusna estimacija statističkog očekivanja, koja je 'imuna' na iznimke, a korišten odsječak Matlab koda za njeno određivanje u programu 'dog6_formanti.m' je dan u nastavku:

```
% Usrednjavanje pozicija formanta za svaki pojedini glas
% provedeno je korištenjem funkcije median, koja je mnogo
% robusnija nego obična srednja vrijednost u slučaju kada
% u signalu postoje iznimke.
```

```
fo=zeros(bg,sel_for);      % Median za svaki glas i svaki formant
for i=1:bg,
    fo(i,:)=median(fr(po(i):za(i),:));
end;
```

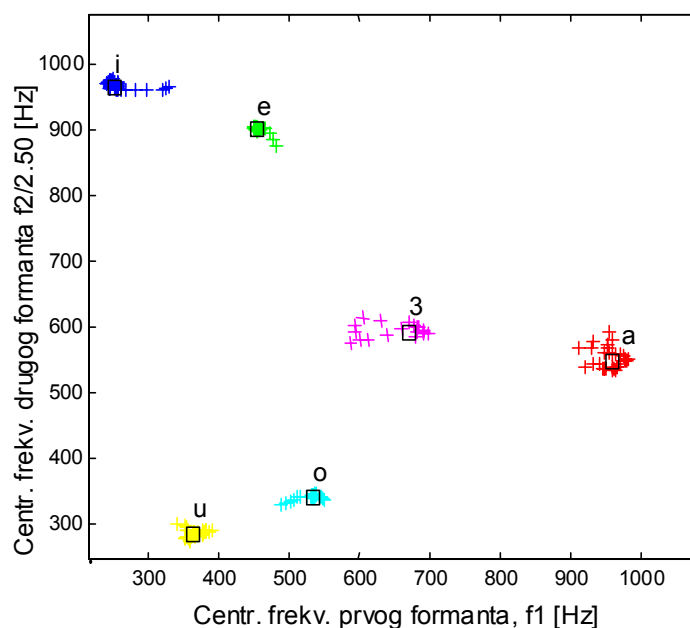
Na slici 6.3-1 su pored krivulja prve tri formante frekvencije crtanim linijama prikazani njihovi mediani za svih šest samoglasnika. Opisanim postupkom je svakom samoglasniku pridijeljen jedan tro-dimenzionalni vektor $[f_1, f_2, f_3]^T$, koji parametarski opisuje pripadni glas. Program ispisuje izračunate mediane i u obliku tablice što za odabranog govornika daje sljedeće vrijednosti:

Median za svaki formant i svaki glas			
glas	f1	f2	f3
a	958	1371	3248
e	455	2254	2926
i	252	2414	3343
o	533	853	3179
u	364	716	2855
3	672	1482	2987

Napomena: konkretne vrijednosti prilikom vašeg izvršavanja ne moraju nužno biti identički jednake navedenima, jer ovise o ručnom izdvajanju samoglasnika.

6.3.1 Prikaz u f_1/f_2 ravnini

Obzirom da se vektori mediana centralnih frekvencija formanta značajno razlikuju od glasa do glasa, kao jednostavna ideja za postupak klasifikacije samoglasnika se nameće ideja temeljena na određivanju pozicije glasa u tom tro-dimenzionalnom prostoru, tj. na temelju određivanja njegove udaljenosti od očekivanih pozicija šest samoglasnika. Radi jednostavnijeg i ilustrativnijeg prikaza, u našem primjeru su kao reprezentanti samoglasnika odabrane samo prve dvije komponente tog vektora, tj. samo prve dvije formantne frekvencije. Radi toga svakom glasu pripada po jedna točka u f_1/f_2 ravnini.



Slika 6.3-3 Položaj samoglasnika u f_1/f_2 ravnini

Na slici 6.3-3 su mediani svakog samoglasnika prikazani crnim kvadratićima, dok svaki križić odgovara jednom okviru analize. Boja križića je određena pripadnošću tog okvira pojedinom glasu određenom početnom manualnom segmentacijom glasova, tj. svi okviri koji pripadaju glasu 'a' su crvene boje, zatim svi koji pripadaju glasu 'e' su zelene boje itd. Iz ove slike je uistinu vidljivo da se točke koje pripadaju istom glasu grupiraju oko točke mediana tog glasa i da je rasipanje relativno malo. Obratite pažnju da je y -os skalirana, tj. iz razloga da bi se dobio približno isti raspon vrijednosti po obje osi, kao y -os je odabrana druga formantna frekvencija podijeljena sa faktorom skale $f_{\text{faktor}}=2.5$. Slijedeći korak jest projektiranje klasifikatora koji će automatski na osnovi pozicije točke u f_1/f_2 ravnini odrediti o kojem se samoglasniku radi.

6.3.2 Klasifikator za automatsko prepoznavanje samoglasnika

Jedan od vrlo poznatih i često korištenih postupaka klasifikacije jest postupak najbližeg susjeda (engl. nearest neighbor). Ako se ulazni vektorski proces klasificira u N klasa C_1, C_2, \dots, C_N , tada je svakoj klasi pridijeljen jedan vektor reprezentant $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$. Za ulazni vektor \mathbf{x} se kaže da pripada klasi C_i ako je udaljenost tog ulaznog vektora do reprezentanta klase C_i označenog sa \mathbf{x}_i manja od udaljenosti do bilo kojeg drugog reprezentanta \mathbf{x}_j , tj.

$$\mathbf{x} \in C_i \Leftrightarrow d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j) \forall j \in [1, 2, \dots, N], j \neq i \quad (6.3-1)$$

Funkcija $d(\mathbf{x}, \mathbf{y})$ može biti bilo koja funkcija koja ima svojstva koja mora zadovoljavati funkcija udaljenosti. Međutim, zbog jednostavnosti se vrlo često koristi obična Euklidska udaljenost, ili pak težinska Euklidska udaljenost, koja je za slučaj općenitih težina definirana kao:

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \cdot \mathbf{W} \cdot (\mathbf{x} - \mathbf{y}) \quad (6.3-2)$$

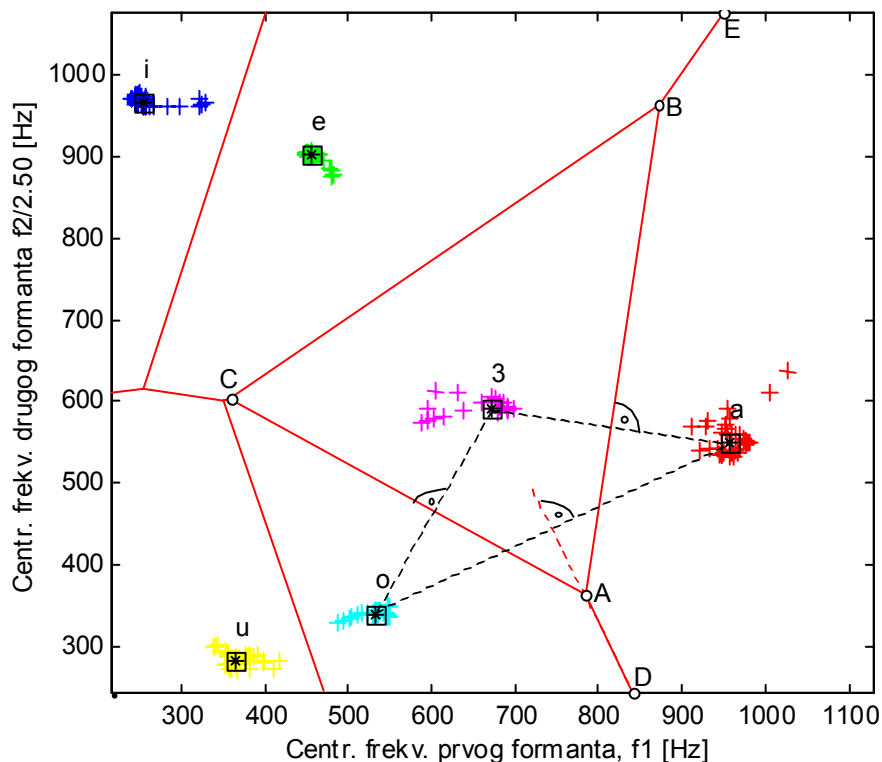
gdje je \mathbf{W} težinska matrica, a sa T je označena transpozicija. Nužan uvjet da bi funkcija $d(\mathbf{x}, \mathbf{y})$ bila funkcija udaljenosti jest da matrica \mathbf{W} bude pozitivno definitna matrica, tj. da su joj sve vlastite vrijednosti strogo veće od nule. Za slučaj obične Euklidske udaljenosti, težinska matrica \mathbf{W} postaje obična jedinična matrica.

Na slici 6.3-4 su prikazani isti samoglasnici sa slike 6.3-3, samo što su crvenim linijama do crtane granice pojedinih klasa, gdje su te klase definirane dvo-dimenzionalnim vektorima reprezentanata svakog glasa, tj. vektorima mediana prve dvije formantne frekvencije. Korištena mjera udaljenosti je obična Euklidska udaljenost u $f_1 / (f_2/2.5)$ ravnini. Granice klase su definirane simetralama između reprezentantnih točaka. Tako npr., granica između glasa 'o' i glasa '3' je definirana simetralom \overline{AC} koja je okomita na spojnicu reprezentanata tih glasova $\overline{O3}$. Slično vrijedi za granicu glasova '3' i 'a' koja je definirana simetralom \overline{AB} . U oba slučaja segment simetrale koji predstavlja granicu klase se sječe sa spojnicom reprezentanata. Za razliku od navedenih primjera, granica glasova 'o' i 'a' je određena simetralom \overline{AD} koja izvire iz 'tromeđe' u točki A, tj. točka A je jednako udaljena od reprezentanta sva tri glasa 'o', '3' i 'a'. U ovom slučaju se spojnica \overline{Oa} siječe s njihovom simetralom izvan njenog segmenta koji predstavlja granicu, tj. na 'nevidljivom' dijelu simetrale koji je naznačen crtkanom linijom.

Na opisani način su dobivene klase ili regije koje pripadaju svakom samoglasniku i koje su geometrijski opisane poligonima. Neki od tih poligona su zatvoreni, kao npr. trokut ABC koji definira klasu za glas '3', dok su drugi otvoreni, tj. imaju granicu samo s jedne strane, dok su s druge strane neograničeni, tj. šire se do $\pm\infty$ po bilo kojoj osi. Naravno u našem primjeru

ti su poligoni ipak ograničeni i to s donje strane sa nulom, a s gornje strane sa $f_s/2$, tj. $0 < f_1 < f_s/2$ i $0 < f_2 < f_s/2$. Poligoni koji definiraju pojedine klase se još nazivaju Voronoi regijama. Prikaz granica klasa se ostvaruje funkcijom 'voroni_regions2.m' koja se nalazi u glavnim direktoriju vježbe.

Na slici 6.3-4 je vidljivo da svi okviri za koje je manualnom segmentacijom određeno da pripadaju pojedinom samoglasniku (tj. imaju istu boju na slici) uistinu padaju 'duboko' unutar Voronoi regije (klase) koja pripada tom glasu. To znači da je klasifikator temeljen na najbližem reprezentantu u $f_1 / (f_2/2.5)$ ravni vrlo uspješan za automatsku klasifikaciju samoglasnika. Takvo ponašanje se moglo i očekivati, obzirom da je klasifikator projektiran direktno na osnovu parametara formantnih karakteristika samoglasnika baš tog govornika.



Slika 6.3-4 Voronoi regije definirane reprezentantima samoglasnika u f_1/f_2 ravni uz korištenje Euklidske udaljenosti

Postupak klasifikacije temeljene na principu najbližeg susjeda se koristi i u postupcima vektorske kvantizacije. U tim primjenama ulazni vektor sa komponentama beskonačne preciznosti se zamjenjuje najbližim vektorom reprezentantom, te se na prijemnu stranu šalje samo indeks tog 'najboljeg' reprezentanta u kodnoj knjizi reprezentanata. Njegova rekonstrukcija na prijemnoj strani se provodi jednostavnim 'isčitavanjem' reprezentanta sa adrese kodne knjige koja odgovara tom indeksu. Postupak vektorske kvantizacije je vrlo značajan, jer umjesto da se svaka komponenta vektora kvantizira nezavisno kao skalarna vrijednost, cijeli vektor se kvantizira kao jedna 'jedinka', čime se ostvaruju značajne uštede u potrebnog brzini prijenosa.

6.3.3 Korištenje univerzalnog klasifikatora

Do sada su parametri korištenog klasifikatora bili određeni direktno iz snimljenih podataka za jedan izgovor dotičnog govornika. Međutim, za stvarne primjene interesantnije je koristiti univerzalni klasifikator koji bi trebao osigurati jednako uspješnu klasifikaciju svih

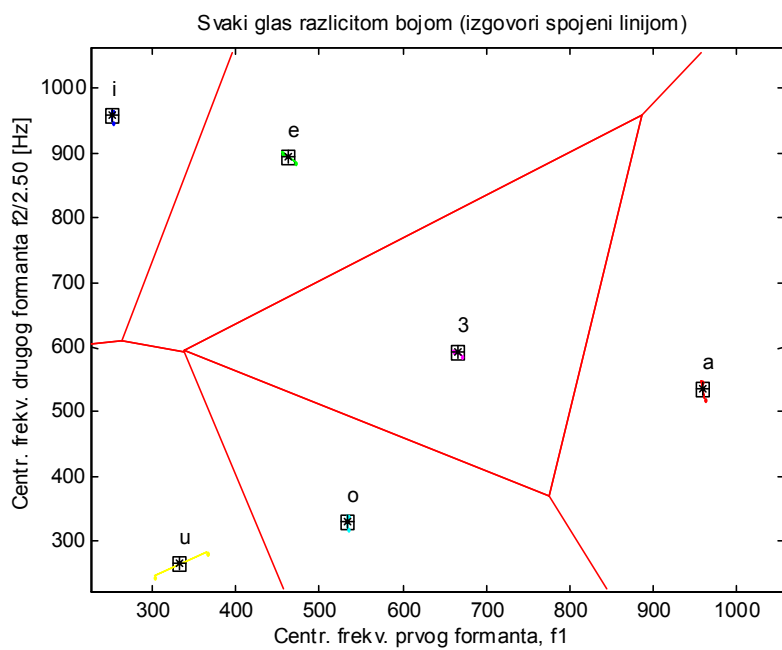
samoglasnika i to za sve govornike. Drugi interesantni problem jest pitanje kolika je ponovljivost rezultata za istog govornika, tj. koliko će biti rasipanje mediana formantnih frekvencija samoglasnika u dva potpuno odvojena izgovora istog govornika. Oba navedena problema moguće je istražiti korištenjem dodatnog programa 'dog6_prikaz.m' koji služi za zbirni prikaz rezultata određenih u neposredno objašnjenom programu za formantnu analizu 'dog6_formanti.m'. Glavna namjena ovog programa je usporedni prikaz rezultata za više govornika, no u okviru ove vježbe biti će korišten isključivo za prikaz više izgovora istog govornika. Ulaz u ovaj program su '.mat' datoteke koje se nalaze u 'rez_files\' direktoriju, a koje sadrže rezultate formantne analize. Parametri koje je potrebno podesiti prije poziva ovog programa su definirani u njegovom zaglavlju:

```
faktor=2.5;           % Faktor za dijeljenje druge formantne
                      % frekvencije u prikazima
rez_dire='rez_files\'; % Direktorij s obradjenim rezultatima
                      % formantne analize

odabrani=[2];         % Indeksi odabranih govornika za prikaz

sel=20;              % redni broj govornika kojeg se zeli istaknuti na prikazima
pref='sp';           % prefix naziva datoteka sa obradjenim rezultatima
broj_izg=2;          % broj izgovora po svakom odabranom govorniku
sel_for=3;           % broj izdvojenih formanata u postupku analize
fix_class=0;         % Odabir klasifikatora samoglasnika:
                      %   0 = klasifikator odredjen na osnovu
                      %       odabranih govornika, usrednjavanjem
                      %       svih njihovih izgovora
                      %   1 = fixni klasifikator odredjen na
                      %       osnovu velike baze od 14 govornika
```

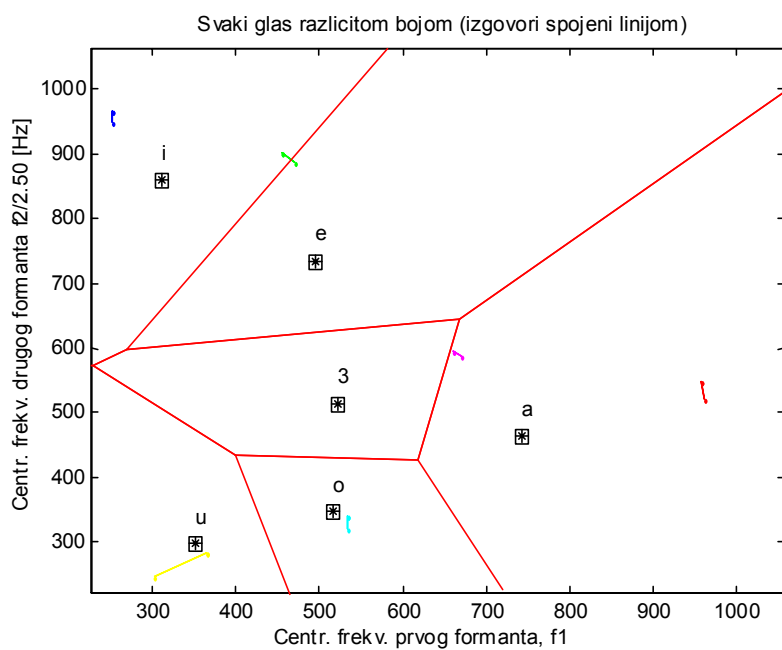
U retku 'odabrani' se nalazi niz indeksa govornika čiji će rezultati biti prikazivani. U pripremnom dijelu vježbe, potrebno je provesti prikaz za govornika sa rednim brojem 2, kao što je naznačeno u gornjem kodu. Varijabla 'sel' nije značajna u slučaju kada se prikazuju rezultati za samo jednog govornika. Varijabla 'broj_izg' definira broj obrađenih izgovora za svakog govornika, dok varijabla 'sel_for' definira broj ručno izdvojenih formanata u postupku formantne analize i mora odgovarati rezultatima pohranjenim u '.mat' datotekama. Zadnja varijabla određuje koji će se klasifikator koristiti kod zbirnog prikaza rezultata.



Slika 6.3-5 Voronoi regije za drugog govornika definirane usrednjavanjem rezultata dvaju izgovora tog govornika

U prvom slučaju program je izvršen s ulaznim parametrima kao što je gore navedeno, tj. sa Voronoi regijama određenim direktnim usrednjavanjem svih izgovora tog odabranog govornika (`fix_class=0`). Na slici 6.3-5 su prikazani konačni rezultati za dva izgovora drugog govornika. Svaki izgovor je označen malom točkom, a izgovori istog glasa su spojeni linijom. Točkice i linija su različitih boja za svaki glas. Reprezentanti koji definiraju Voronoi regije označeni su crnom zvjezdicom u kvadratiću, a određeni su usrednjavanjem formantnih frekvencija ta dva izgovora. Osim za glas 'u' za koji postoji nešto veća razlika između dva izgovora, svi ostali glasovi su praktički na istim pozicijama u ta dva izgovora i gotovo da su potpuno prekriveni simbolom reprezentanta regije.

U slučaju korištenja fiksnog klasifikatora (`fix_class=0`), čiji su parametri određeni usrednjavanjem rezultata za 13 muških i 1 ženskog govornika dobiva se slika 6.3-6



Slika 6.3-6 Voronoi regije za drugog govornika definirane 'univerzalnim' reprezentantima određenim na osnovu veće baze

Vidljivo je da je klasifikacija temeljena na ovakvom 'univerzalnom' klasifikatoru mnogo lošija, tj. oba izgovora glasa '3' označenog ljubičastom bojom su 'pobjegla' u regiju glasa 'a', dok je jedan izgovor glasa 'e' (zeleni boja) pobjegao u regiju glasa 'i'. Razlog za ovako lošu klasifikaciju leži u činjenici što je 'univerzalni' klasifikator projektiran na osnovu rezultata praktički isključivo muških govornika, dok se prikazani rezultati odnosi na govornika pod rednim brojem 2 koji je ženski govornik.

6.4 Praktični dio laboratorijske vježbe

Zadatak 1.

a) Provesti LPC analizu vaših izgovora samoglasnika kao što je opisano u pripremnom dijelu vježbe. Vaše datoteke numerirati sa rednim brojem 0. (sp0_1, sp0_2 i sp0_3).

b) Provesti formantnu analizu sa sva tri vaša izgovora primjenom programa 'dog6_formanti.m', te od ta tri izgovora odabrati dva koji imaju najčišću formantnu strukturu. Dva odabrana izgovora i sve pripadne datoteke (wav, mat i rez) preimenovati tako da dva najbolja izgovora imaju indekse izgovora 1 i 2.

U izlaznu MS Word datoteku pohraniti sve izlazne slike, te ih komentirati.

- a) U tablice unesi vrijednosti formantnih frekvencija za svaki samoglasnik (vrijednosti su ispisane u MATLAB Command Window)

Tablica 6.4-1 Frekvencije formanata (prvi izgovor)

glas	f1	f2	f3
a			
e			
i			
o			
u			
3			

Tablica 6.4-2 Frekvencije formanata (drugi izgovor)

glas	f1	f2	f3
a			
e			
i			
o			
u			
3			

Zadatak 2.:

a) Nakon uspješno provedene formantne analize svih vaših izgovora samoglasnika, primjenom programa 'dog6_prikaz.m' prikazati zbirne rezultate za sve vaše pravilne izgovore. Provesti prikaze za oba slučaja klasifikatora (fiksno, kao i klasifikatora određenog na osnovu vaših podataka).

b) U tablicu 6.4-3 upišite koji su samoglasnici prepoznati (u koju su regiju pali) za oba slučaja klasifikatora

Tablica 6.4-3 Prepoznati samoglasnici

stvarno izgovoreni glas	prepoznati samoglasnik	
	prvi izgovor	drugi izgovor
a		
e		
i		
o		
u		
3		

7. AUTOMATSKA KLASIFIKACIJA GOVORNIKA NA OSNOVU FORMANTNE STRUKTURE

Kao što je već najavljeno u okviru prošle vježbe, u ovoj vježbi će se razmatrati varijabilnost formantnih karakteristika izgovora različitih govornika, te će se ta varijabilnost pokušati iskoristiti u svrhu automatske klasifikacije govornika. Klasifikacija se uvijek provodi na osnovu određenih značajki signala koje su najčešće predstavljene s vektorom značajki (eng. feature vector) za svakog govornika. U ovoj vježbi će kao vektor značajki biti korišten vektor centralnih frekvencija prva tri formanta pri izgovoru šest samoglasnika u hrvatskom jeziku. Prema tome radi se o jednom 18 dimenzionalnom vektoru (3 formanta puta 6 glasova), pa će se i klasifikacija provoditi u takvom 18-dimenzionalnom prostoru. Dakle u svrhu identifikacije govornika, koristiti će se način izgovora svih šest samoglasnika. Prilikom prepoznavanja samoglasnika korištene su samo prve dvije formantne frekvencije, jer je njihova zavisnost o govorniku manja nego za više formantne frekvencije. Za razliku od toga, prilikom prepoznavanja govornika koristiti ćemo sve tri frekvencije, jer sada upravo želimo iskoristiti tu varijabilnost izgovora između različitih govornika.

7.1 Analiza varijabilnosti formantne strukture istog i različitih govornika

U prvom dijelu ove vježbe razmatrati ćemo varijabilnosti formantne strukture koje postoje između različitih govornika. Za potrebe ove vježbe, koristi se isti program koji se koristio u drugom dijelu prethodne vježbe, tj. radi se o programu 'dog6_prikaz.m'. Ulazni parametri koje je potrebno podesiti prije izvođenja programa već su objašnjeni u poglavlju 6.3.3. Jedina razlika je u tome što je sada umjesto jednog govornika potrebno u vektoru 'odabrani' odabrati željeni niz govornika. U nastavku će izvođenje programa biti ilustrirano na primjeru pet govornika s identifikacijskim brojevima: 14, 15, 16, 17 i 20. Za svih ovih 5 govornika postoje 3 izgovora s pravilnom formantnom strukturom, pa parametre programa možete podesiti na slijedeći način:

```
odabrani=[14 15 16 17 20]; % Indeksi odabranih govornika za prikaz  
broj_izg=3; % broj izgovora po svakom odabranom govorniku
```

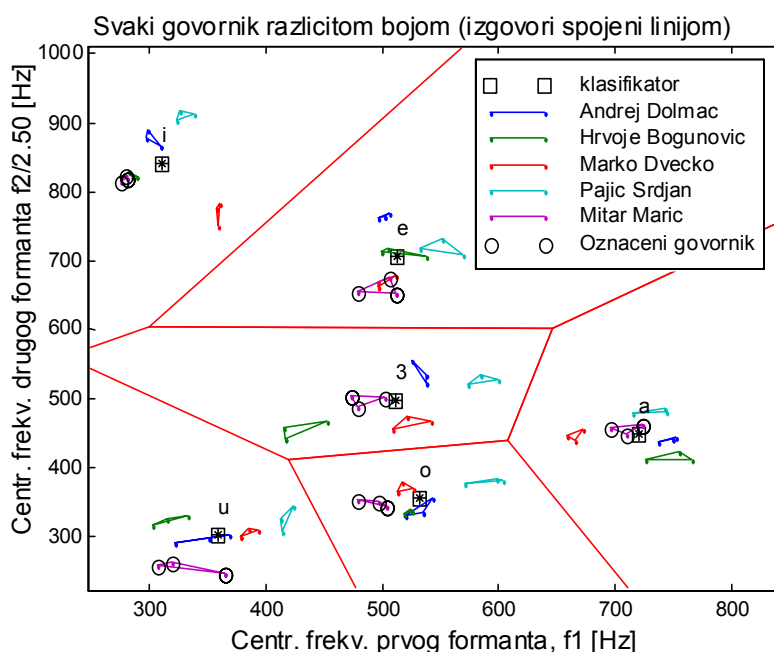
Preostale parametre definirajte na isti način kao što je prikazano u poglavlju 6.3.3. U početnom dijelu programa 'dog6_prikaz.m', rezultati formantne analize odabranih govornika se učitavaju iz direktorija 'rez_files\', te se medijani prve tri formantne frekvencije za sve glasove pohranjuju u veliku 4-dimenzionalnu matricu `foo`, koja je alocirana i inicijalizirana u nulu na slijedeći način:

```
foo=zeros(bg,sel_for,broj_gov,broj_izg);
```

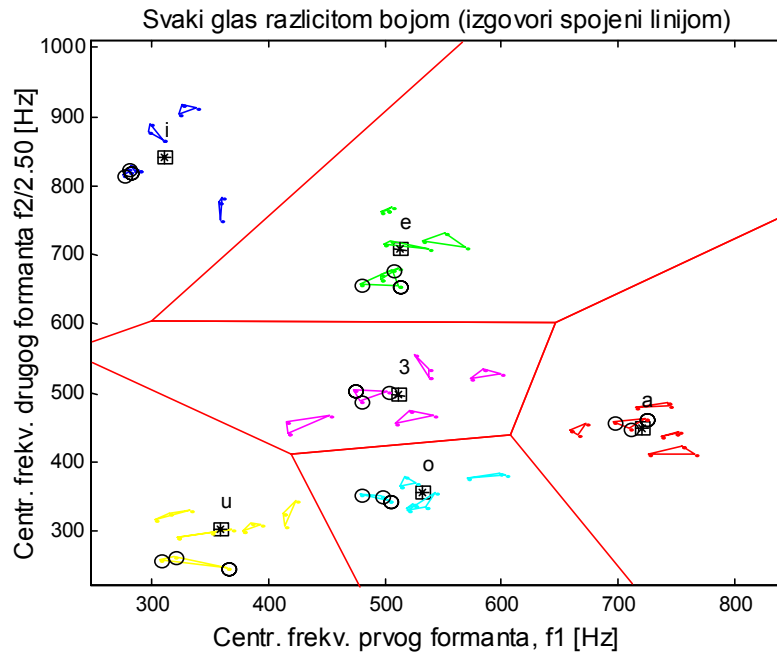
Prva dimenzija matrice je indeks glasa od 1 do `bg=6`, druga dimenzija je indeks formanta od 1 do `sel_for=3`, dok je treća dimenzija indeks govornika unutar vektora odabranih govornika 'odabrani' koji se kreće od 1 do `broj_gov`. Vrijednost varijable `broj_gov` je jednaka dimenzija vektora 'odabrani' tj. broju korištenih govornika, koji u ovom primjeru iznosi 5. Konačno četvrta dimenzija je indeks izgovora od 1 do `broj_izg=3`. U toku učitavanja rezultata, program za svakog govornika i svaki izgovor prikazuje pozicije svih okvira analize samoglasnika u f_1/f_2 ravnini zajedno sa njihovim reprezentantima koji se pohranjuju u matricu `foo`. Na slikama se prikazuju i Voronoi regije koje odgovaraju tim reprezentantima određenim funkcijom median na skupu podataka odabranog izgovora svakog govornika. To su u stvari iste slike koje je prikazivao i program 'dog6_formanti.m' prilikom formantne analize, tj. prilikom generiranja rezultata u direktoriju 'rez_files\'.

Nakon što program učitava rezultate za sve izgovore svih odabranih govornika, prikazuju se zbirni rezultati. Riječ je o prikazu podataka pohranjenih u matrici f_{oo} , tj. za svakog govornika se prikazuju isključivo reprezentanti, tj. medijani svakog glasa, svake formantne frekvencije i svakog izgovora, a ne više parametri svakog okvira analize, kao što je to bio slučaj u uvodnom dijelu programa. Naravno, obzirom da se prikaz provodi u f_1/f_2 ravnini, treći formant nije vidljiv iako podaci za njega postoje u matrici f_{oo} . Da bi bilo lakše prepoznati kojem glasu, kojem izgovoru, odnosno kojem govorniku pripada pojedina točka u prikazu, program prikazuje dvije zbirne slike, koje su za ovaj primjer prikazane na slikama 7.1-1 i 7.1-2. Na obje slike je prikazan isti skup podataka, dok je jedina razlika u načinu označavanja pojedinih elemenata slike. Na slici 7.1-1, sve točke koje odgovaraju istom govorniku su prikazane u istoj boji, dok se na slici 7.1-2 provodi grupiranje na osnovu samoglasnika, tj. svakom glasu odgovara jedna boja. Na prvu sliku je dodana i legenda u kojoj je navedeno kojem govorniku pripada pojedini skup točaka (boja). U oba slučaja sve točke koje pripadaju različitim izgovorima istog govornika su spojene linijama. Ako postoje samo dva izgovora onda je to pravac sa dvije točkice na kraju, a ako se radi o tri izgovora, tada je to trokut koji spaja sve tri točkice izgovora.

U slučaju kada se prikazuje velik broj govornika, tada je i pored prikaza u različitim bojama teško razaznati koje točke pripadaju pojedinom govorniku. Radi toga program omogućava dodatno označavanje jednog odabranog govornika u tim zbirnim prikazima, na način da se oko točkica koje odgovaraju tom govorniku doda mala crna kružnica. U primjerima na slikama 7.1-1 i 7.1-2, ta je mogućnost iskorištena za isticanje govornika s identifikacijskim brojem 20, što je ostvareno postavljanjem varijable `sel` na vrijednost koja odgovara identifikacijskom broju tog govornika, `sel=20`. U slučaju da je varijablom `sel` označen govornik koji se ne nalazi u listi odabranih govornika u varijabli 'odabrani', tada će navedeno dodatno označavanje biti izostavljeno. Inicijalizacija ove varijable se također provodi u zaglavlju programa 'dog6_prikaz.m'.



Slika 7.1-1 Prikaz reprezentanata formantne analize za tri izgovora pet različitih govornika (boje označavaju govornike)



Slika 7.1-2 Prikaz reprezentanata formantne analize za tri izgovora pet različitih govornika (boje označavaju glasove)

Kao što je vidljivo na slikama 7.1-1 i 7.1-2, pored samih točaka i njihovih spojnih linija, ucrtane su i Voronoi regije koje odgovaraju pojedinim glasovima. Reprezentanti tih regija su prikazani zvjezdicom u crnim kvadratićima, a mogu biti određeni na dva načina zavisno o varijabli `fix_class` koja je također definirana u zaglavlju programa 'dog6_prikaz.m'. Kao što je već bilo objašnjeno u prošloj vježbi, u slučaju da je `fix_class=1`, koristi se fiksni klasifikator čiji su parametri definirani u dvostupčanoj matrici `f12_fix` i koja je unaprijed određena usrednjavanjem rezultata formantne analize većeg broja muških govornika, tj. predstavlja svojevrsan univerzalni klasifikator. U drugom slučaju kada je `fix_class=0` (što odgovara slikama 7.1-1 i 7.1-2), reprezentanti klasifikatora se određuju običnim usrednjavanjem svih izgovora konkretnog odabira govornika definiranog varijablom 'odabrani', a sadržanog u matrici `foo`. Usrednjavanje se provodi sljedećim blokom naredbi:

```
% Odredi prosjecnu frekvenciju prvog formanta za
% svaki glas, usrednjavanjem preko svih govornika
% i izgovora
f1=mean(reshape(foo(:,1,:,:),bg,broj_gov*broj_izg))');

% Ista stvar za drugi formant
f2=mean(reshape(foo(:,2,:,:),bg,broj_gov*broj_izg))');
```

7.1.1 Rad s višedimenzionalnim matricama u Matlabu i funkcija 'reshape'

Iako Matlab podržava višedimenzionalne matrice, većina funkcija je definirana isključivo na skalarima, vektorima ili običnim dvodimenzionalnim matricama. Radi toga se u gornjem odsječku za izračunavanje klasifikatora samoglasnika koristi funkcija 'reshape', koja omogućava 'preslagivanje' matrice iz bilo kojeg početnog oblika u bilo koji novi oblik. Konkretno, u gornjim naredbama trodimenzionalnu matricu dobivenu fiksiranjem druge dimenzije matrice `foo` preslagujemo u običnu dvodimenzionalnu matricu te na stupce takove dvodimenzionalne matrice primjenjujemo funkciju 'mean'. Time smo u stvari zadnje dvije dimenzije objedinili u jednu dimenziju, čime smo sve izgovore svih govornika grupirali. Broj stupaca te nove dvodimenzionalne matrice odgovara broju glasova, dok broj redaka odgovara

produktu broja govornika i broja izgovora. Pri takvom preslagivanju jedini zahtjev je da ukupni broj elemenata bude jednak u obje matrice: polaznoj i završnoj. Obzirom da će ova funkcija biti intenzivno korištena i u ostatku programa potrebno je ukratko objasniti njen način rada.

Svi podaci u Matlab-u su interno u memoriji i tako pohranjeni kao jednodimenzionalna polja, tj. sve matrice neovisno o svojim dimenzijama se pohranjuju kao vektori, dok se informacije o dimenzijama matrice pohranjuju u posebnoj strukturi koja pripada toj varijabli. U svrhu razumijevanja načina rada funkcije 'reshape', značajno je poznavati redoslijed zapisa elemenata matrice u takvom jednodimenzionalnom polju. U Matlab-u je taj redoslijed određen 'ekspanzijom' indeksa višedimenzionalnih matrica s lijeva na desno, što će biti ilustrirano primjerom trodimenzionalne matrice sa dimenzijama I, J i K , koja je alocirana i inicijalizirana u nulu naredbom: $M = \text{zeros}(I, J, K)$. Dimenzija jednodimenzionalnog polja P za pohranu u memoriji neka je označena sa $N = I * J * K$ sa indeksom elemenata polja $n = 1, 2, \dots, N$. Veze između elemenata matrice M i elemenata tog polja P mogu biti definirane slijedećom pseudonaredbom:

$$M(i, j, k) = P((k-1) * J + (j-1) * I + i)$$

gdje indeksi po pojedinim dimenzijama matrice M , i, j, k , mijenjaju vrijednost od 1 do I , od 1 do J odnosno od 1 do K . U slučaju dvodimenzionalne matrice, zapis u memoriji odgovara raspisu matrice po stupcima, tj. kao da su stupci matrice posloženi jedan ispod drugog u memoriji.

U funkciji 'reshape' prvi argument je polazna matrica, a zatim slijede dimenzije izlazne matrice. Tako npr. naredba $M2 = \text{reshape}(M, I, J * K)$ će formirati novu dovdimenzionalnu matricu dimenzija I puta $(J * K)$, objedinjavanjem zadnje dvije dimenzije matrice M . Važno je napomenuti da je sadržaj i slijed elemenata u pripadnim jednodimenzionalnim poljima u memoriji identičan za obje matrice, tj. razlikuju se isključivo strukture sa informacijama o dimenzionalnosti. Zbog navedenog možemo zaključiti da funkcija 'reshape' ne radi ništa pametno, nego samo kopira sadržaj polja ulazne matrice u polje koje odgovara novoj matrici i formira novu strukturu koja opisuje dimenzije nove matrice.

7.2 Mjera udaljenosti između dva izgovora

U svrhu automatske klasifikacije govornika, ali i radi vrednovanja ponovljivosti izgovora istog govornika potrebno je definirati mjeru udaljenosti između dva izgovora zadanog skupa glasova (u našem primjeru šest samoglasnika). Mjera udaljenosti bi morala biti takva da udaljenost dva izgovora istog govornika bude manja nego udaljenost između dva izgovora različitih govornika. Svaki izgovor je opisan matricom centralnih frekvencija prvih sel_for formanta za bg samoglasnika. Kao jedan od mogućih logičnih odabira mjere udaljenosti, se nameće srednja kvadratna vrijednost relativnih promjena formantnih frekvencija. Neka su dva izgovora čiju udaljenost želimo odrediti su opisani matricama $F_A = [f_{a,i,j}]$ i $F_B = [f_{b,i,j}]$, gdje indeks retka i odgovara indeksu glasa, a indeks stupca j odgovara indeksu formanta. Kvadrat mjere udaljenosti između ta dva izgovora $d^2(A, B)$ će tada definiran kao:

$$d^2(A, B) = \frac{1}{\text{bg}} \cdot \frac{1}{\text{sel_for}} \sum_{i=1}^{\text{bg}} \sum_{j=1}^{\text{sel_for}} w_{i,j} \cdot \left(\frac{f_{a,i,j} - f_{b,i,j}}{f_{a,i,j} + f_{b,i,j}} \right)^2 \quad (7.2-1)$$

Apsolutne varijacije formantnih frekvencija su veće za više formante, čak i prilikom ponovljenog izgovora istog govornika. Radi toga se kao mjera udaljenosti koristi relativna

promjena formantnih frekvencija opisana gornjim kvocijentom. U svrhu poboljšanja rada klasifikatora, pogodno je u mjeru udaljenosti dodati i težine $w_{i,j}$, čija je uloga da svaki parametar izgovora tereti drugom težinom. Primjenom ovih težina moguće je istaknuti doprinos parametara kod kojih je ponovljivost višestrukih izgovora istog govornika vrlo dobra, a umanjiti značaj parametara čija je varijabilnost veća. Način određivanja ovih težina biti će opisan u narednom poglavlju.

7.3 Određivanje varijabilnosti izgovora istog govornika

Da bi bilo moguće projektirati automatski klasifikator govornika tj. odrediti težine $w_{i,j}$ koje figuriraju u izrazu (7.2-1) potrebno je poznavati varijabilnosti parametara formantne analize koje postoje kod višestrukih izgovora istog govornika. Za ocjenu te varijabilnosti, možemo se poslužiti prosječnom udaljenosti višestrukih izgovora istog govornika do srednje vrijednosti svih njegovih izgovora tog glasa, tj. svojevrsnom mjerom rasipanja izgovora istog govornika. Ako sa $foo(i,j,k,n)$ označimo median j -te formantne frekvencije k -tog govornika prilikom n -tog izgovora i -tog glasa, tada možemo usrednjavanjem preko svih izgovora tog govornika dobiti parametre prosječnog izgovora tog govornika za taj glas, označene sa $mfoo(i,j,k)$. Sada je moguće za svaki izgovor odrediti relativno odstupanje u odnosu na srednju vrijednost, kao:

$$pfoo(i,j,k,n) = (foo(i,j,k,n) - mfoo(i,j,k)) / mfoo(i,j,k) \quad (7.3-1)$$

Potrebno je još jednom naglasiti da su u matrici $pfoo$ pohranjene relativne razlike u odnosu na prosječni izgovor svakog pojedinog govornika. Odsječak programa koji služi za izračunavanje matrica $mfoo$ i $pfoo$ prikazan je u nastavku.

```
mfoo=zeros(bg,sel_for,broj_gov);
pfoo=0*foo;
for i=1:bg,
    for in_spe=1:broj_gov,
        f_i_in_spe=reshape(foo(i,:,in_spe,:),sel_for,broj_izg)';
        mfoo(i,:,in_spe)=mean(f_i_in_spe);
        ocek=ones(broj_izg,1)*mfoo(i,:,in_spe);
        pfoo(i,:,in_spe,:)=((f_i_in_spe-ocek)./ocek)';
    end;
end;
```

Što su varijacije pojedinog parametra između višestrukih izgovora istog govornika manje, to se taj parametar može sa većom težinom uzeti prilikom klasifikacije govornika, jer je 'ponovljiv'. S druge strane, parametrima za koje postoji veliko rasipanje i kod istog govornika nema smisla pridavati veliku važnost, pri klasifikaciji govornika. Radi toga se kao težina $w_{i,j}$ odabire recipročna vrijednost srednje kvadratne vrijednosti relativnih odstupanja svakog parametra, tj. svake formantne frekvencije svih bg samoglasnika. To usrednjavanje kvadratnih odstupanja se provodi preko svih izgovora svih govornika.

Međutim, zbog loše provedene analize ili loše snimljene zvučne sekvence ponekad postoji vrlo velika relativna promjena parametara između višestrukih izgovora istog govornika. Ovakve slučajeve treba tretirati kao iznimke, jer ako bi i njih uvrstili u ocjenu srednjeg kvadratnog odstupanja, tada bi ta vrijednost bila dominantno određena upravo tim iznimkama, te bi konačna težina pridijeljena tom parametru bila vrlo mala. Zbog toga se u programu prvo izračunava median apsolutne vrijednosti relativnog odstupanja me_f , te se na osnovu izračunatog mediana postavlja prag za razlikovanje normalnih uzoraka od iznimaka. Kvadrat relativne promjene se tada usrednjava samo za uzorke koji su ispod navedenog

praga, tj. samo za dobre uzorke, te se konačno pohranjuje u matricu `ms` kao što je ilustrirano u slijedećem odsječku programa:

```
pra_med=6;
ms=zeros(sel_for,bg);
for i_for=1:sel_for,
    prom_f=reshape(pfoo(:,i_for,:),bg,broj_gov*broj_izg);
    me_f=median(abs(prom_f'));
    ok_f=abs(prom_f)<pra_med*(me_f'*ones(1,broj_gov*broj_izg));
    ms(i_for,:)=sum((prom_f.^2).*ok_f') ./ sum(ok_f');
end;
```

Obzirom da se usrednjavanje tih kvadratnih varijacija provodi preko svih izgovora svih govornika, konačna dimenzija matrice `ms` iznosi `sel_for` redaka puta `bg` stupaca. Recipročna vrijednost elemenata te matrice odgovara težinama $w_{i,j}$. Detaljna objašnjenja svake naredbe, kao i dimenzije svake matrice možete naći u samom programu. Prag za razlikovanje dobrih izgovora od iznimaka određen je produktom varijable `pra_med` i retka sa medianima apsolutnih vrijednosti relativnih varijacija `me_f`. Vrijednost varijable `pra_med` treba odrediti eksperimentalno da izbací samo očite iznimke.

Zadatak 1

Za isti primjer sa početka ove vježbe, sa pet odabranih govornika i tri izgovora po govorniku, potrebno je nakon što program izračuna matricu `ms` prikazati težine $w_{i,j}$, normirane tako da najmanja težina iznosi 1. Diskutirati koji parametri imaju najveću ponovljivost, a koji najmanju (Napomena, ... težina je recipročna vrijednost elemenata matrice `ms`).

7.4 Formiranje matrice udaljenosti

U slučaju kada postoje po dva izgovora svakog govornika, tada se jedan skup izgovora može tretirati kao niz reprezentanata (po jedan za svakog govornika) koji definiraju klasifikator za automatsku selekciju govornika. Drugi skup izgovora tada predstavlja 'test' skup koji se koristi za provjeru rada klasifikatora. Za ispravan rad klasifikatora nužno je da kada se drugi izgovor i -tog govornika uspoređi sa prvim izgovorom svih govornika, tj. sa skupom reprezentanata, da se tada najmanja udaljenost ostvari baš za prvi izgovor i -tog govornika. To svojstvo mora biti zadovoljeno za svakog govornika i iz skupa govornika. Obratite pažnju da se prilikom klasifikacije nikad nije uspoređivao 2. izgovor i -tog govornika sa 2. izgovorom j -tog govornika, ili pak 1. izgovor i -tog govornika sa 1. izgovorom j -tog govornika. To znači da uz neki drugačiji odabir skupa reprezentanata i skupa za provjeru, tako odabrani klasifikator možda i ne bi radio. Odabir koji će izgovori formirati skup reprezentanata, a koji izgovori skup za provjeru može biti potpuno proizvoljan. Tako npr. uz gov govornika i izg izgovora po svakom govorniku, skup reprezentanata se može formirati na izg^{gov} različitih načina, jer se za svakog od gov govornika može na slučajan način odabrati jedan od izg izgovora. Preostalih $izg-1$ izgovora svakog govornika tada formira test skup za provjeru klasifikatora. Da bi svaki slučajni odabir reprezentanata formirao ispravan klasifikator, tj. takav koji će provesti ispravnu klasifikaciju svih preostalih izgovora i to svih govornika, nužno je da:

- **najveća udaljenost između bilo koja dva različita izgovora istog govornika bude manja od najmanje udaljenosti bilo kojeg para izgovora tog i bilo kojeg drugog govornika.**

Očito je da je za navedenu provjeru dovoljno izračunati udaljenost svakog izgovora svakog govornika do svakog izgovora tog istog i svakog drugog govornika. Dakle potrebno je formirati matricu udaljenosti \mathbf{M} čija je dimenzija $(gov*izg) \times (gov*izg)$. Takva matrica \mathbf{M} je u

stvari jedna super-matrica koja se sastoji od $(gov) \times (gov)$ pod-matrica \mathbf{M}_{ij} od kojih svaka opisuje udaljenosti između i -tog i j -tog govornika, kao što je prikazano na slici 7.4-1.

$$\begin{array}{c}
 \text{Indeks drugog govornika u paru} \\
 \begin{array}{cccc}
 j=1 & j=2 & j=3 & j=gov
 \end{array} \\
 \mathbf{M} = \begin{bmatrix}
 \mathbf{M}_{1,1} & \mathbf{M}_{1,2} & \mathbf{M}_{1,3} & \cdots & \mathbf{M}_{1,gov} \\
 \mathbf{M}_{2,1} & \mathbf{M}_{2,2} & \mathbf{M}_{2,3} & \cdots & \mathbf{M}_{2,gov} \\
 \mathbf{M}_{3,1} & \mathbf{M}_{3,2} & \mathbf{M}_{3,3} & \cdots & \mathbf{M}_{3,gov} \\
 \vdots & \vdots & \vdots & & \vdots \\
 \mathbf{M}_{gov,1} & \mathbf{M}_{gov,2} & \mathbf{M}_{gov,3} & \cdots & \mathbf{M}_{gov,gov}
 \end{bmatrix}
 \begin{array}{l}
 i=1 \\
 i=2 \\
 i=3 \\
 \vdots \\
 i=gov
 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{M} \\ \vdots \end{array}} \right\} \begin{array}{l} \text{Indeks prvog} \\ \text{govornika u paru} \end{array}$$

Slika 7.4-1 Super matrica \mathbf{M} koja definira udaljenost između svih parova govornika

Dimenzija svake takve pod-matrice $\mathbf{M}_{i,j}$ je $(izg) \times (izg)$ i za odabrani par govornika i - j definira udaljenosti svakog para izgovora ta dva govornika. Njena struktura je prikazana na slici 7.4-2.

$$\begin{array}{c}
 \text{Indeks izgovora j-tog govornika} \\
 \begin{array}{cccc}
 n=1 & n=2 & n=3 & n=izg
 \end{array} \\
 \mathbf{M}_{ij} = \begin{bmatrix}
 m_{i,j,1,1} & m_{i,j,1,2} & m_{i,j,1,3} & \cdots & m_{i,j,1,izg} \\
 m_{i,j,2,1} & m_{i,j,2,2} & m_{i,j,2,3} & \cdots & m_{i,j,2,izg} \\
 m_{i,j,3,1} & m_{i,j,3,2} & m_{i,j,3,3} & \cdots & m_{i,j,3,izg} \\
 \vdots & \vdots & \vdots & & \vdots \\
 m_{i,j,izg,1} & m_{i,j,izg,2} & m_{i,j,izg,3} & \cdots & m_{i,j,izg,izg}
 \end{bmatrix}
 \begin{array}{l}
 k=1 \\
 k=2 \\
 k=3 \\
 \vdots \\
 k=izg
 \end{array}
 \end{array}
 \left. \vphantom{\begin{array}{c} \mathbf{M}_{ij} \\ \vdots \end{array}} \right\} \begin{array}{l} \text{Indeks izgovora} \\ i\text{-tog govornika} \end{array}$$

Slika 7.4-2 Matrica \mathbf{M}_{ij} koja definira udaljenost svih parova izgovora za jedan odabrani par govornika i - j

Postoje određene pravilnosti u strukturi tih matrica. Uočite da matrice $\mathbf{M}_{i,j}$ i $\mathbf{M}_{j,i}$ definiraju udaljenosti između istog para govornika i i j , jedino što se u prvom slučaju govornik i tretira kao prvi govornik, a govornik j kao drugi, dok je u drugom slučaju upravo obrnuto. Radi toga radi se samo o zamjeni stupaca i redaka, dok su vrijednosti koje figuriraju u obje matrice jednake, tj. vrijedi: $\mathbf{M}_{i,j} = \mathbf{M}_{j,i}^T$. $\forall i,j \in [1,2,\dots,gov]$. To znači da su sve interesantne udaljenosti jednoznačno opisane s matricama na dijagonali od \mathbf{M} , kao i matricama samo ispod ili samo iznad nje. Obzirom da navedeno svojstvo transpozicije vrijedi i za slučaj kada je $i=j$, slijedi da $\mathbf{M}_{i,i} = \mathbf{M}_{i,i}^T$. $\forall i \in [1,2,\dots,gov]$, tj. sve pod-matrice na dijagonali supermatrice \mathbf{M} su simetrične. Ovakvo ponašanje je potpuno logično, jer te matrice opisuju udaljenosti između različitih izgovora istog govornika, a pri tome je udaljenost od k -tog do n -tog izgovora potpuno jednaka udaljenosti od n -tog do k -tog izgovora tog istog govornika. Potrebno je još naglasiti da su dijagonalni elementi dijagonalnih pod-matrica jednaki nuli, tj. $m_{i,i,k,k} = 0$ $\forall i \in [1,2,\dots,gov]$, $\forall k \in [1,2,\dots,izg]$, jer u stvari ti elementi opisuju udaljenosti istih izgovora

istih govornika koje svakako moraju biti jednake 0. Izračunavanje ove super-matrice **M** (naziv varijable u Matlabu je `udalj`) se provodi pomoću slijedećeg odsječka u Matlabu:

```
udalj=zeros(broj_izg,broj_izg,broj_gov,broj_gov);

for in_spe1=1:broj_gov,
    for in_spe2=1:broj_gov,
        for izg1=1:broj_izg,
            templ=foo(:, :, in_spe1, izg1);
            for izg2=1:broj_izg,
                targ=foo(:, :, in_spe2, izg2);
                udalj(izg1, izg2, in_spe1, in_spe2)=...
                    sqrt(mean( ( (templ-targ)./(templ+targ)).^2 )
                        ./ms')) * fak_for' / sum(fak_for);
            end;
        end;
    end;
end;
```

Prve dvije dimenzije matrice `udalj` definiraju sve kombinacije izgovora dva odabrana govornika, dok druge dvije dimenzije određuju sve kombinacije govornika. U gornjem odsječku, funkcija `mean` provodi usrednjavanje terećenih kvadratnih odstupanja preko svih glasova, dok se usrednjavanje preko svih formanata provodi korištenjem još jedne empirijske težine definirane u retku `fak_for`, kojim je moguće dodatno forsirati značaj svakog formanta.

U svrhu provjere rada klasifikatora, nakon izračunavanja ove matrice, program izračunava još nekoliko pomoćnih matrica, te ih prikazuje u tabličnom obliku.. To su slijedeće matrice:

`u` matrica prosječne udaljenosti bilo kojeg para govornika
`u_max` ... matrica najveće udaljenosti bilo kojeg para govornika
`u_min` matrica najmanje udaljenosti bilo kojeg para govornika

Sve tri matrice su dimenzija `broj_gov` puta `broj_gov` jer definiraju udaljenost bilo kojeg para govornika (istog ili različitih). Prilikom izračunavanja matrice `u`, provodi se usrednjavanje samo elemenata različitih od nule, tj. u slučaju izračunavanja prosječne udaljenosti između istih govornika, usrednjavanje se provodi samo za kombinacije sa različitim izgovorom. Prilikom formiranja matrica `u_max` i `u_min` pored same udaljenosti se pohranjuje i indeksi para izgovora uz koji se postiže dotični ekstrem u matricama `ind_min` i `ind_max`. Formalno se radi o dva indeksa (indeks retka izgovora prvog govornika `ind_red` i indeks stupca izgovora drugog govornika `ind_stup`), ali su radi jednostavnosti ti indeksi pohranjeni kao jedan kompozitni indeks koji u stvari odgovara produktom kodu, tj.

$$\text{indeks} = (\text{ind_stup} - 1) * \text{broj_izg} + \text{ind_red} \quad (7.4-1)$$

U nastavku je prikazan odjesečak programa za izračunavanje gore navadenih matrica.

```
u=zeros(broj_gov,broj_gov);
u_min=u; u_max=u;
ind_min=u; ind_max=u;

for in_spe1=1:broj_gov,
    for in_spe2=1:broj_gov,
        if ( in_spe1-in_spe2==0 ),
            skala=(broj_izg^2)-broj_izg;
        else,
            skala=broj_izg^2;
```



```

end;
u_1_2=udalj(:, :, in_spe1, in_spe2);
u(in_spe1, in_spe2)=sum(sum(u_1_2))/skala;
[u_min(in_spe1, in_spe2), ind_min(in_spe1, in_spe2)]=min(u_1_2(:));
[u_max(in_spe1, in_spe2), ind_max(in_spe1, in_spe2)]=max(u_1_2(:));
end;
end;

```

Konačni rezultati programa su prikazani u obliku sljedećih tabela:

Prosječne udaljenosti govornika -

.... usrednjeno preko svih kombinacija izgovora

govornik	govornik				
	14	15	16	17	20
14	0.7				
15	3.1	0.7			
16	3.1	3.9	0.8		
17	2.3	3.7	3.1	0.8	
20	3.6	2.7	4.2	4.0	1.0

Ovo je u stvari ispis matrice u koja je simetrična, pa je prikazan samo njen donji trokutasti dio. U nastavku slijede najbliže kombinacije govornika, određene traženjem minimalne udaljenosti u svakom retku i svakom stupcu.

Najblize veze govornika na osnovu usrednjenih izgovora

```

14 -> 14
15 -> 15
16 -> 16
17 -> 17
20 -> 20

```

Ovo je prikaz svih elemenata različitih od nule svih pod-matrica $M_{i,i}$ na glavoj dijagonali super-matrice M , tj. ova tablica definira ponovljivost različitih izgovora istog govornika. U zadnjem stupcu je prikazana i maksimalna vrijednost, koja se u stvari nalazi u dijagonalnim elementima matrice u_{max} .

Udaljenosti razlicitih izgovora istog govornika:

Govornik	1->2	1->3	2->3	max
Andrej Dolmac	0.83	0.68	0.71	0.83
Hrvoje Bogunovic	0.69	0.75	0.78	0.78
Marko Dvecko	0.63	0.90	1.02	1.02
Srdjan Pajic	0.81	0.72	0.91	0.91
Mitar Maric	1.09	1.06	0.84	1.09

Zadnje dvije tablice definiraju najbliže i najdalje izgovore različitih govornika. Ova prva je posebno kritična, jer sve vrijednosti u zadnjem stupcu tablice minimalnih udaljenosti moraju biti strogo veće od maksimalnih udaljenosti različitih izgovora istih govornika, prikazanih u zadnjem stupcu prethodne tabele. Ako je to zadovoljeno, tada će klasifikator raditi ispravno za bilo koji odabir skupa reprezentanata i skupa za provjeru.

Kombinacije s najblizim izgovorom razlicitih govornika:

Govornik	/ Izgovor	->	Najblizi par	/ Izgovor	Min udalj
----------	-----------	----	--------------	-----------	-----------

Andrej Dolmac	/ 3	->	Srdjan Pajic	/ 1	2.20
Hrvoje Bogunovic	/ 3	->	Mitar Maric	/ 3	2.61
Marko Dvecko	/ 1	->	Srdjan Pajic	/ 2	2.85
Srdjan Pajic	/ 1	->	Andrej Dolmac	/ 3	2.20
Mitar Maric	/ 3	->	Hrvoje Bogunovic	/ 3	2.61

Konačno i tablica s najvećim udaljenostima iz koje je moguće za svakog govornika odrediti njegov najdalji par.

Kombinacije s najdaljim izgovorom razlicitih govornika:

Govornik	/ Izgovor	->	Najdalji par	/ Izgovor	Max udalj
Andrej Dolmac	/ 2	->	Mitar Maric	/ 2	3.95
Hrvoje Bogunovic	/ 2	->	Marko Dvecko	/ 3	4.21
Marko Dvecko	/ 3	->	Mitar Maric	/ 2	4.44
Srdjan Pajic	/ 1	->	Mitar Maric	/ 2	4.42
Mitar Maric	/ 2	->	Marko Dvecko	/ 3	4.44

Prva matrica koja se izračunava je matrica prosječne udaljenosti svakog para govornika. Na osnovu ove matrice moguće je odrediti "sličnosti" govornika.

Zadatak 2.

Pored navedenih govornika, u skup govornika uključiti i rezultate formantne analize za vaše izgovore samoglasnika (govornik s identifikacijskim brojem 0). Napomena, ... koristiti rezultate iz prethodne vježbe. Provjeriti da li klasifikator radi pravilno i nakon uključjenja vaših izgovora, te odrediti tko vam je najbliži i najdalji par.

Zadatak 3.

Odabrati slijedeći skup govornika:

```
odabrani=[1 2 3 4 5 6 7 8 12 14 15 16 17 18 20];
```

i fiksirati broj izgovora na 2, te ponoviti izvršenje programa za tako odabrane govornike. Analizirati u koliko slučajeva klasifikator ne radi pravilo i do kojih zamjena dolazi.

Zadatak 4.

Za isti gore navedeni skup govornika, prikazati matricu težina i usporediti je s rezultatima prvog zadatka koji je bio određen na osnovu samo pet govornika. Koji su glasovi najviše a koji najmanje ponovljivi?

8. POSTUPCI LINEARNE PREDIKCIJE

Student: Biškupić Tihomir

POČETAK

KRAJ

9. AUTOKORELACIJSKI POSTUPAK IZRAČUNAVANJA PREDIKTORA

Student: Carević Srđan
POČETAK

9.1 Uvod

Linearna predikcija je tehnički postupak koji se temelji na pretpostavci da se trenutni uzorak bilo kojega signala može predvidjeti pomoću linearne kombinacije konačnog broja prošlih uzoraka. Parametri linearnog prediktora su težinski koeficijenti sa kojima se prije sumiranja množe uzorci signala. Množenje sa težinskim koeficijentima je potrebno zato što iznosi svih prethodnih uzoraka ne utječu jednako na iznos trenutnog uzorka. Broj težinskih koeficijenata ujedno određuje i red prediktora. Prijenosna funkcija linearnog prediktora reda p dana je izrazom (9.1-1).

$$P(z) = \sum_{k=1}^p a_k \cdot z^{-k} \quad (9.1-1)$$

Cilj LPC analize (*eng. Linear Prediction Coding*) je određivanje parametara linearnog prediktora tako da razlika između originalnih uzoraka i uzoraka dobivenih predikcijom bude minimalna. Međutim, budući da se vokalni trakt ponaša kao linearni sustav sa vremenski promjenjivim parametrima, linearni prediktor dobiven LPC analizom će imati minimalnu pogrešku samo na određenom vremenskom intervalu. Zbog toga je potrebno postupak LPC analize provoditi kontinuirano na okvirima signala (*eng. frame*) u kojima se vokalni trakt može promatrati kao linearni vremenski nepromjenjiv sustav. Razmak između dvije analize iznosi tipično 10 do 20 milisekundi. Parametri linearnog prediktora se mogu odrediti pomoću neke od ovih metoda:

- metoda autokorelacije
- metoda kovarijance
- ljestvičaste strukture

U ovoj vježbi, za određivanje parametara linearnog prediktora, koristiti će se metoda autokorelacije. Za tu metodu karakteristično je da se dio signala najprije izdvoji pomoću vremenikoga otvora, na izdvojenom okviru se zatim izračuna vrijednost autokorelacijske funkcije, a nakon toga se iz dobivenih rezultata sa nekim od postupaka rješavanja linearnih sustava jednadžbi izračunaju vrijednosti koeficijenata prediktora.

9.2 Primjer vježbe

Za izvođenje ove vježbe potrebno je pokrenuti *Matlab 5.0*. Nakon što se pojavi komandna linija potrebno se je pozicionirati u radni direktorij za ovu vježbu pomoću naredbe

```
>>cd c:\users\dog\dog3_dir
```

```
Sada se naredbom what može provjeriti da li se u
```

```
direktoriju nalaze datoteke potrebne za izvođenje vježbe. Rezultat ispisa bi trebao biti kako
```

```
slijedi:
```

```
>>what
```

```
M-files in the current directory C:\users\dog\dog3_dir
```

```
dog3      wavread2
```

MAT-files in the current directory C:\users\dog\dog3_dir

samo

Osim ovih datoteka u istome direktoriju trebala bi se nalaziti i datoteka *samo.wav* što se najlakše može provjeriti zadavanjem naredbe *dir*.

9.2.1 Datoteka *samo.wav*

Datoteka *samo.wav* je zvučna datoteka koja u sebi sadrži snimljeni uzorak govornoga signala. Za kreiranje navedene datoteke korišten je *Sound Recorder*, a snimanje je izvedeno uz sljedeće parametre:

- frekvencija uzorkovanja: 8000 Hz
- broj kanala: mono
- kvantizacija: 16 bita

U datoteci se nalaze samoglasnici snimljeni sljedećim redom: *a, e, i, o, u, 3*. Ovdje je sa *3* označen neutralni samoglasnik tzv. *šva* (*schva*) koji se u kraćoj varijanti označava kao zrcaljeno slovo *e* (britanski izgovor riječi *ago*), a u dužoj kao *3*: (britanski izgovor riječi *her*). Ukoliko samostalno snimate ovu datoteku tada treba paziti da ne dođe do prekoračenja dinamike (*eng. clipping*) te da prilikom izgovora glas bude što ravnomjerniji. Između samoglasnika je potrebno ostaviti pauzu od oko 0.5 sekundi.

9.2.2 Datoteka *dog3.m*

Ako postoje sve navedene datoteke, za početak vježbe dovoljno je pokrenuti datoteku *dog3.m*. Kako program *dog3.m* radi može se vidjeti ako ga se otvori u *Notepad*-u. Na početku datoteke navedena su sljedeća četiri parametra:

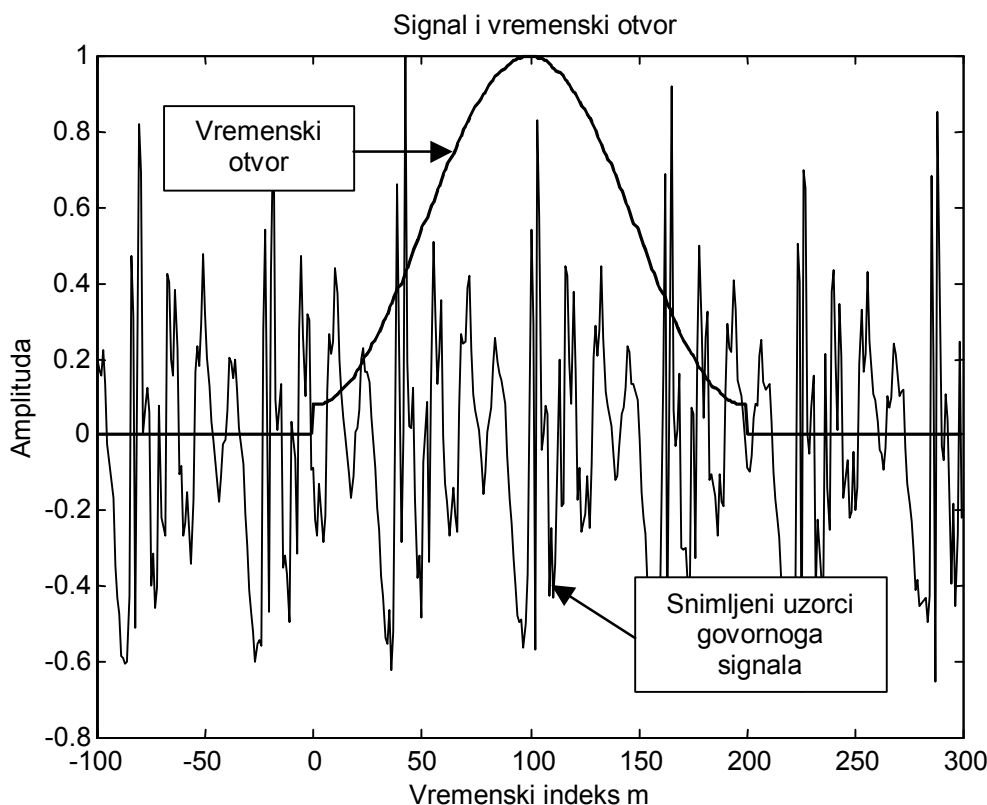
- *fs* frekvencija uzorkovanja govornoga signala izražena u Hz
- *N* širina vremenskoga otvora izražena brojem uzoraka
- *fr_step* pomak vremenskoga otvora izražen brojem uzoraka
- *P* red linearnog prediktora

Ako se program pogleda par redova ispod ovih parametara, može se uočiti da program *dog3.m* u sebi poziva datoteku *wavread2.m* koja obavlja učitavanje snimljenih uzoraka govornoga signala i njihovo spremanje u varijable *Matlab*-a.

Pokrenemo li *dog3.m* zadavanjem naredbe

```
>>dog3
```

na ekranu će se najprije pojaviti prikaz valnog oblika svih snimljenih samoglasnika i kurzor pomoću kojega se vrši odabir željenoga dijela govornoga signala. Program čeka sve dok korisnik ne klikne unutar jednoga “paketa” koji predstavlja samoglasnik, a zatim na sljedećoj slici prikaže odabrani dio signala i vremenski otvor kao što je to prikazano na slici 9.2-1. U prikazanom slučaju odabran je samoglasnik *e*. Budući da je pomoću kurzora teško odabrati dva puta isti blok, a radi boljeg razumijevanja rada programa pogodnije je da se dobivaju iste slike i numerički rezultati kao i u ovome primjeru vježbe, u programu *dog3.m* potrebno je u varijablu **odabir_bloka** upisati vrijednost nula, a u varijablu **blok** vrijednost 124. Naravno, ovo vrijedi samo ako koristite postojeću datoteku *samo.wav*.



Slika 9.2-1 Odabrani okvir samoglasnika *e* i vremenski otvor

Prije prikaza izvršen je sljedeći blok naredbi:

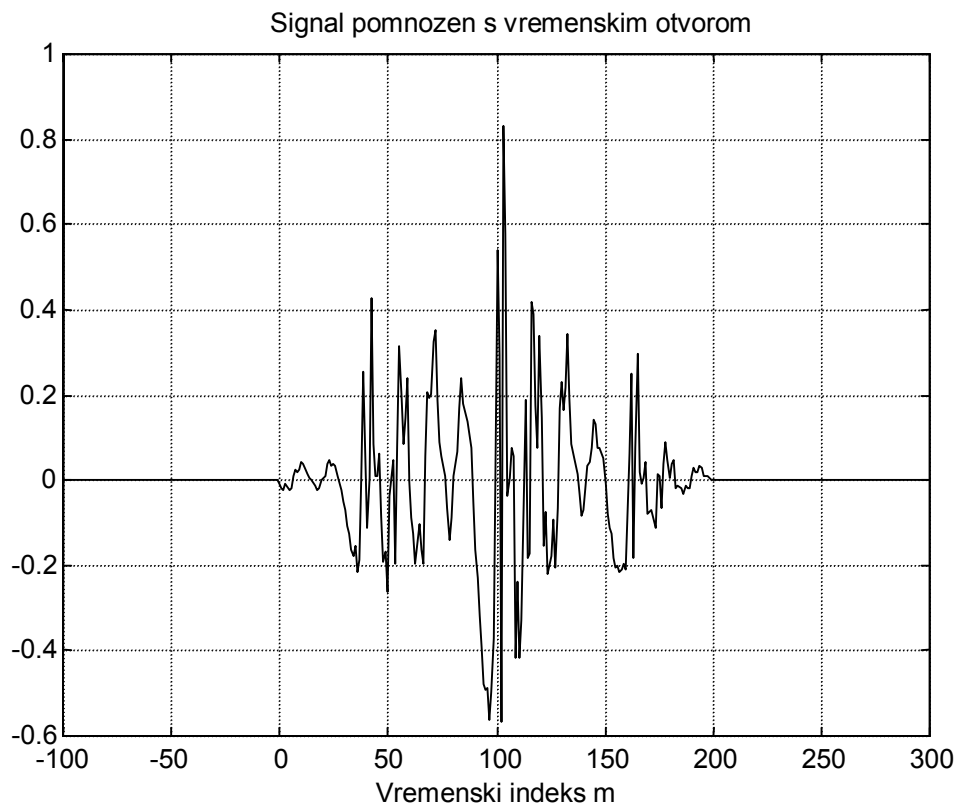
```
ee=ee/max(abs(ee));
fN2=floor(N/2);
cN2=N-fN2;
x=[-fN2:N+cN2-1];
w=hamming(N);
w=[zeros(1,fN2) (w(:))' zeros(1,cN2)];
```

Prvom naredbom izvršeno je normiranje odabranog dijela govornog signala, koji se nalazi u varijabli *ee*, na interval ± 1 . Sa sljedeće tri naredbe stvoren je vektor *x* koji će definirati vrijednosti na vremenskoj osi tako da se može vidjeti izgled signala i vremenskog otvora prije i poslije odabranog okvira dužine *N*. Na kraju je kreiran *Hammingov* vremenski otvor *w* dužine *N* te je zatim njegova dužina povećana dodavanjem nula tako da bude istih dimenzija kao i vektor *x* kako bi ih mogli prikazati na istome grafičkome prikazu.

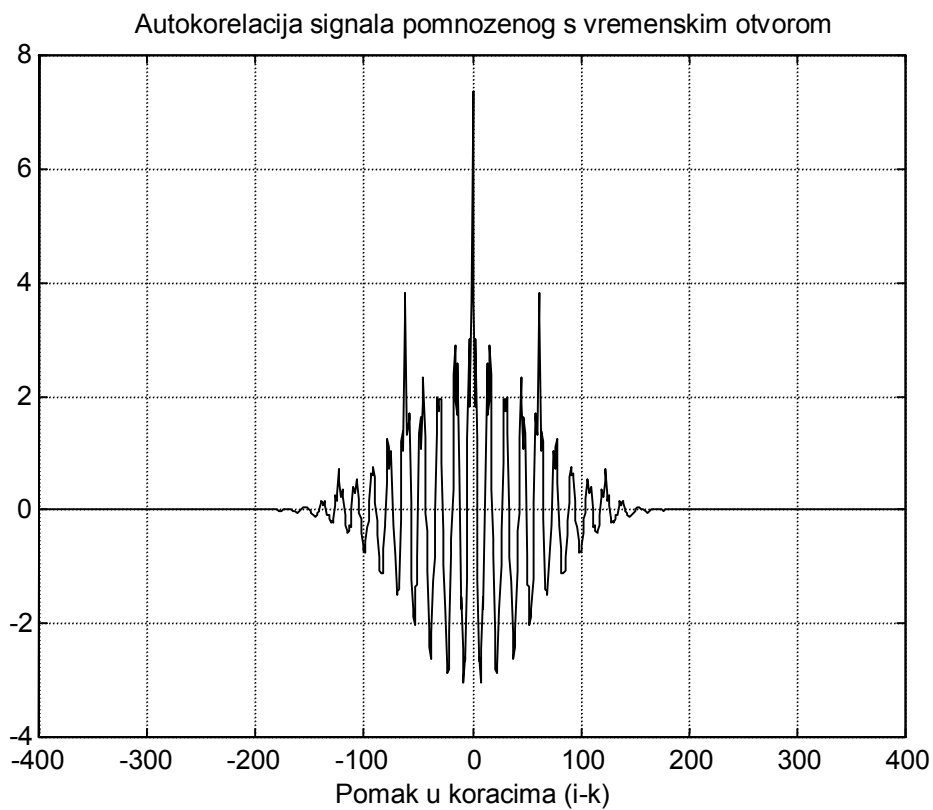
Na sljedećem grafu, prikazanom na slici 9.2-2, prikazan je rezultat množenja signala i vremenskoga otvora što je izvedeno korištenjem naredbe:

```
ee=ee.*w;
```

Amplitude dijela signala obuhvaćenog vremenskim otvorom, nakon množenja su promijenile iznos to više što je uzorak signala udaljeniji od središta vremenskog otvora. Uzrok tome je oblik *Hammingovog* vremenskog otvora koji prema rubu sve više smanjuje amplitude signala. Korištenjem npr. *pravokutnog* vremenskog otvora ne bi došlo do spomenutog izobličenja u vremenskoj domeni, međutim dobila bi se loša frekvencijska rezolucija. Hammingov vremenski otvor se u digitalnoj obradi govora koristi zato što mu glavna latica nije preširoka, a bočne latice su mu dovoljno prigušene.



Slika 9.2-2 Uzorci samoglasnika *e* pomnoženi sa vremenskim otvorom



Slika 9.2-3 Autokorelacija govornoga signala pomnoženog sa vremenskim otvorom

Zadavanjem naredbi

```
rr=xcorr(ee);
rr=(rr(:))';
xrr=[-2*N+1:2*N-1];
plot(xrr,rr);
```

izračunava se i zatim prikazuje autokorelacija signala pomnoženoga sa vremenskim otvorom. Dobiveni prikaz trebao bi nalikovati na onaj prikazan na slici 9.2-3. Na slici se može uočiti svojstvo simetričnosti autokorelacijske funkcije tj. vrijednost funkcije je ista za istu apsolutnu vrijednost pomaka bez obzira da li je pomak pozitivan ili negativan. Program sada u komandnom prozoru *Matlab*-a ispisuje vrijednosti autokorelacijske funkcije za pomake od -5 do +5 iz kojih se može vidjeti da se poklapaju u svim decimalama. Pomoću naredbe

```
disp(max(abs(rr(1:2*N-1)-rr(4*N-1:-1:2*N+1))))
```

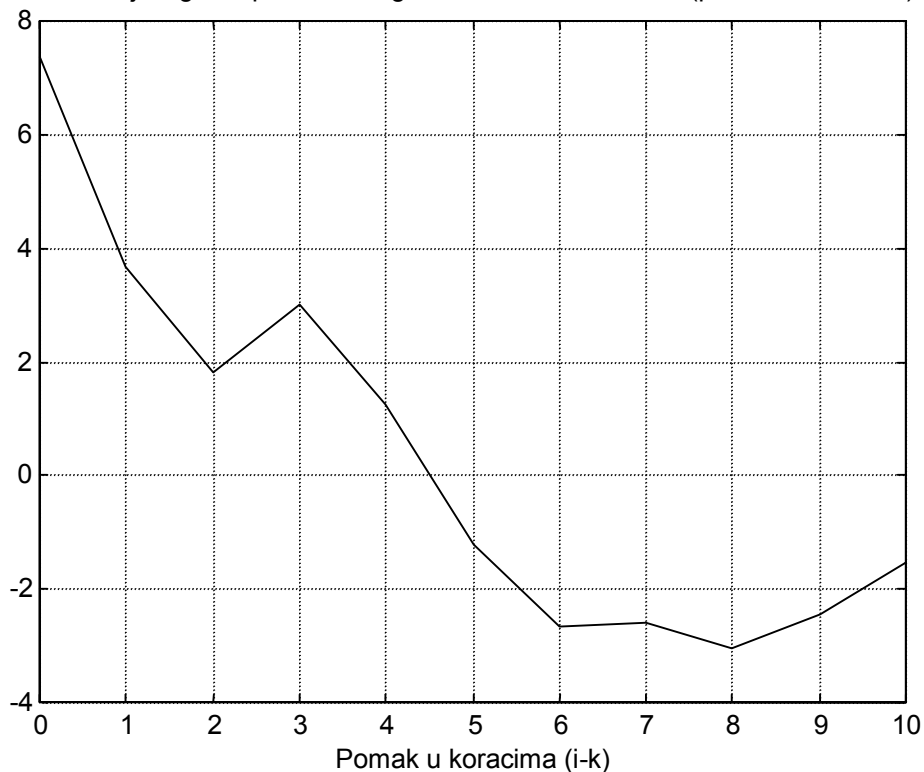
moguće je provjeriti maksimalni apsolutni iznos odstupanja na cijelom intervalu za koji je računata vrijednost autokorelacijske funkcije. U promatranom slučaju dobiven je iznos nula.

Za daljnje proračune potrebne su vrijednosti autokorelacije za pomake od 0 do P. Njihovo izdvajanje izvršeno je naredbom

```
R=rr(2*N+0:2*N+P);
```

, a izgled izdvojenog dijela prikazan je na slici 9.2-4.

Autokorelacija signala pomnoženog s vremenskim otvorom (pomaci od 0 do P)



Slika 9.2-4 Prikaz dijela autokorelacije signala pomnoženoga sa vremenskim otvornom za pomake od 0 do P

U nastavku programa najprije se naredbama

```
% Matrica sa indeksom retka i
i=(ones(P,1)*[1:P])';
% Matrica sa indeksom stupca k
k=ones(P,1)*[1:P];
imk=i-k;
imka=abs(imk);
```


formira matrica indeksa (i-k) koji su argument autokorelacijske funkcije u matrici \mathbf{R}_m koja će biti formirana kasnije. U posljednjoj naredbi iskorišteno je svojstvo simetričnosti autokorelacijske funkcije oko nule tako da se u matrici \mathbf{R}_m nalaze samo apsolutne vrijednosti pomaka. Sljedeći korak je formiranje matrice \mathbf{R}_m i slobodnog stupca \mathbf{R}_v pomoću naredbi

```
Rm=zeros(P,P);
Rm(:)=R(imka+1);
Rv=R(2:P+1)';
```

Poznavajući ove matrice, moguće je prema izrazu (9.2-1) napisati linearni sustav jednačbi

$$\mathbf{R}_m * \alpha = \mathbf{R}_v \quad (9.2-1)$$

, gdje je sa α označen vektor stupac koji sadrži parametre linearnog prediktora koje želimo odrediti. Matrica \mathbf{R}_m je tzv. Toeplitz matrica (čita se *Teplić*) i za nju je karakteristično da je simetrična oko glavne dijagonale te da po ostalim dijagonalama ima iste vrijednosti. Opći izgled matrice \mathbf{R}_m za prediktor p-tog reda prikazan je na slici 9.2-5.

$$\begin{bmatrix} R(0) & R(1) & R(2) & R(3) & \dots & R(p-4) & R(p-3) & R(p-2) & R(p-1) \\ R(1) & R(0) & R(1) & R(2) & \dots & R(p-5) & R(p-4) & R(p-3) & R(p-2) \\ R(2) & R(1) & R(0) & R(1) & \dots & R(p-6) & R(p-5) & R(p-4) & R(p-3) \\ R(3) & R(2) & R(1) & R(0) & \dots & R(p-7) & R(p-6) & R(p-5) & R(p-4) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ R(p-4) & R(p-5) & R(p-6) & R(p-7) & \dots & R(0) & R(1) & R(2) & R(3) \\ R(p-3) & R(p-4) & R(p-5) & R(p-6) & \dots & R(1) & R(0) & R(1) & R(2) \\ R(p-2) & R(p-3) & R(p-4) & R(p-5) & \dots & R(2) & R(1) & R(0) & R(1) \\ R(p-1) & R(p-2) & R(p-3) & R(p-4) & \dots & R(3) & R(2) & R(1) & R(0) \end{bmatrix}$$

Slika 9.2-5 Opći izgled marice \mathbf{R}_m za prediktor p-tog reda

Zbog Toeplitz strukture, ovaj linearni sustav je moguće riješiti pomoću više iterativnih postupaka. Jedan od poznatijih je Levinson-Durbinov rekurzivni postupak. U ovome slučaju sustav je riješen tako da je najprije izračunata inverzna matrica matrice \mathbf{R}_m te je zatim cijela jednačba pomnožena s lijeva sa tom inverznom matricom. Cijelu navedenu operaciju obavila je naredba:

```
alf=inv(Rm)*Rv;
```

U vektor stupcu **alf** nalaze se parametri linearnog prediktora. Korištenjem matrice **alf** formirana je matrica **a** koja sadrži koeficijente inverznog filtra. Prijenosna funkcija inverznog filtra u z-domeni dana je izrazom (9.2-2).

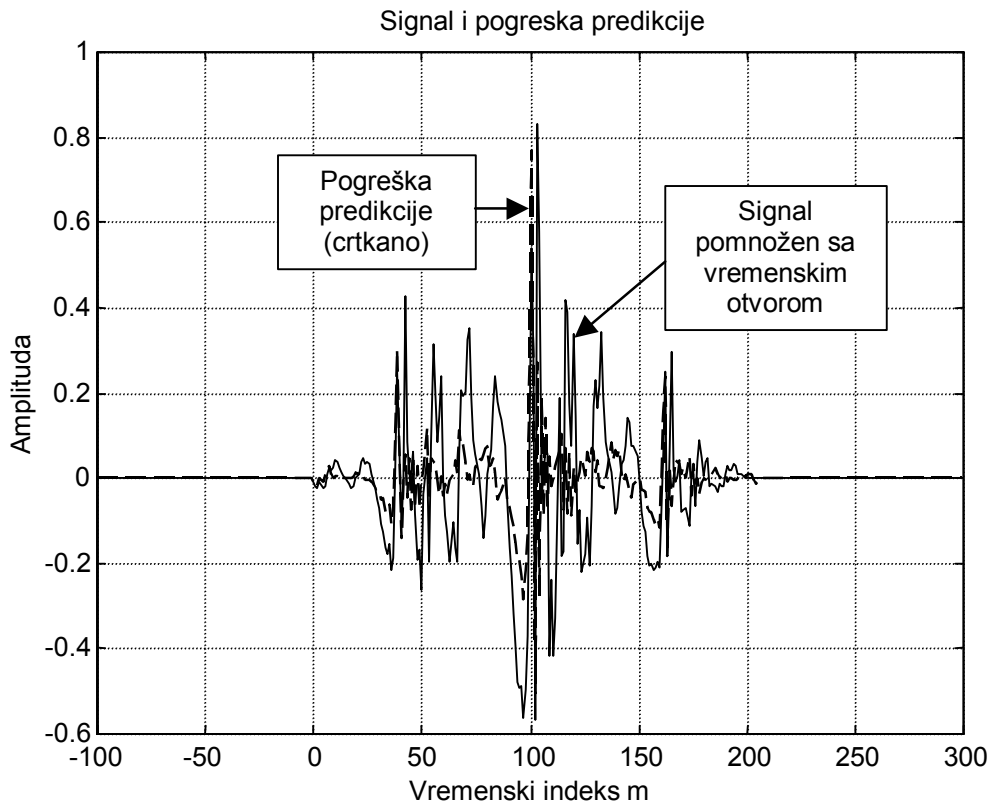
$$A(z) = 1 - \sum_{k=1}^p a_k * z^{-k} \quad (9.2-2)$$

Inverzni filter opisuje prijenosnu funkciju za pogrešku predikcije. Poznavajući koeficijente inverznog filtra, pomoću naredbi:

```
a=[1 -alf'];
```

```
rezid=conv(a,ee);
rezid=rezid(1:2*N);
plot(x,[ee' rezid]);
```

možemo odrediti i nacrtati rezidualni signal tj. signal razlike između stvarnoga signala i izlaznog signala prediktora. Dobiveni rezultati prikazani su na slici 9.2-6 iz koje je vidljivo da se velika greška pojavi svaki put kada naiđe veliki puls.



Slika 9.2-6 Signal pomnožen sa vremenskim otvorom i pogreška predikcije

Sljedeći korak je izračunavanje sumarne kvadratne pogreške prediktor. To je moguće izvesti na dva načina. Općenito, sumarna kvadratna pogreška prediktora definirana je kao suma kvadrata razlike između trenutnih vrijednosti originalnih uzoraka i trenutnih vrijednosti uzoraka dobivenih prediktorom kao što je to opisano izrazom (9.2-3).

$$E_n = \sum_m (s_n(m) - \sum_{k=1}^p s_n(m-k))^2 \quad (9.2-3)$$

Indeks n odnosi se na promatrani blok signala. Budući da je u prethodnom koraku već određena pogreška predikcije za svaki uzorak, zadavanjem naredbe:

```
ssrezid=sum(rezid.^2);
```

u promatranome slučaju dobivena je sumarna kvadratna pogreška predikcije iznosa 1.991167. Ukoliko su poznate vrijednosti težinskih koeficijenata prediktora, što je slučaj u ovome primjeru, sumarna kvadratna pogreška predikcije se može odrediti i prema izrazu (9.2-4).

$$E_n = R(0) - \sum_{k=1}^p a_k * R(k) \quad (9.2-4)$$

Izraz (9.2-4) zapisan kao naredba *Matlab*-a ima sljedeći oblik:

```
E=R(1)-R(2:P+1)*alf;
```

Budući da indeksi elemenata matrica u *Matlab*-u mogu biti samo cijeli brojevi veći od nule, element $R(1)$ zapravo je iznos autokorelacijske funkcije za pomak nula $R(0)$. Izvršavanjem ove naredbe dobiven je isti iznos sumarne kvadratne pogreške predikcije kao i u prvome slučaju.

Koeficijenti prediktora, prema izrazu (9.2-5), definiraju prijenosnu funkciju digitalnog filtra $H(z)$ koji ima samo polove (*eng. all-pole IIR filter*).

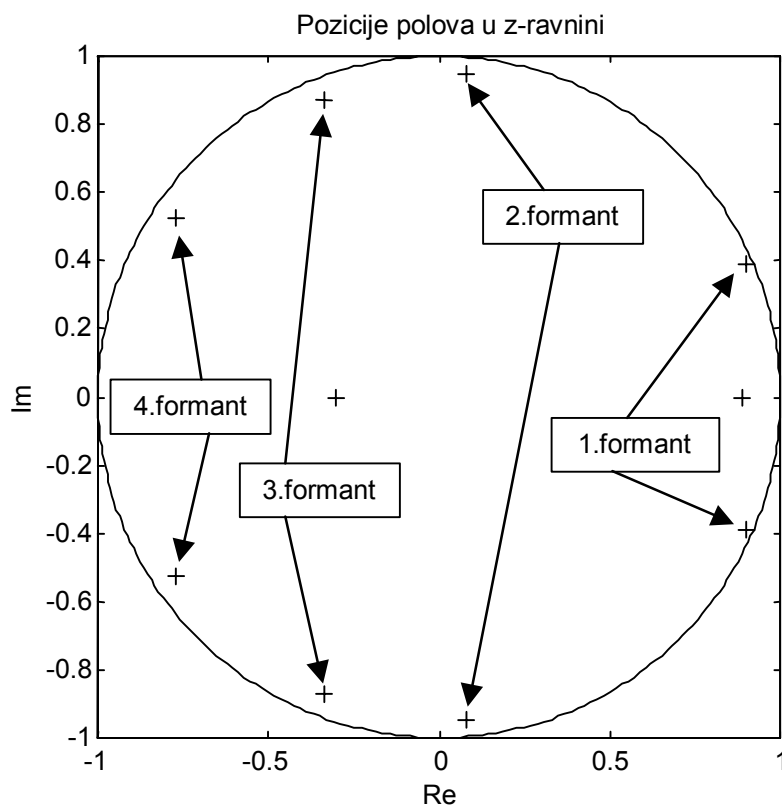
$$H(z) = \frac{\sqrt{E_n}}{1 - \sum_{k=1}^p a_k * z^{-k}} \quad (9.2-5)$$

Kao što se može vidjeti iz izraza (9.2-5), u nazivniku se nalazi prijenosna funkcija inverznoga filtra. Naredbom:

```
ko=roots(a);
```

dobivamo nule inverznog filtra odnosno polove digitalnog filtra $H(z)$. Iscrtavanje položaja polova u z -ravnini ostvaruje se sljedećim nizom naredbi:

```
kut=[0:300]/300*2*pi;
plot(sin(kut),cos(kut));axis('square');
hold on;
plot(real(ko),imag(ko),'+')
, a dobiveni rezultati prikazani su na slici 9.2-7.
```



Slika 9.2-7 Položaj polova prijenosne funkcije $H(z)$ u z -ravnini

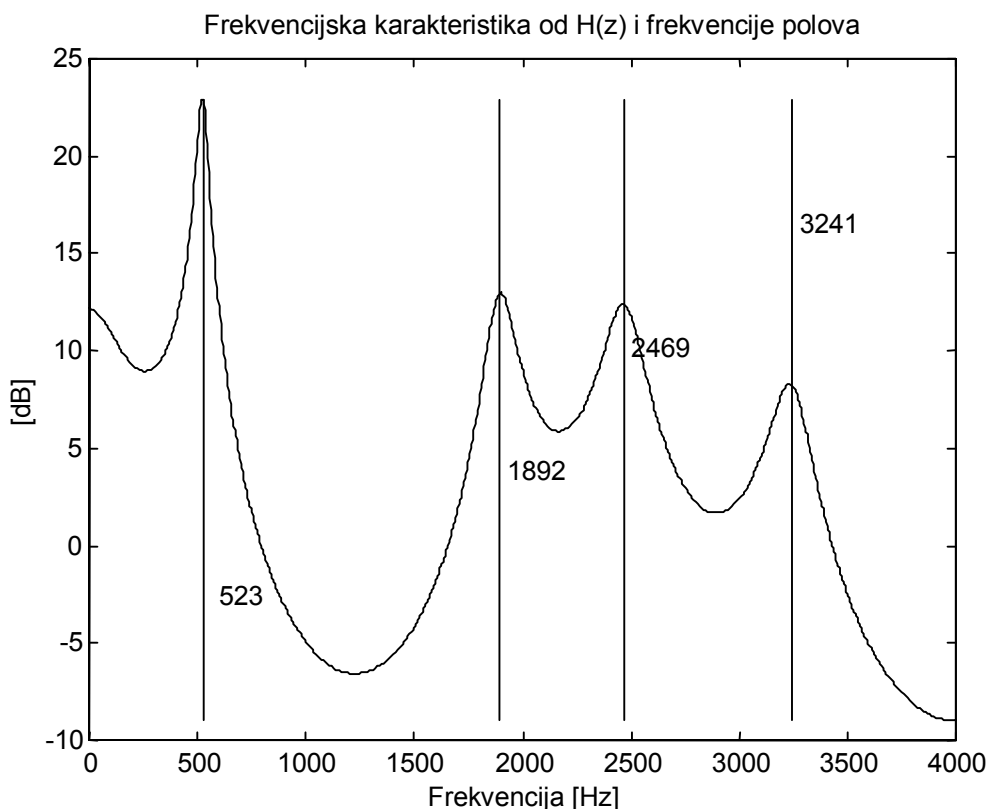
Sa slike je vidljivo da se svi polovi nalaze unutar jedinične kružnice što znači da je sustav stabilan. Budući da su koeficijenti prijenosne funkcije $H(z)$ realni brojevi, polovi kojima je imaginarni dio različit od nule javljaju se u konjugirano kompleksnim parovima. Što je pol

bliži kružnici formant koji je određen njime imati će manju širinu i viši vrh. Frekvencija formanta, izražena u radijanima, određena je kutem između pozitivnog dijela realne osi i vektora koji spaja ishodište koordinatnog sustava i pol. Kut π odgovara $fs/2$.

Sljedeći korak je iscrtavanje frekvencijske karakteristike od $H(z)$ u logaritamskom mjerilu. Taj zadatak je obavljen pomoću sljedećeg bloka naredbi:

```
fre=angle(ko);
fre=sort(fre(find(fre>0.01 & fre<pi-0.01)))/pi*fs/2;
[h,om]=freqz(sqrt(E),a,1024);
Budući da se u varijabli h nalazi kompleksna amplituda, prije logaritmiranja potrebno je
odrediti njezin modul pomoću funkcije abs.
lo=20*log10(abs(h));
plot(om/pi*fs/2,lo)
hold on;
malo=max(lo);
milo=min(lo);
for ii=1:max(size(fre)),
    plot([fre(ii) fre(ii)],[milo malo],'r-');
    sfre=sprintf('%5.0f',fre(ii));
    text(fre(ii)+0,milo+(malo-milo)*ii/(max(size(fre))+1), sfre);
end;
```

Dobiveni rezultat prikazuje slika 9.2-8 na kojoj su naznačeni iznosi rezonantnih frekvencija.

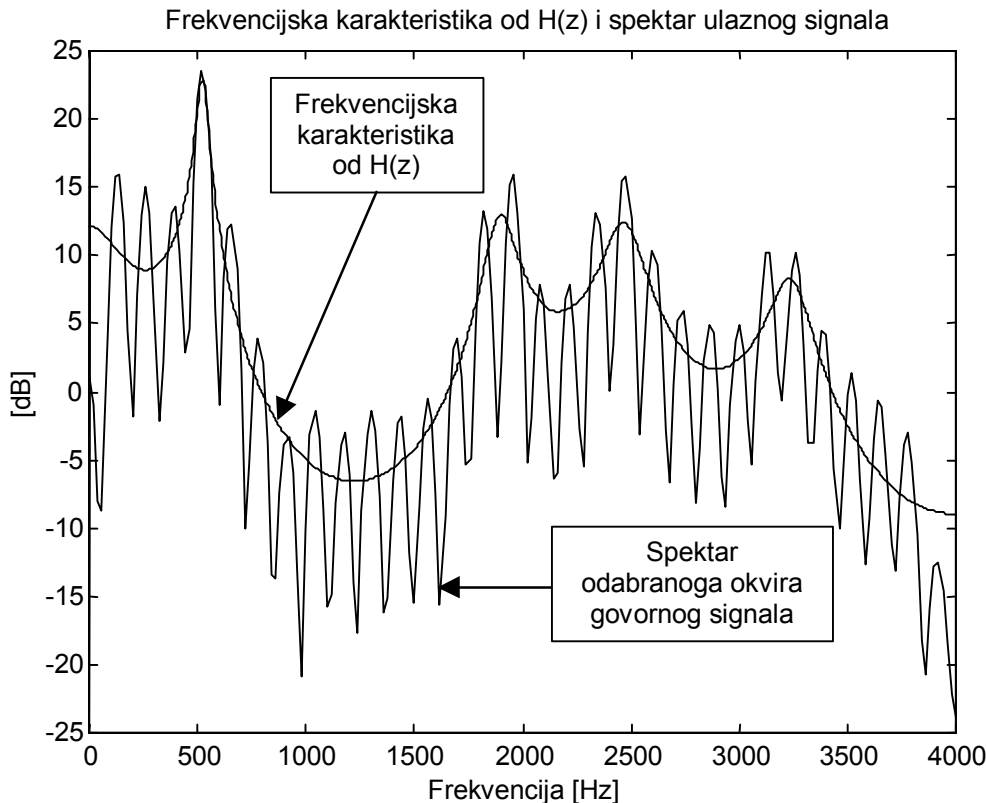


Slika 9.2-8 Amplitudno-frekvencijska karakteristika prijenosne funkcije $H(z)$

Rezonantna nadvišenja javljaju se na frekvencijama koje odgovaraju frekvencijama polova. Za rezonantno nadvišenje u digitalnoj obradi govora upotrebljava se naziv *formant*. Zbog gubitaka koji postoje u vokalnom traktu, formanti višega reda imaju zatupljenije vrhove i veću širinu što je i vidljivo sa slike 9.2-8. Ako sada pomoću naredbi:

```
sp=fft(ee);
lo2=20*log10(abs(sp(1:N+1)));
ff=[0:N]/N*fs/2;
```

izračunamo brzu Fourierovu transformaciju promatranog okvira govornoga signala te ga iscrtamo na istome grafu sa frekvencijskom karakteristikom prijenosne funkcije $H(z)$, kao što je to prikazano na slici 9.2-9, vidjeti ćemo da amplitudno-frekvencijska karakteristika od $H(z)$ opisuje ovojnici (*eng. envelope*) spektrograma za promatrani okvir signala.



Slika 9.2-9 Frekvencijska karakteristika od $H(z)$ i spektrogram odabranog okvira govornog signala

Na slici 9.2-10 prikazan je spektar rezidualnog signala koji je dobiven izvođenjem naredbi:

```
spr=fft(rezid);
lo3=20*log10(abs(spr(1:N+1)));
plot(ff,lo3);
```

Izgled toga spektrograma određen je korištenim vremenskim otvorom. Spektar vremenskoga otvora ponovio se na razmaku pitch perioda, pa je sumiranje glavnih i bočnih latica brojnih vremenskih otvora stvorilo takav kontinuirani spektar.

Izvođenjem naredbi:

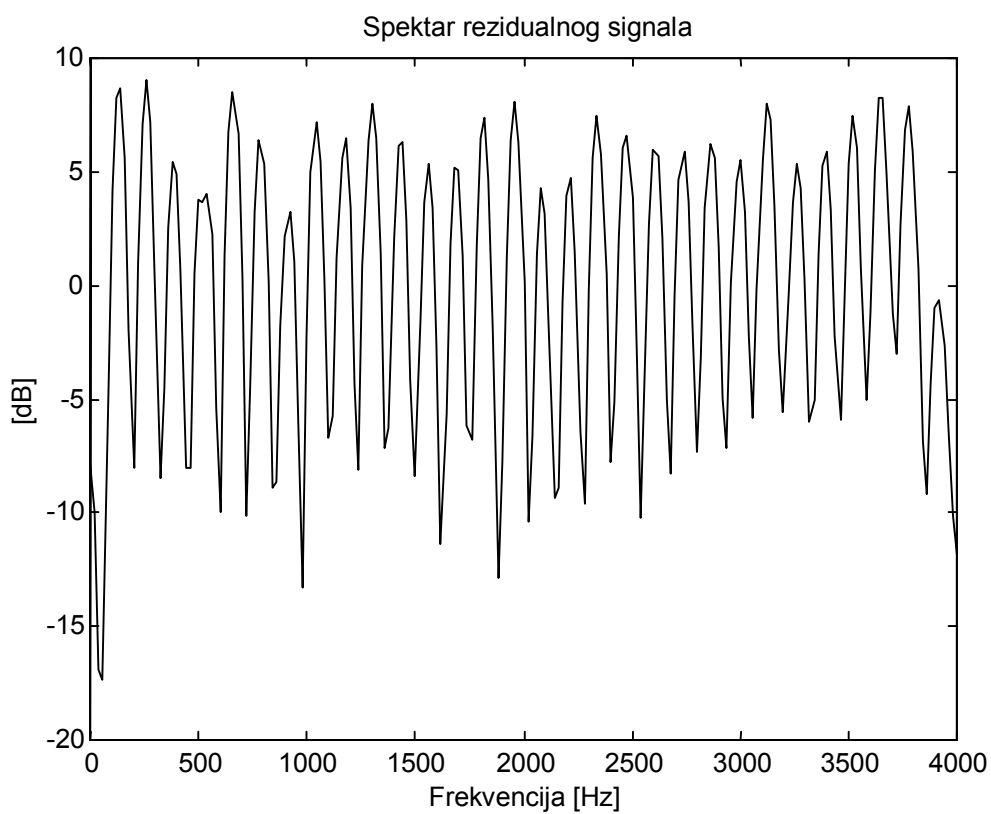
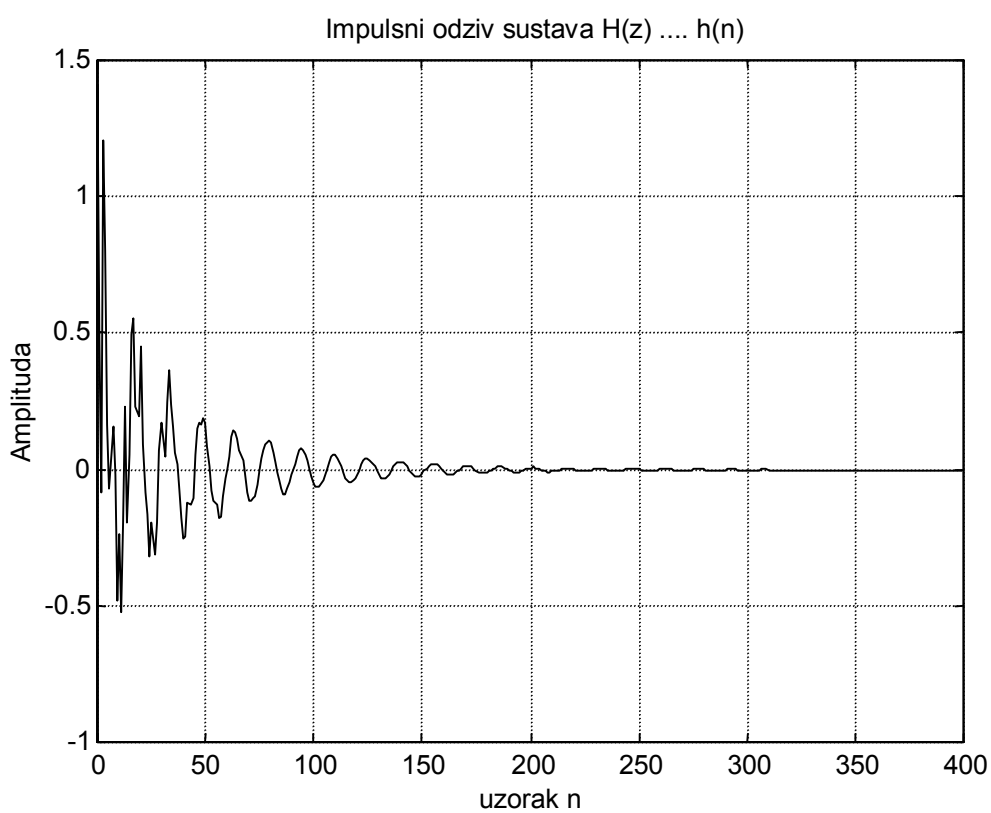
```
imp_od=filter(sqrt(E),a,[1 zeros(1,2*N-1)]);
plot([0:2*N-1],imp_od);
```

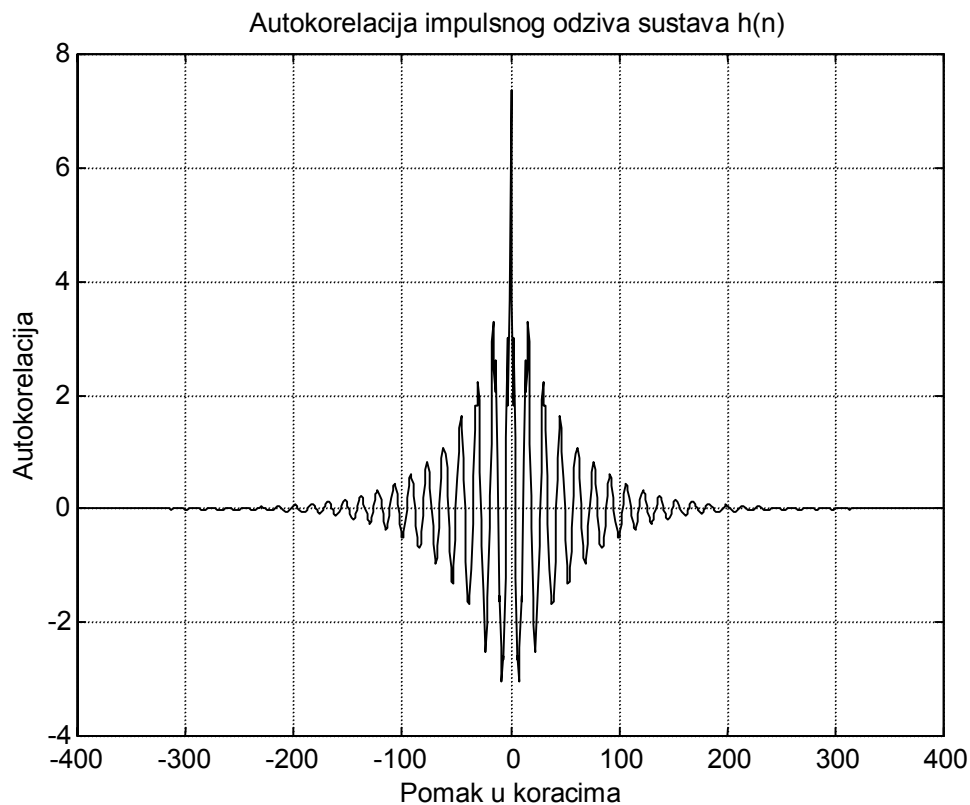
izračunava se i zatim iscrtava impulsni odziv digitalnog filtra $H(z)$. Dobiveni rezultat prikazan je na slici 9.2-11, sa koje se može vidjeti da impulsni odziv trne prema beskonačno što je još jedan dokaz da je filter $H(z)$ stabilan.

Autokorelacija impulsnog odziva filtra dobivena je izvođenjem naredbi:

```
rrimp=xcorr(imp_od);
rrimp=(rrimp(:))';
plot(xrr,rrimp);
```

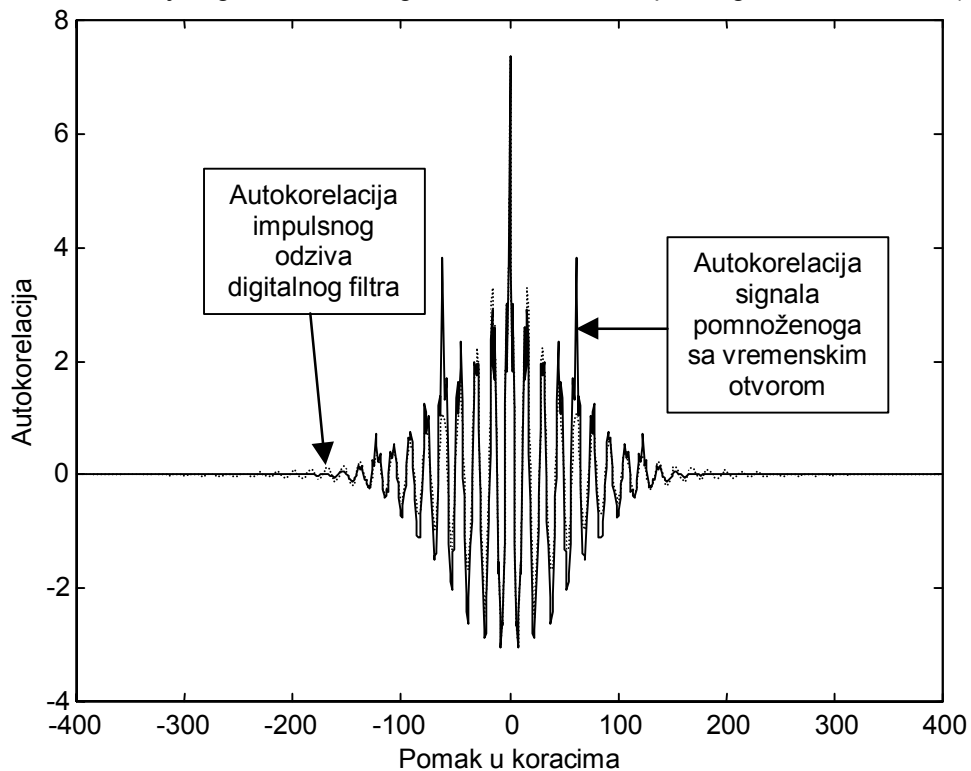
, a njezin izgled može se vidjeti na slici 9.2-12.

**Slika 9.2-10** Spektar signala pogreške predikcije**Slika 9.2-11** Impulsni odziv digitalnog filtra $H(z)$



Slika 9.2-12 Autokorelacija impulsnog odziva digitalnog filtra $H(z)$

Autokorelacija signala množenog vr. otv. i autokor. impulsnog odziva sustava $h(n)$



Slika 9.2-13 Usporedba autokorelacije signala i autokorelacije impulsnog odziva digitalnog filtra $H(z)$

Međutim, za nas je mnogo značajnije da radi usporedbe na istoj slici prikažemo autokorelaciju impulsnog odziva filtra $H(z)$ i autokorelaciju signala pomnoženoga sa vremenskim otvorom koja je bila ranije prikazana na slici 9.2-3. Taj problem rješava naredba:

```
plot(xrr, [rr' rrimp'] );
```

, a rješenje prikazuje slika 9.2-13 sa koje se vidi da u najvećem dijelu postoji dobro poklapanje rezultata obje autokorelacije. Prije iscrtavanja ove slike, program je u komandni prozor ispisao vrijednosti autokorelacije za pomake od 0 do P za signal pomnožen sa vremenskim otvorom i impulsni odziv filtra $H(z)$. Njihovom usporedbom možemo utvrditi da su jednakih iznosa, a to vrijedi za bilo koji red prediktora P .

9.3 Zadaci za samostalno izvođenje

Zadatak 1.

Koristeći prediktor desetoga reda ($P=10$) kakav je korišten u primjeru vježbe, ponoviti postupak za neki drugi samoglasnik. Dobivene slike pomoću COPY+PASTE postupka prenesite u *MS Word*, a zatim tu datoteku snimate na disk. Rezultate koje *Matlab* ispisuje u komandni prozor možete snimiti tako da prije pokretanja programa *dog3.m* zadate naredbu *diary*. Za detaljan opis korištenja navedene naredbe utipkajte

```
>>help diary
```

Kolika je sumarna kvadratna pogreška predikcije u tom slučaju?

Zadatak 2.

Ponoviti postupak iz primjera vježbe sa samoglasnikom *e*, ali za neki drugi red prediktora (npr. $P=4$, $P=6$, $P=14$, $P=20$) i ustanoviti kakav utjecaj red prediktora ima na sumarnu kvadratnu pogrešku predikcije.

KRAJ

10. LEVINSON-DURBINOV ALGORITAM I ANALIZA DOBITKA PREDIKCIJSKOG KODIRANJA

Student: Krešimir Kočiš

POČETAK

KRAJ

11. KOVARIJANTNI POSTUPAK LINEARNE PREDIKCIJE

Student: Drago Kozic

POČETAK

Kao što je poznato predikcija ili predviđanje je metoda kojom pokušavamo uzorke govornog signala aproksimirati (predvidjeti) linearnom kombinacijom prethodnih uzoraka. Zadatak se svodi na kraju na određivanje alfa koeficijenata reda predikcije p ($\alpha_1, \alpha_2, \dots, \alpha_p$) sa što manjom pogreškom $e(n)$. Tu će biti obrađena kovarijantna metoda linearne predikcije.

11.1 Kovarijantna metoda predikcije i Cholesky-jeva kompozicija

Za razliku od autokorelacijskog postupka linearne predikcije kod kovarijantnog nema ograničavanja signala vremenskim otvorima. Tu jednostavno fiksiramo interval sumacije pogreške predikcije odnosno definiramo granice indeksa m :

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) \quad (11.1-1)$$

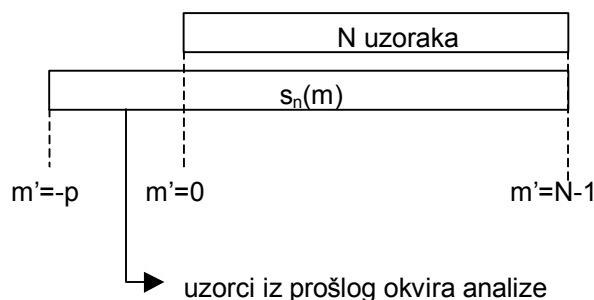
Iste granice imamo naravno i u izrazu za matricu Φ :

$$\Phi_m(i,k) = \sum_{m=0}^{N-1} s_n(m-i)s_n(m-k) \quad (11.1-2)$$

gdje indeksi leže u intervalima $1 \leq i \leq p$ odnosno $0 \leq k \leq p$. Uz preindeksiranje jednadžbe sa $m' = m - i$, istu možemo zapisati kao:

$$\Phi_m(i,k) = \sum_{m'=-i}^{N-1-i} s_n(m')s_n(m'+i-k) \quad (11.1-3)$$

Najdonja granica signala koja nam je potrebna je $-p$ dok je najgornja $N-1$. Drugim riječima, naš najlijeviji uzorak će biti $s_n(-p)$ a najdesniji $s_n(N-1)$ što ujedno predstavlja i raspon uzoraka jednog okvira analize. To se može prikazati i shematski.



Slika 11.1-1 Prikaz raspona uzoraka jednog okvira analize

Cilj nam je postići minimalnu srednju kvadratnu pogrešku predikcije i kod kovarijantnog postupka predikcije. Zato parcijalne derivacije E_n izjednačimo s nulom kako bi dobili sustav p jednadžbi s p nepoznanica:

$$\sum_{k=1}^p \alpha_k \Phi_n(i, k) = \Phi_n(i, 0) \quad (11.1-4)$$

Indeksi leže u intervalima $1 \leq i \leq p$ odnosno $1 \leq k \leq p$. Jednadžbu (11.1-4) možemo napisati i u matičnom obliku, gdje je α stupac s p elemenata, $\Phi_n(i, k)$ matrica dimenzije $p \times p$ a $\Phi_n(i, 0)$ slobodni stupac s p elemenata.

Određivanjem matrice Φ_n vidimo da je ona simetrična. Međutim kod ovog postupka članovi u pojedinoj paralelnoj dijagonali nisu jednaki kao što je to slučaj kod autokorelacijskog postupka. Ali i tu postoji pojednostavljenje u pogledu određivanja matrice. Svaki donji desni član može se izračunati na osnovi prethodnog gornjeg lijevog člana prema jednadžbi:

$$\Phi_n(i+1, k+1) = \Phi_n(i, k) + s_n(-i-1)s_n(-k-1) - s_n(N-i-1)s_n(N-k-1) \quad (11.1-5)$$

Trebamo dakle izračunati samo prvi stupac matrice i onda prema formuli ostale članove. Vidimo da nam je potreban i član $\Phi_n(0, 0)$.

Matricu sada imamo i potrebno ju je riješiti odnosno odrediti koeficijente prediktora $\alpha_1, \alpha_2, \dots, \alpha_p$. Uz oznaku slobodnog stupca sa Ψ možemo na kraći način zapisati matičnu formulu kao:

$$\Phi_n \alpha = \Psi \quad (11.1-6)$$

Matrica Φ_n ima dobro svojstvo što je pozitivno definitna simetrična matrica (njezine vlastite vrijednosti su pozitivne). Takva matrica se rješava metodom Cholesky-jeve kompozicije odnosno metodom drugog korijena. Matricu raspišemo kao produkt tri matrice V, D i V^T koje imaju sljedeća svojstva:

V ...dolnja trokutasta matrica s jedinicama na dijagonali

D ...dijagonalna matrica

V^T ...transponirana matrica V odnosno gornja trokutasta matrica

$$V = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ V_{21} & 1 & 0 & \dots & 0 \\ V_{31} & V_{32} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ V_{p1} & V_{p2} & V_{p3} & V_{p,p-1} & 1 \end{bmatrix} \quad D = \begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ 0 & 0 & d_3 & \dots & 0 \\ \dots & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & 0 & d_p \end{bmatrix}$$

Produkt te tri matrice iznosi za npr. $p=3$:

$$\mathbf{V} = \begin{bmatrix} d_1 & d_1 V_{21} & d_1 V_{31} \\ d_1 V_{21} & d_1 V_{21}^2 + d_2 & d_1 V_{21} V_{31} + d_2 V_{32} \\ d_1 V_{31} & d_1 V_{21} V_{31} + d_2 V_{32} & d_1 V_{31}^2 + d_2 V_{32}^2 + d_3 \end{bmatrix}$$

Daljnjim raspisivanjem jednadžbi s elementima matrice dolazimo do dvije glavne formule pomoću kojih možemo odrediti sve članove matrica \mathbf{V} odnosno \mathbf{V}^T i \mathbf{D} :

$$V_{ik} d_k = \Phi_n(i, k) - \sum_{j=1}^{k-1} V_{ij} d_j V_{kj} \quad (11.1-7)$$

$$d_i = \Phi_n(i, i) - \sum_{j=1}^{i-1} d_j V_{ij}^2 \quad (11.1-8)$$

Odamah primjetimo specijalni slučaj za $i=1$ kada je $d_1 = \Phi_n(1, 1)$.

Pronalaženjem matrica dobili smo novi sustav jednadžbi koji u matričnom obliku glasi:

$$\mathbf{VDV}^T \boldsymbol{\alpha} = \boldsymbol{\Psi} \quad (11.1-9)$$

Uvedemo novi stupac $\mathbf{Y} = \mathbf{DV}^T \boldsymbol{\alpha}$ i napravimo dva sustava jednadžbi:

$$\begin{array}{lll} 1. \text{ sustav:} & \mathbf{VY} = \boldsymbol{\Psi} & \Rightarrow \mathbf{Y} \\ 2. \text{ sustav:} & \mathbf{DV}^T \boldsymbol{\alpha} = \mathbf{Y} & \Rightarrow \mathbf{V}^T \boldsymbol{\alpha} = \mathbf{D}^{-1} \mathbf{Y} \end{array}$$

Koristi se tzv. unazadna supstitucija, gdje rješenja od prije stavljamo u izraz za sljedeće rješenje. Tako izlaze jednadžbe za određivanje stupca \mathbf{Y} i na kraju dobivanje stupca s koeficijentima predikcije $\boldsymbol{\alpha}$.

$$Y_i = \Psi_i - \sum_{k=1}^{i-1} V_{ik} Y_k \quad 2 \leq i \leq p \quad (11.1-10)$$

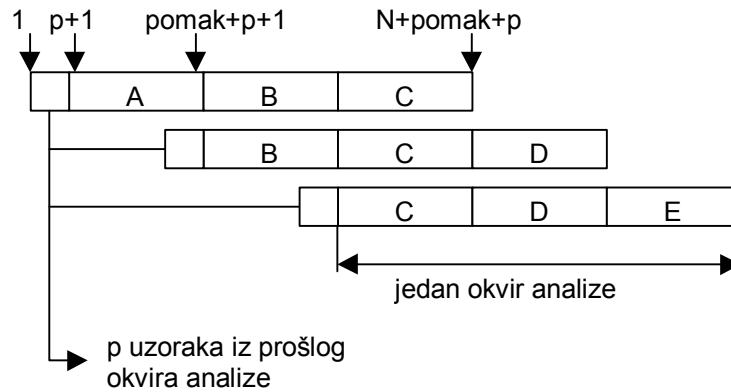
$$\alpha_i \left(\frac{Y_i}{d_i} \right) - \sum_{k=i+1}^p V_{ki} \alpha_k \quad p-1 \leq i \leq 1 \quad (11.1-11)$$

Kada svu teoriju prenosimo u Matlab stvari nisu tako jednostavne. Prva prepreka je ta što Matlab ne poznaje indeksa 0. Znači da moramo uvijek svaki indeks povećati za jedan što može uzrokovati poteškoće. U Matlabu se izračunavaju koeficijenti prediktora kojima treba promijeniti predznak jer su ti koeficijenti u prijenosnoj funkciji filtra negativni.

Da bi što brže i što lakše došli do matrice Φ_n iskoristit ćemo njezina svojstva kao i neke pomoćne trikove. U našem izvornom signalu koji je dužine oko 3,5 do 5 sekundi ima uzorkovanje od 8 kHz približno 30000 uzoraka. Za dužinu okvira dovoljno je uzeti 250 uzoraka. Uvijek kada se pomaknemo sa okvirom naprijed, novi okvir prekriva dio starog. To možemo iskoristiti tako da uzmemo za korak npr. 1/3 (faktor=3) od broja uzoraka okvira. Tako uz korak 80 i faktor 3 imamo $N=240$ uzoraka po okviru. Zbog prekrivanja dvije trećine

novog okvira sa starim pojavi se ideja da taj dio ne računamo ponovo. Trebalo bi nekako zapamtiti sve tri trećine okvira i u svakom koraku računati samo jednu novu trećinu. Tako od starog oduzmemo prvu trećinu i dodamo samo novu zadnju trećinu.

To se može dobro prikazati sljedećom shemom:



Slika 11.1-2 podjela pojedinih okvira analize na trećine

Vidimo da je uvijek prisutnih i p uzoraka iz prethodnog okvira. Strelicama su na prvom okviru analize prikazani indeksi uzoraka signala kakvi će biti u Matlabu. Pomak je indeks koji ovisi o trenutnom okviru. Broj uzoraka cjelokupnog signala se u Matlabu računa kao:

```
sl=max(size(dat));
```

Broj okvira za analizu je:

```
f=floor((sl-N-p)/korak)+1;
```

a pomak:

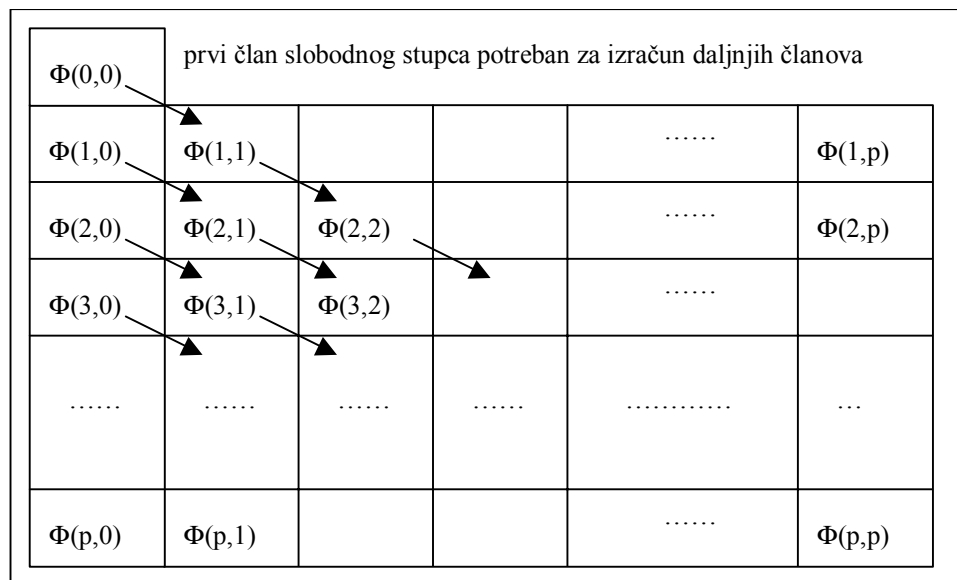
```
pomak=(brojac-1)*korak+p+1;
```

gdje **brojac** ima vrijednosti od 1 do f .

Pohranu trećina uzoraka okvira provedemo pomoću cirkularnog iščitavanja vrijednosti iz stupaca (u našem slučaju 3 stupca) matrice dimenzije $(p+1) \times 1$. Matricu nazovemo npr. **st** (skraćeno od stanje). Da bi što brže i što lakše došli do elemenata cijeloga signala upotrebljavamo indeksiranje trećine uzoraka okvira. To se u Matlabu napravi na sljedeći način:

```
indeksi=[0:-1:-p]'*ones(1,korak)+ones(p+1,1)*[0:korak-1];
```

Nakon toga možemo odrediti slobodni stupac pomoću jednadžbe (11.1-2) uz napomenu da sada indeks i ima granice $0 \leq i \leq p$ odnosno $1 \leq i \leq p+1$ za Matlab jer trebamo član s indeksima (1,1) slobodnog stupca, koji je ujedno i prvi stupac matrice Φ_n , da bi mogli izračunati donji desni član s indeksima (2,2) (svi indeksi se od sada nadalje odnose na indeksiranje u Matlabu). Također napravimo fintu da prvi stupac, trenutno sa najstarijim indeksom, pustimo prazan. Da bi situacija s indeksima bila jasnija prikazimo shematski izgled matrice Φ_n :



slobodni stupac ψ i
ujedno prvi stupac
matrice Φ_n ($\Phi_n(i,1)$;
 $1 \leq i \leq p+1$)

matrica $\Phi_n(i,j)$
 $1 \leq i \leq p+1$
 $1 \leq j \leq p+1$
odnosno dimenzije $(p+1) \times (p+1)$

Slika 11.1-3 Prikaz izračuna
elemenata matrice
 Φ_n

Na slici 11.1-3 strelice predstavljaju iz kojeg prethodnog se prema jednadžbi (11.1-5) izračunava sljedeći član matrice.

Kada je matrica Φ_n cijela izračunata pomoću jednadžbe (11.1-6) određujemo alfa koeficijente za sve okvire analize. Nakon toga možemo izračunati ostale matrice tako da se pomičemo po signalu uvijek za korak naprijed i odredimo pomak po signalu. Prema slici 11.1-2 treba izračunati samo jednu novu trećinu sljedećeg okvira jer njegove prve dvije već imamo. Odredimo najstariji indeks matrice za cirkularno iščitavanje **st** i novi slobodni stupac oduzimanjem stupca sa najstarijim indeksom i pribrajanjem novo izračunate trećine prethodnom slobodnom stupcu, te postupkom prema slici 11.1-3 i jednadžbi (11.1-5) sve ostale elemente matrice. Za izračun alfa koeficijenata upotrebljavamo Cholesky-jevu kompoziciju pomoću jednadžbi (11.1-7), (11.1-8), (11.1-10) i (11.1-11), gdje također trebamo paziti da promijenimo odgovarajuće indekse prema potrebama Matlaba.

Sada možemo sve to ukratko prikazati jednim primjerom tako da ćemo ići po programu i zaustaviti se na zanimljivim detaljima. Za primjer uzmimo zvučni signal “drago1.wav”. Programi koje upotrebljavamo su “poziv_covar.m” koji poziva funkciju “Lpc.covar.m”, dok taj poziva funkciju “Cholesky.m” i “nestab.m”.

U programu “poziv_covar.m” odredimo parametre LPC analize (korak, faktor, red prediktora,...) dok je $N = \text{faktor} * \text{korak}$. Također je definirana donja granična frekvencija visoko propusnog filtra za otklanjanje nisko-frekvencijskih komponenti. Program onda zove funkciju “Lpc.covar.m”. Poslije otklanjanja nisko-frekvencijskih komponenti počinje glavni dio programa a to je LPC analiza kovarijantnom metodom.

Najprije se odrede indeksi potrebni za izračunavanje prvog stupca matrice kovarijance što smo već prikazali prije. Nakon toga se izračunava matrica stanja za pohranu djelomičnih stupaca (trećina okvira) kovarijance gdje se, kako je već prije spomenuto, prvi stupac pušta prazan:

```
st=zeros(p+1,faktor);
for i=1:faktor-1,
    st(:,i+1)=dat(indeksi+(i-1)*korak+p+1)*dat([0:korak-1]+(i-1)*korak+p+1)';
end;
```

i odredi inicijalni stupac kovarijance ($A+B+C$ ako gledamo sliku 11.1-2; C je prazan stupac i ima trenutno najstariji indeks te se na početku zbrajaju samo $A+B$):

```
% Inicijalni stupac kovarijance A+B
Fi=sum(st')';
```

kako bi mogla početi LPC analiza za sve okvire analize.

Za svaki okvir analize radimo isti postupak. Najprije izračunamo dodatak za stupac kovarijance za novi pod-blok i indeks najstarijeg pod-bloka:

```
% Izracunavanje dodatka na stupac kovarijance za novi pod-blok
i=brojac+faktor-1;
novo=dat(indeksi+(i-1)*korak+p+1)*dat([0:korak-1]+(i-1)*korak+p+1)';

% Indeks najstarijeg pod-bloka kojeg treba odbiti i zatim preko
% njega prepisati novo-izracunati podblok
indeks_najst=rem(i,faktor)+1;
```

Nakon toga izračunamo novi stupac kovarijance (slobodni stupac) oduzimanjem stupca pod-bloka s najstarijim indeksom i dodavanjem novo izračunatog pod-bloka:

```
% Azuriranje stupca kovarijance
Fi=Fi-st(:,indeks_najst)+novo;
```

prepišemo novi stupac preko onoga s najstarijim indeksom:

```
% Upis dodatka preko najstarijeg u matricu stanja
st(:, indeks_najst)=novo;
```

te formiramo matricu kovarijance na osnovu prvog (slobodnog) stupca:

```
% Formiranje matrice kovarijance na osnovu prvog stupca
FI=zeros(p+1,p+1);

% Upisi prvi redak i prvi stupac
FI(:,1)=Fi;
FI(1,:)=Fi';

% Za sve ostale elemente racunaj rekursivno
for kk=0:p-1,
    ii=[kk:p-1];
    pomak=p+(brojac-1)*korak+1;
    FI(ii+2, kk+2)=FI(ii+1, kk+1)+dat(pomak-ii-1)'*dat(pomak-kk-1) - ...
        dat(pomak+N-ii-1)'*dat(pomak+N-kk-1);
    FI(kk+2, ii+2)=FI(ii+2, kk+2)';
end;
fi=FI(2:p+1, 2:p+1);          % samo matrica fi
psi=FI(2:p+1, 1);             % samo slobodni stupac
```

Zatim možemo pozvati funkciju Cholesky, kojom izračunavamo alfa koeficijente predikcije trenutnog okvira. Ulazni parametri te funkcije su matrica Φ i slobodni stupac Ψ . Izlazni podaci su alfa koeficijenti trenutnog okvira te stupci \mathbf{Y} i \mathbf{d} koji će nam kasnije trebati. Ta funkcija izgleda ovako:

```
function [alfa,Y,ds]=Cholesky(fi,psi)

%%
%           CHOLESKY-jeva KOMPOZICIJA
%%
[p,dumm1]=size(fi);
[dumm2,dumm3]=size(psi);

if (abs(p-dumm1)+ abs(p-dumm2) + abs(1-dumm3)),
    error('Krive ulazne dimenzije matrica');
end;

d=zeros(p,p);                %stupac D,tu kao matrica (zbog Matlaba)
V=eye(p);                    %matrica V sa jedinicama na dijagonali
suma=0;                      %pomoćna varijabla

%početak Cholesky-jeve metode
for k=1:p,
    for x=1:k-1,
        suma=suma+V(k,x)^2*d(x,x);
    end;
    d(k,k)=fi(k,k)-suma;
    suma=0;
    l=[k+1:p];
    for m=1:k-1,
        suma=suma+V(l,m)*d(m,m)*V(k,m);
    end;
    V(l,k)=(fi(l,k)-suma)/d(k,k);
    suma=0;
```



```

end;

% Dobivanjem novog sustava određujemo Y stupac
Y=zeros(p,1);
Y(1)=psi(1); %prvi član Y stupca

% Ostali elementi vektora Y
for l=2:p,
    Y(l)=psi(l)-V(l,1:l-1)*Y(1:l-1);
end;

%izračun alfa koeficijenata trenutnog stupca
alfa=zeros(p,1);
alfa(p)=Y(p)/d(p,p); %zadnji član stupca alfa

%ostali članovi
for l=p-1:-1:1,
    alfa(l)=(Y(l)/d(l,l))-V(l+1:p,l)'*alfa(l+1:p);
end;

ds=diag(d);

```

Iz predavanja nam je poznato da kovarijantni postupak linearne predikcije signala ne daje stabilan filter. U programu “nestab.m” računaju se ti nestabilni okviri analize. Nestabilni okviri su oni kojima polovi prediktora izlaze iz jedinične kružnice. Potrebno je dakle izračunati polove prediktora svih okvira analize. Najprije možemo izračunati korijene i njihov maksimalni modul za sve okvire analize što u Matlabu glasi:

```

max_a(i)=max(abs(roots(a_m(i,:))))); % odredi korijene i njihov max.
% modul

```

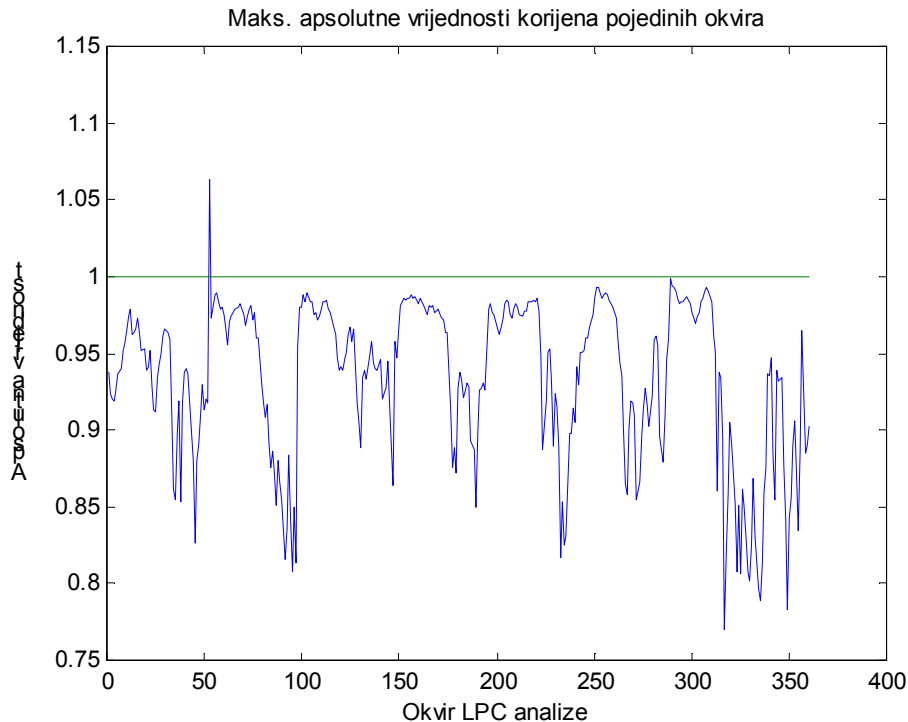
Prolazom svih okvira i upisom koda:

```

plot([1:bf],max_a,[1 bf],[1 1]);

```

grafički možemo prikazati maksimalne apsolutne vrijednosti pojedinih okvira gdje možemo vidjeti moguće nestabilne okvire tj. one kojih apsolutne vrijednosti prelaze jedinicu (slika 11.1-4).



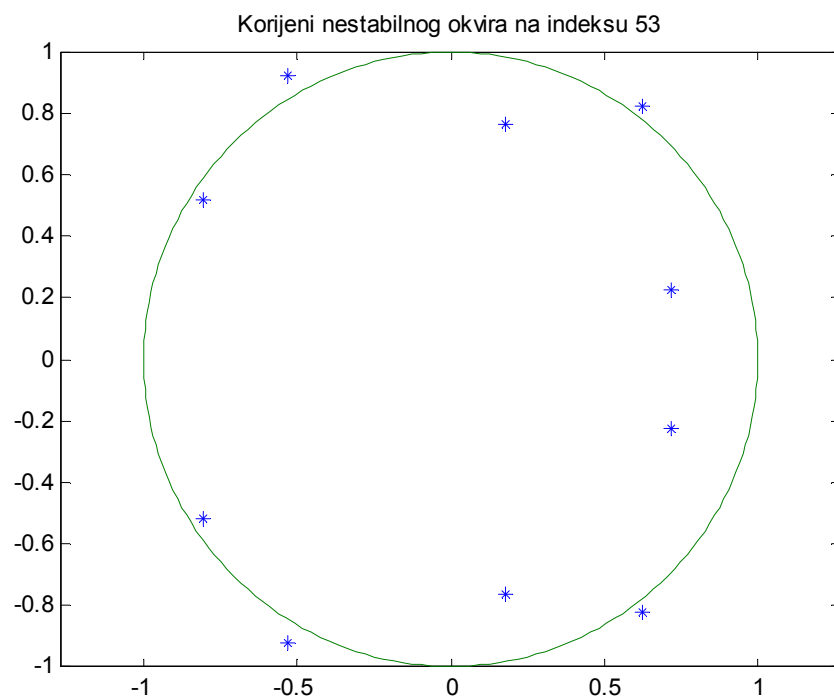
Slika 11.1-4 Polovi jednog od okvira prikazani u jediničnoj kružnici

Nakon toga odaberemo one najkritičnije sa najvećom apsolutnom vrijednošću (okvire koji su najnestabilniji). U Matlabu to glasi ovako:

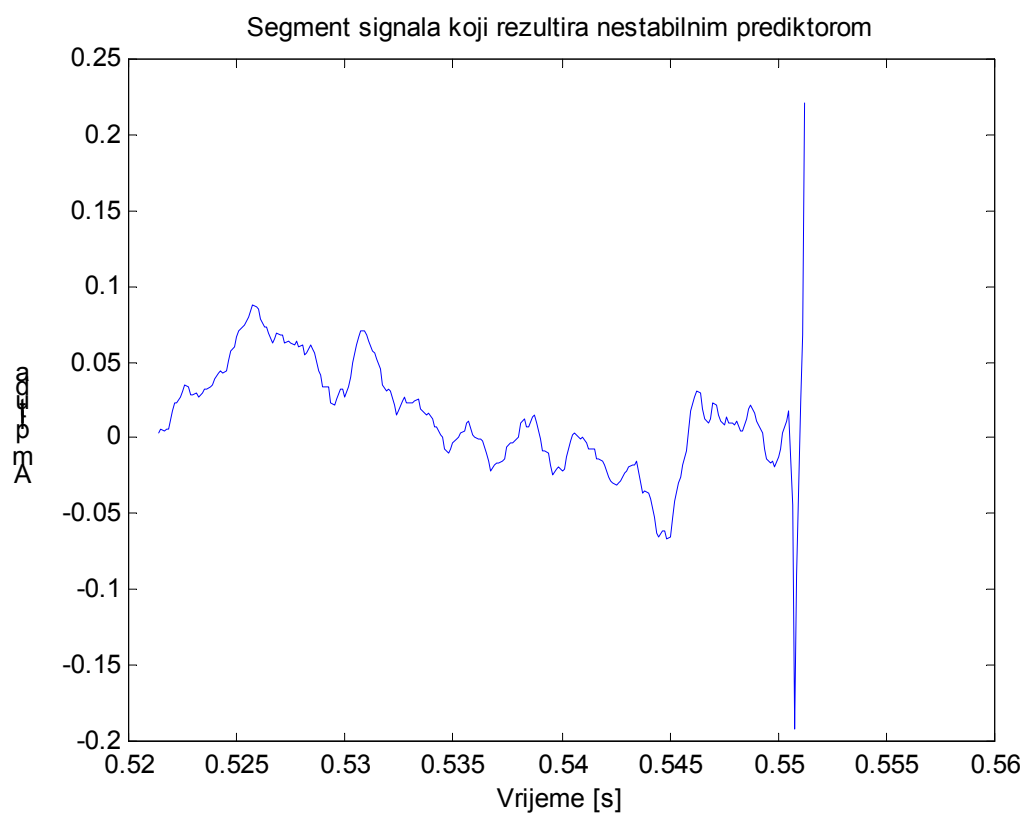
```
[max_abs_vr br_okv]=max(max_a);           % Nadji iznos i poziciju max. preko
                                           % svih okvira
max_korijeni=roots(a_m(br_okv,:));        % Zapamti te najgore (s najvećim
                                           % modulom) korijene

okvir=find(max_a>=1);                     % Odredi indekse svih okvira kod
                                           % kojih je barem jedan
                                           % pol van jedinične kružnice
brojac=length(okvir);                     % Ukupni broj nestabilnih okvira
```

Na kraju možemo onda još grafički prikazati najnestabilnije okvire. Najprije u jediničnoj kružnici i naknadno dio signala gdje dolazi do nepravilnosti. Vidimo (slika 11.1-6) da dolazi (u većini primjera) kod nestabilnosti do raspirivanja signala što zapravo prouzrokuje tu nestabilnost (na drugim primjerima se to možda i bolje vidi).



Slika 11.1-5 Korijeni nestabilnog okvira analize



Slika 11.1-6 Segment signala s nestabilnim predikt.

Vidimo da je 53. okvir nestabilan, jer ima čak 2 konjugirano kompleksna pola izvan jedinične kružnice (slika 11.1-5).

Zadatak 1.:

- Napisati Matlab program za određivanje matrice Φ_n i koeficijenata predikcije α kovarijantnom metodom predikcije i Cholesky-jevom kompozicijom
- Napisati program u Matlabu koji će prebrojiti sve nestabilne okvire
- Grafički prikazati jedan od tih okvira analize u jediničnoj kružnici (npr. onaj s najvećim odstupanjem u jediničnoj kružnici), dio signala gdje dolazi do te nestabilnosti te maksimume absolutne vrijednosti korijena svih okvira i naći gdje se vidi nestabilnost.
- Odrediti postotak nestabilnih LPC okvira

11.2 Utjecaj smanjenja reda predikcije na rezidualnu energiju i stabilnost filtra dobivenog kovarijantnom metodom predikcije

Rezidualna energija odnosno pogreška predikcije pojedinog okvira analize računa se prema sljedećem izrazu:

$$E_n = \sum_{m=0}^{N-1} e_n^2(m) = \Phi_n(0,0) - \sum_{k=1}^p \frac{Y_k^2}{d_k} \quad (11.2-1)$$

Ovim izrazom možemo provjeriti do kojeg reda predikcije ima smisla ići ako smanjujemo red predikcije p . Kakva se krivulja dobiva vidjet ćemo tijekom laboratorijske vježbe.

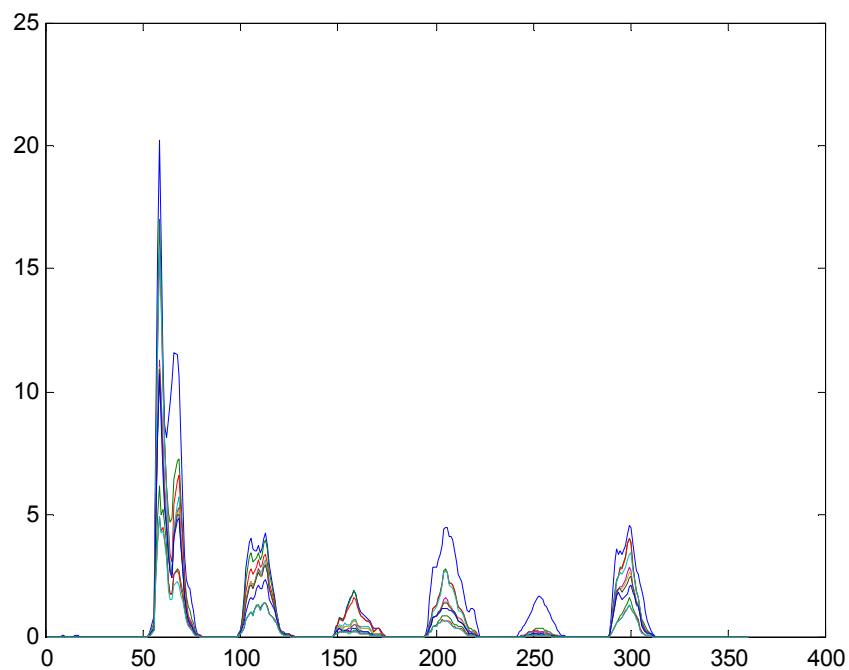
Prema jednadžbi (11.2-1) izraz za izračun rezidualne energije obzirom na red predikcije glasio bi:

```
Enn(brojac,:)=(FI(1,1)-[0; cumsum((y.^2)./d)]') ; % rezidualna energija
% obzirom na red pred.
```

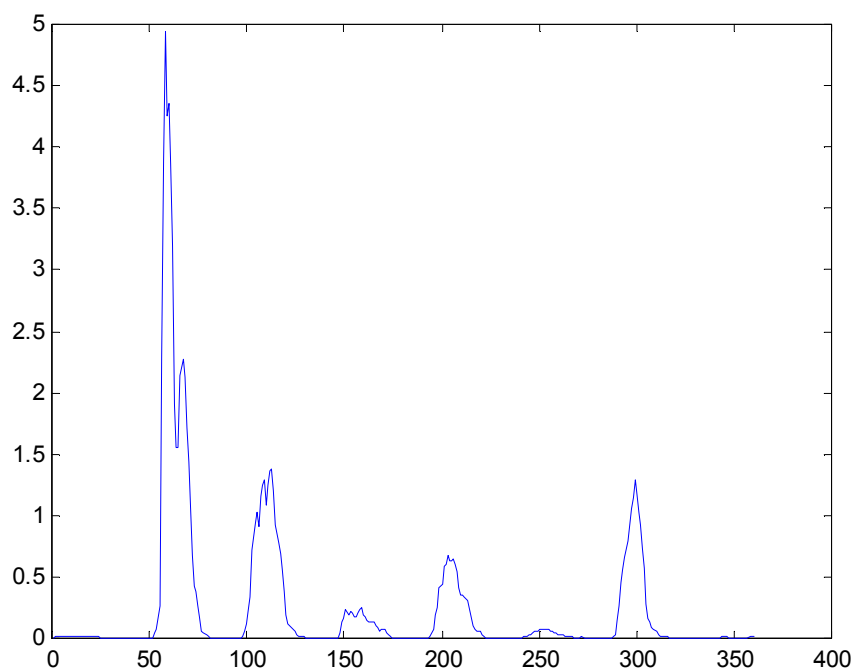
Ovaj rezultat pokazuje nam u biti koliki utjecaj ima red predikcije na točnost predviđanja signala odnosno do kojeg reda predikcije ima uopće smisla ići. To se još najbolje vidi ako stvar prikažemo grafički (slika 11.2-1).

Možemo izračunati i grafički prikazati i rezidualnu energiju za najviši red predikcije (slika 11.2-2):

```
En_m=Enn(:,p+1); % rezidualna energija
% za najveći red pred.
```



Slika 11.2-1 Utjecaj reda pred. na rezidualnu energiju



Slika 11.2-2 Rezidualna energija za najviši red pred.

Zadatak 2.:

- a) Grafički prikazati ovisnost rezidualne energije o redu predikcije i rezidualnu energiju za najveći red predikcije.

11.3 Rješenja:**1.a)**

Treba samo spojiti sve djelove programa opisane i zapisane u gornjoj teoriji sa nešto malo dodataka.

1.b)

```
global dat;
load dragol; % Ucitavanje rezultata LPC analize

korak=param(1);
faktor=param(2);
N=faktor*korak;

[bf,p]=size(a_m);
p=p-1;

max_a=zeros(bf,1); % vektor s max. modulima korijena
                    % svakog okvira analize
for i=1:bf, % prodji za sve okvire ....
    max_a(i)=max(abs(roots(a_m(i,:)))); % odredi korijene i njihov max.
    % modul
end;

figure(1);
plot([1:bf],max_a,[1 bf],[1 1]);
title('Maks. apsolutne vrijednosti korijena pojedinih okvira');
xlabel('Okvir LPC analize');
ylabel('Apsolutna vrijednost');
pause;

[max_abs_vr br_okv]=max(max_a); % Nadji iznos i poziciju max. preko
                                % svih okvira
max_korijeni=roots(a_m(br_okv,:)); % Zapamti te najgore (s najvećim
                                % modulom) korijene

okvir=find(max_a>=1); % Odredi indekse svih okvira kod
                    % kojih je barem jedan
                    % pol van jedinичne kruznice
brojac=length(okvir); % Ukupni broj nestabilnih okvira

fprintf('\n\n\n'); % Ispis rezultata
for i=1:brojac,
    fprintf('%d. nestabilni je %d. okvir \n',i,okvir(i));
end;

fprintf('\nMaksimalna apsolutna vrijednost: %d\n',max_abs_vr);
fprintf('\nKorijeni s maksimalnom apsolutnom vrijednosti:\n');
disp(max_korijeni);
fprintf('\nOkvir s maksimalnom apsolutnom vr.: %d\n',br_okv);
fprintf('\nUkupni broj nestabilnih okvira: %d, što iznosi %5.2f posto
nestabilnih okvira\n', ...
                                brojjac,(brojac/bf*100));
```

```

x=sin(2*pi*[0:200]/200);
y=cos(2*pi*[0:200]/200);

[dummy,fs]=WAVREAD2(wavefile);
sl=max(size(dat));
dat=dat(1:sl)/max(abs(dat));

for i=1:brojac,
    max_korijeni=roots(a_m(okvir(i),:));
    figure(2);
    plot(real(max_korijeni),imag(max_korijeni),'*',x,y); axis('equal');
    str=sprintf('Korijeni nestabilnog okvira na indeksu %d',okvir(i));
    title(str);

    figure(3);
    pom=p+(okvir(i)-1)*korak;
    plot([1:N]+pom)/fs,dat([1:N]+pom));
    title('Segment signala koji rezultira nestabilnim prediktorom');
    xlabel('Vrijeme [s]');
    ylabel('Amplituda');
    pause;
end;

```

1.c)

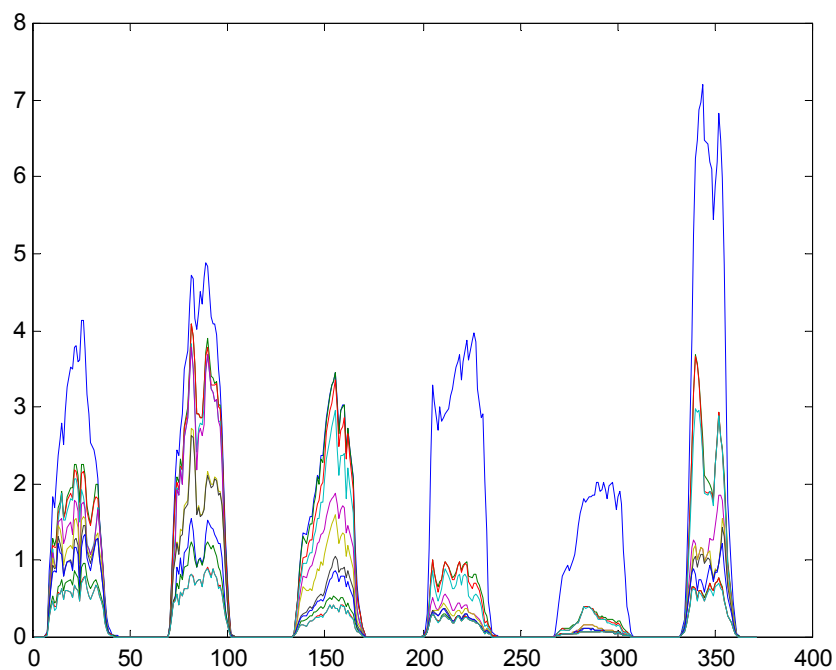
Okvir s maksimalnom nestabilnošću u jediničnoj kružnici prikazan je u teorijskom djelu na slici 11.1-5, dio signala gdje dolazi do nestabilnosti na slici 11.1-6, a maksimalne apsolutne vrijednosti na slici 11.1-4, na kojoj se kao nestabilni okviri prepoznaju oni kojima je apsolutna vrijednost veća od jedinice.

1.d)

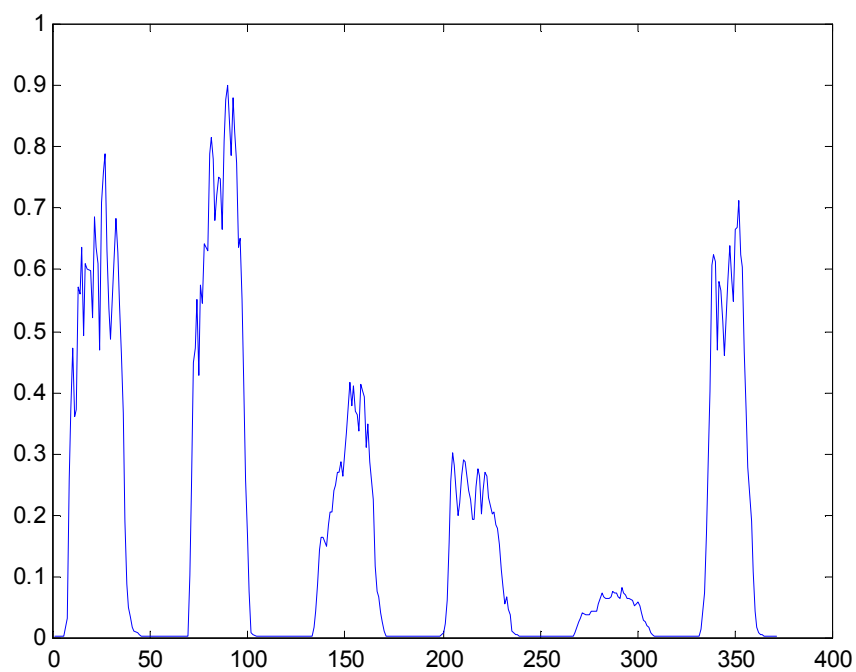
Postotak nestabilnih okvira u ovom primjeru (za zvučni signal “drago1.wav”) je 0,28%

2.a)

Za primjer signala “drago1.wav” vidjeli smo rezultate u teorijskom djelu laboratorijske vježbe na slikama 11.2-1 i 12.2-2. U zadatku pogledajmo jedan drugi primjer (npr. za signal “drago2.wav”) da bi mogli približno procijeniti za koje glasove je dovoljan koji red predikcije za zadovoljavajuću točnost (slika 11.3-1). Vidimo da je kod glas “o” već kod reda predikcije $p=5$ zadovoljavajuća točnost (dalje se pogreška više ne mijenja puno) a kod glasa “u” već kod reda $p=2$ ili $p=3$. Za razliku od toga kod glasa “a” taj red je tek negdje oko 8 ili 9 dok je najgori glas “i” koji zahtjeva maksimalan red predikcije tj. $p=10$.



Slika 11.3-1 Utjecaj reda pred. na
rezidualnu energiju



Slika 11.3-2 Rezidualna energija
za najviši red pred.

KRAJ

12. UTJECAJ KVANTIZACIJE KOEFICIJENATA LPC FILTRA

12.1 Kvantizacija koeficijenata LPC filtra kod sustava za kodiranje govornog signala

Kod sustava koji služe za prijenos ili pohranu govora, koji su temeljeni na parametarskom modelu govornog signala, potrebno je na prijemnu stranu poslati koeficijente LPC filtra. Obzirom da se prijenos obično provodi kroz digitalni komunikacijski kanal ograničene propusnosti (brzine prijenosa), poželjno je ostvariti čim veći stupanj sažimanja (kompresije). To znači da je cilj ostvariti što veću kvalitetu rekonstruiranog govornog signala uz utrošak što manjeg broja bita prilikom prijenosa ili pohrane. Koeficijenti LPC filtra su u Matlab-u izračunati u aritmetici pomičnog zareza dvostruke preciznosti (engl. double precision floating point number). Zapis jednog realnog broja u toj preciznosti zahtijeva 64 bita (8 By). Uz tipični red predikcije od $p=10$ koji se koristi kada je frekvencija otipkavanja $f_s=8\text{kHz}$, pohrana 10 a-koeficijenata LPC filtra zahtijeva 10×64 bita. Obzirom da je parametre LPC filtra potrebno prenijeti na prijemnu stranu tipično 50 puta u sekundi (zbog nestacionarnosti spektralnih svojstava govornog signala), potrebna brzina prijenosa iznosi $50 \times 10 \times 64 = 32000$ bita u sekundi (bps). Ovo je samo dva puta manje nego da smo sve vremenske uzorke govornog signala kvantizirali sa 8 bita ($8000 \times 8 = 64000$ bps), a parametri LPC filtra su samo jedan dio modela. Naime, na prijemnu stranu je potrebno poslati i parametre pobudnog signala, npr. informaciju i zvučnosti, osnovnu frekvenciju titranja glasnica f_0 , amplitudu pobudnog signala, itd.. Iz ove diskusije je očito da je potrebno provesti učinkovitije kodiranje parametara LPC filtra, tj. možda takva preciznost koja se ostvaruje s aritmetikom pomičnog zareza uopće nije potrebna. Postupak smanjenja preciznosti se naziva kvantizacijom, tj. umjesto realnog broja beskonačne (ili vrlo velike) preciznosti, koristi se tzv. kvantizirana vrijednost koja se bira iz konačnog skupa vrijednosti.

12.1.1 Kvantizacija realnih brojeva beskonačne preciznosti

Ovaj postupak je najlakše opisati na primjeru. Ako pretpostavimo da se vrijednost realnog broja x kojeg je potrebno kvantizirati nalazi u intervalu $x \in [0, 1)$, te ako odaberemo kvantizaciju sa tri bita (osam razina), tada kvantizirana vrijednost \hat{x} realnog broja x poprima vrijednosti iz konačnog skupa vrijednosti Q , tj. $\hat{x} \in Q = \{0.000 \ 0.125 \ 0.250 \ 0.375 \ 0.500 \ 0.625 \ 0.750 \ 0.875\}$. Ovakav odabir vrijednosti skupa Q odgovara kvantizaciji koja se provodi odsijecanjem prema dolje. To znači da se cijeli interval realnih brojeva $x \in [0, 0.125)$ zamjenjuje s prvom vrijednosti iz konačnog skupa Q , tj. sa $\hat{x} = 0.000$. Svim realnim brojevima u tom intervalu pridjeljuje se kod 000 binarno. Ako je $x \in [0.125, 0.250)$, tada kvantizirana vrijednost iznosi $\hat{x} = 0.125$ i ima kod 001, itd. Na ovaj način je na prijemnu stranu potrebno poslati samo indeks (kod) odabrane kvantizirane vrijednosti 0 do 7. Obzirom da takva ista tablica Q postoji i na strani prijemnika, postupak dekodiranja se svodi na čitanje vrijednosti iz tablice Q sa indeksa koji je određen poslanim kodom.

Opisanu operaciju kvantizacije s b -bita moguće je simulirati u Matlab-u slijedećom naredbom:

```
xk=floor(x*(2^b))/(2^b)
```

Prilikom odsijecanja prema dolje pogreška kvantizacije $\hat{x} - x$ nikad nije pozitivna, tj. kvantizirana vrijednost \hat{x} uvijek je manja ili jednaka od x . Maksimalna vrijednost pogreške

iznosi $-1/(2^b)$. Ako se kvantizacija provodi zaokruživanjem, a ne odsijecanjem, tada konačni skup vrijednosti Q ima slijedeće elemente $\hat{x} \in Q = \{0.0625 \ 0.1875 \ 0.3125 \ 0.4375 \ 0.5625 \ 0.6875 \ 0.8125 \ 0.9375\}$.

12.1.2 Zahtjevi na sustav za kvantizaciju LPC koeficijenata

Postoje dva glavna zahtjeva koji se postavljaju na sustav za kvantizaciju koeficijenata LPC filtra.

Ako se prilikom sinteze govornog signala na prijemnoj strani koristi LPC filter s kvantiziranim koeficijentima umjesto filtra s originalnim koeficijentima, tada slušna degradacija rekonstruiranog signala mora biti čim je moguće manja. U najboljem slučaju to izobličenje mora biti nečujno, tj. da nije moguće slušanjem razlučiti ta dva signala; prvi nastao pobuđivanjem idealnog LPC filtra, a drugi kao odziv kvantiziranog LPC filtra na istu pobudu. Frekvencijske karakteristike ta dva filtra ne moraju nužno biti identične, jer ako su te razlike dovoljno male, tada ljudsko uho ne može osjetiti razliku. Minimalna potrebna točnost kvantizacije koeficijenta (odnosno broj bita koji se pridjeljuje pojedinom koeficijentu) se određuje upravo na taj način, tj. broj bita se povećava tako dugo dok više ne postoji slušna razlika između odziva originalnog i kvantiziranog filtra.

Drugi zahtjev je osiguranje stabilnosti LPC filtra. To znači na uslijed promjene koeficijenata LPC filtra kvantizacijom neki od polova prijenosne funkcije $H(z)$ mogu izaći izvan jedinične kružnice. Uz takve polove prilikom sinteze može doći do eksponencijalnog raspirivanja, koje će sigurno uzrokovati veliku slušnu degradaciju signala.

Zbog toga je prilikom kvantizacije potrebno osigurati da LPC filter ostane stabilan. To je moguće provesti na dva načina. U prvom slučaju se kvantizacija provodi bez obzira na ovaj zahtjev, a zatim se provodi provjera stabilnosti i modifikacija prijenosne funkcije kojom se svi polovi ponovno uvlače unutar jedinične kružnice. U drugom slučaju se kvantizacija provodi na skupu parametara koji su dobiveni određenom 1-na-1 transformacijom a -koeficijenata filtra, za koji je provjeru stabilnosti moguće provesti direktno na osnovu vrijednosti tih transformiranih koeficijenata, tj. istovremeno sa samim postupkom kvantizacijom. Zavisno o odabranom transformiranom skupu parametara, postoje određena pravila koje ti parametri moraju zadovoljavati, čime će automatski biti zagarantirana i stabilnost LPC filtra. Primjer takvih transformiranih parametara su k -koeficijenti (koji se nazivaju i PARCOR koeficijenti), LAR i ARSIN parametri, kao i skup frekvencija spektralnih linija (engl. line spectrum frequencies).

12.2 Kvantizacija a -koeficijenata LPC filtra

Postupak kvantizacije koeficijenata LPC filtra biti će ilustriran u okviru ove vježbe na primjeru direktne kvantizacije a -koeficijenata, odnosno kvantizacije k -koeficijenata i LAR-koeficijenata. U ovom poglavlju biti će provedena ilustracija takvog postupka za slučaj a -koeficijenata, dok će u praktičnom dijelu vježbe to trebati ponoviti i za preostale reprezentacije LPC koeficijenata.

Prvi korak je učitavanje MAT datoteke koja je izlaz programa za LPC analizu u kojoj su pohranjeni a -koeficijenti filtra za sve okvire analize obrađenog govornog signala. Ti se koeficijenti nalaze u matrici a_m sa bf redaka i $p+1$ stupac, gdje je bf ukupni broj okvira, a p red predikcije.

```
load samo
[bf,p]=size(a_m);
p=p-1;
```

Obzirom da je prvi koeficijent a_0 polinoma $A(z)$ u nazivniku funkcije $H(z)$ uvijek jednak jedinici, njega nije potrebno prenašati na prijemnu stranu, pa ga nije ni potrebno kvantizirati. Radi toga iz matrice a_m privremeno izbacujemo prvi stupac:

```
a_m=a_m(:,2:p+1);
```

Kvantizaciju koeficijenata pohranjenih u ovoj matrici provesti ćemo postupkom skalarne kvantizacije, kod koje se kvantizacija svakog pojedinog koeficijenta provodi nezavisnim jedno-dimenzionalnim kvantizatorom koji je prilagođen statistici upravo tog koeficijenta. To znači da ako se npr. vrijednosti koeficijenta a_3 kreću u rasponu od -3 do 1.5 , tada je kvantizacijske vrijednosti \hat{x} u skupu Q potrebno postaviti upravo u taj interval (a ne šire), da bi se što bolje iskoristio broj bita koji je na raspolaganju za taj koeficijent. Navedeni raspon je različit za svaki od LPC koeficijenata i ovisi o njegovoj funkciji gustoće vjerojatnosti (engl. probability density function). U ovoj vježbi koristiti će se najjednostavniji način kvantizacije kod kojeg su kvantizacijske vrijednosti \hat{x} u skupu Q jednoliko razmaknute u intervalu koji je definiran minimalnom i maksimalnom vrijednosti varijable x koja se kvantizira. Takav tip kvantizatora se naziva jednoliki ili uniformni kvantizator. Prvi korak u postupku kvantizacije je normalizacija matrice a -koeficijenata, tj. svaki od koeficijenata a_1 do a_p biti će linearno transformiran u koeficijente a_1' do a_p' čiji je raspon od 0 do 1.

```
a_m_min=min(a_m);    % p-dimenzionalni redak s najmanjom vrijednosti svakog
                      % a-koeficijenta
a_m_max=max(a_m);    % p-dimenzionalni redak s najvećom vrijednosti svakog
                      % a-koeficijenta
a_m_raspon=a_m_max-a_m_min; % raspon svakog koeficijenta

% provedi normalizaciju
a_m_norm=(a_m-(ones(bf,1)*a_m_min))./(ones(bf,1)*(a_m_raspon*(1+eps)));
```

U gornjim naredbama normalizacije korišteni su reci sa informacijom o rasponima a -koeficijenata. Ovi podaci predstavljaju parametre kvantizatora i moraju biti poznati na obje strane (predajnika ili kodera i prijemnika odnosno dekodera). U ovom slučaju ti su parametri određeni na osnovu stvarne statistike za učitane MAT odnosno pripadnu WAV datoteku sa govornim signalom. Kod stvarnih kodera, ti parametri moraju biti određeni unaprijed, na osnovu velike govorne baze snimljene s velikim brojem govornika (muški, ženski, dječji) i to u brojnim akustičkim uvjetima (tip mikrofona, buka okoline, itd.). Konstanta ϵ u gornjem izrazu je vrlo mali broj koji osigurava da svi elementi a_m_norm budu strogo manji od 1. Sada provedi kvantizaciju svih koeficijenata sa b bita, množenjem sa 2^b i odsijecanjem prema dolje:

```
b=4;
a_m_cijel=floor(a_m_norm*2^b);
```

U varijabli a_m_cijel nalaze se cijeli brojevi u rasponu od 0 do 2^b-1 , koji predstavljaju indekse (kдове) koji se šalju na prijemnu stranu. Postupak dekodiranja se provodi na slijedeći način:

```
a_m_norm_dek=(a_m_cijel+0.5)/(2^b);
```

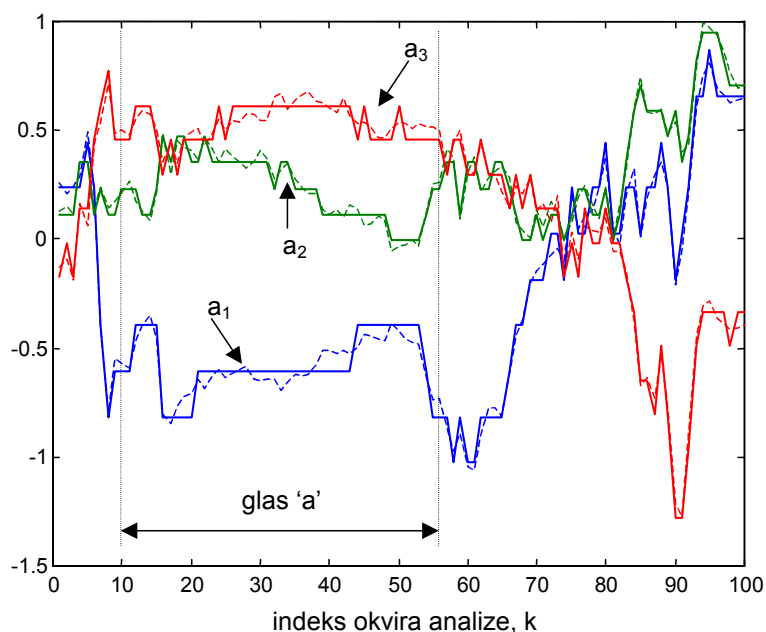
Pribrajanje konstante 0.5 simulira operaciju zaokruživanja. Tako npr., svi elementi matrice a_m_norm u intervalu $x \in [0, 1/(2^b))$ dobivaju kod 0, ali prilikom rekonstrukcije pridjeljuje im se vrijednost $\hat{x}=0.5/(2^b)$ koja se nalazi točno na sredini tog intervala. Na ovaj

način je maksimalna pogreška kvantizacije jednaka $\pm 0.5/(2^b)$, dok bi u slučaju odsijecanja bila od $-1/(2^b)$ do 0. U zadnjem koraku potrebno je dekodiranje normalizirane **a**-koeficijente denormalizirati nazad na originale raspone:

```
a_m_dek=a_m_norm_dek.*(ones(bf,1)*a_m_raspon)+(ones(bf,1)*a_m_min);
```

Rezultate ovog postupka možemo usporediti s originalnim koeficijentima. Tako slijedećom naredbom prikazujemo prva tri koeficijenta za prvih 100 okvira analize. Originalni koeficijenti prikazani su crtkanom linijom dok su kvantizirani prikazani punom linijom. Koeficijent **a**₁ prikazan je plavom bojom, **a**₂ zelenom i **a**₃ crvenom.

```
plot([1:100],a_m(1:100,1),'b--',[1:100],a_m_dek(1:100,1),'b',...
     [1:100],a_m(1:100,2),'g--',[1:100],a_m_dek(1:100,2),'g',...
     [1:100],a_m(1:100,3),'r--',[1:100],a_m_dek(1:100,3),'r');
```



Slika 12.2-1 Prva tri **a**-koeficijenta LPC filtra prilikom izgovora glasa **a**; (originalni crtkanom linijom, kvantizirani punom linijom)

Iz slike 12.2-1 je vidljivo da kvantizirane vrijednosti koeficijenata prate originalne i da je pogreška ponekad pozitivna, ponekad negativna.

Utjecaj kvantizacije na promjenu frekvencijske karakteristike filtra možemo promotriti za proizvoljno odabrani okvir analize **okv**:

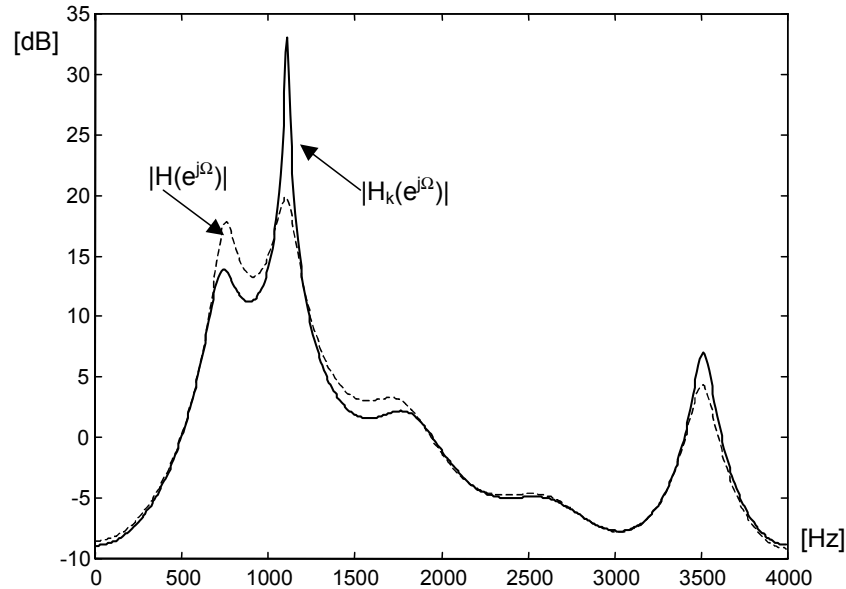
```
okv=30; % Odaberi jedan okvir
a=[1 a_m(okv,:)]; % Izdvoji originalne koeficijente tog okvira
ak=[1 a_m_dek(okv,:)]; % Izdvoji kvantizirane koeficijente tog okvira
```

Izračunajmo amplitudno frekvencijske karakteristike tih filtara u 512 jednoliko razmaknutih frekvencijskih točaka od 0 do π (tj. od 0 do $f_s/2$):

```
h=20*log10(abs(freqz(1,a,512))); % Originalna frekv. karak. u dB
hk=20*log10(abs(freqz(1,ak,512))); % Kvantizirana frekv. karak. u dB
```

Prikaz ove dvije amplitudne karakteristike (originalna crtkanom linijom, kvantizirana punom linijom) ostvarujemo slijedećom naredbom:

```
plot([0:511]/512*fs/2,h,'b--',[0:511]/512*fs/2,hk,'b');
```



Slika 12.2-2 Amplitudno frekventijske karakteristike originalnog (crtkano) i kvantiziranog (puna linija) filtra za 30-ti okvir analize (sredina izgovora glasa 'a')

Iz slike 12.2-2 je vidljivo da je najznačajnija promjena frekventijske karakteristike na mjestu drugog formanta. Uslijed kvantizacije došlo je do značajnog suženja širine drugog formanta, što se manifestiralo kao pojava izraženog šiljka puno užega nego u originalnog karakteristici.

12.2.1 Mjera kvalitete kvantizacije

Kao mjera kvalitete postuka kvantizacije LPC koeficijenata vrlo često se koristi mjera srednjeg kvadratnog spektralnog izobličenja SD^2 , definirana izrazom:

$$SD^2 = \frac{1}{\pi} \int_0^{\pi} \left[20 \log_{10} \left(\frac{1}{|A(e^{j\Omega})|} \right) - 20 \log_{10} \left(\frac{1}{|\hat{A}(e^{j\Omega})|} \right) \right]^2 d\Omega \quad (12.2-1)$$

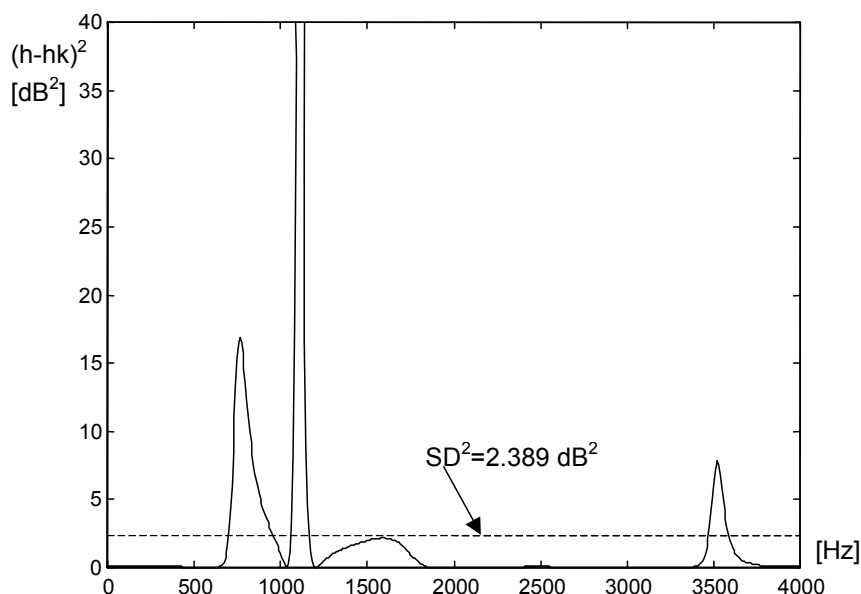
Iz izraza (12.2-1) je vidljivo da se izobličenje SD^2 određuje kao srednja kvadratna vrijednost razlike između amplitudno frekventijske karakteristike originalnog i kvantiziranog filtra (usrednjavanje se provodi preko frekventijske osi).

Ovo izobličenje se može odrediti i numerički u Matlab-u na osnovu otipkanih frekventijskih karakteristika h i h_k , zamjenom integrala sa sumom, tj. pomoću naredbe:

```
SD2=mean((h-hk).^2)
```

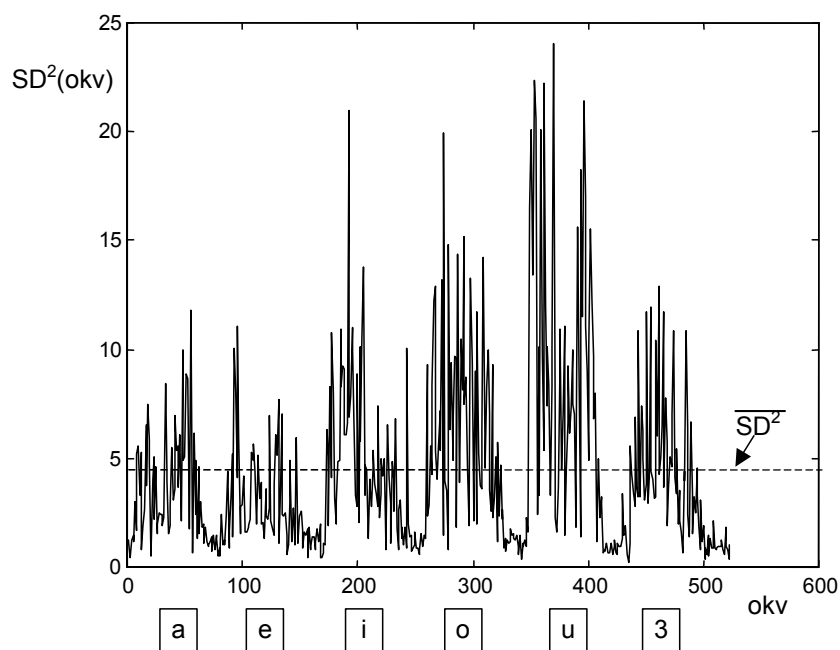
Naravno da je to samo aproksimacija, koja je točnija što su frekventijske karakteristike gušće otipkane. Na slici 12.2-3 prikazan je kvadrat razlike frekventijskih karakteristika h i h_k

koje su već prikazane na slici 12.2-2. Ta istoj slici je crtkanom linijom ucrtana vrijednost srednjeg kvadratnog spektralnog izobličenja SD^2 . Radi boljeg prikaza, ordinata na slici 12.2-3 je odrezana samo do 40 dB^2 , dok međutim kvadratna pogreška na mjestu drugog formanta seže čak do 176 dB^2 .



Slika 12.2-3 Kvadrat razlike amplitudno frekvencijskih karakteristika i srednjeg kvadratnog spektralnog izobličenja SD^2 (crtkano) za 30-ti okvir analize (sredina izgovora glasa 'a')

Opisani postupak određivanja srednjeg kvadratnog spektralnog izobličenja može se provesti za sve okvire analize pohranjenje u matricama a_m , odnosno a_{m_dek} , variranjem okv od 1 do bf.



Slika 12.2-4 Srednje kvadratno izobličenje SD^2 za svih 6 samoglasnika i prosječna vrijednost srednjeg kvadratnog izobličenja (crtkano)

Na slici 12.2-4 prikazan je iznos spektralne pogreške $SD^2(okv)$, za sve LPC okvire u datoteci **samo.mat**. Na slici su ucrtane i približne pozicije svakog od šest samoglasnika. Karakteristično je da je spektralna pogreška najveća za glasove 'o' i 'u', a razlog je u činjenici što ti glasovi imaju vrlo izražene (uske) formante, pa kvantizacija značajno mijenja karakteristiku filtra.

Da bi se dobila neka prosječna mjera kvalitete takvog kodiranja, pogodno je usrednjiti $SD^2(okv)$ preko svih okvira analize, čime se dobiva prosječna vrijednost srednjeg kvadratnog izobličenja $\overline{SD^2}$, koja je crtkanom linijom ucrtana na slici 12.2-4, (napomena: sada se usrednjavanje provodi u vremenskoj domeni).

Istraživanja su pokazala da prag čujnosti spektralnog izobličenja iznosi $\overline{SD^2}=1\text{dB}^2$, što drugim riječima znači da slušanjem nije moguće razlučiti odziv originalnog i kvantiziranog filtra, ako je u prosjeku iznos $SD^2(okv)$ manji od 1dB^2 .

Zadatak 1.:

- Napiši Matlab program za određivanje $SD^2(okv)$,
- provjeri rezultate usporedbom sa slikom 12.2-4,
- Odredi točni iznos $\overline{SD^2} = \underline{\hspace{2cm}}$
- Odredi ovisnost $\overline{SD^2}$ o broju bita b , te je prikaži tabelarno (gafički) za b od 3 do 7
- Pri kojem broju bita se ostvaruje uvjet na nečujnost izobličenja $b_{\min} = \underline{\hspace{2cm}}$

12.2.2 Utjecaj kvantizacije na položaj polova

Utjecaj kvantizacije možemo razmotriti i kao promjenu pozicije polova prijenosne funkcije $H(z)$. Korijene polinoma $A(z)$ u nazivniku prijenosne funkcije $H(z)$ nalazimo funkcijom `roots`:

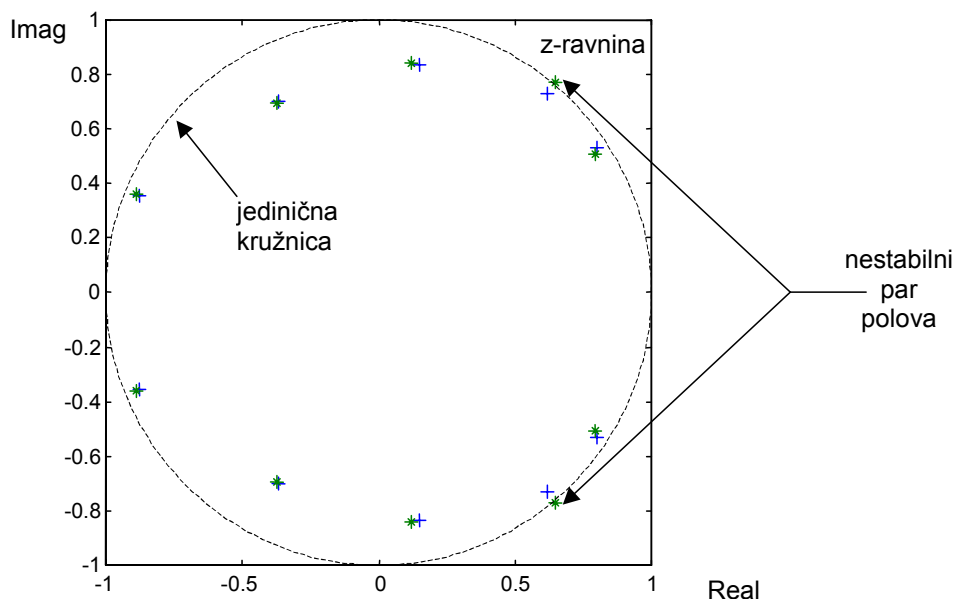
```
r=roots(a);
rk=roots(ak);
plot(real(r),imag(r),'+',real(rk),imag(rk),'*',...
      sin(2*pi*[0:200]/200), cos(2*pi*[0:200]/200)); axis('square');
```

Na slici 12.2-5 prikazana je promjena pozicije polova LPC filtra uzrokovana kvantizacijom (za $okv=30$ i $b=4$). Iz slike je vidljivo da je jedan par konjugirano kompleksnih polova (koji odgovara drugom formantu) čak izašao izvan jedinične kružnice. To je i razlog velike pogreške tog formanta, koja je vidljiva i na slikama 12.2-2 i 12.2-3.

Da bi mogli numerički odrediti utjecaj kvantizacije na poziciju polova, potrebno je odrediti parametre polova (centralne frekvencije i širine formanata). Obzirom da su koeficijenti prijenosne funkcije $H(z)$ realni brojevi, svi se polovi pojavljuju u konjugirano kompleksnim parovima. Prijenosna funkcija je potpuno opisana s polovima u gornjoj poluravnini, tj. sa onima koji imaju pozitivne kutove (0 do π). Izdvajanje samo tih korijena, te određivanje njihovog modula (radijusa) i kuta obavlja se slijedećim naredbama:

```
% Sacuvaj samo korijene u gornjoj poluravnini (kut>0)
r=r(find(angle(r)>0)); % za originale korijene
rk=rk(find(angle(rk)>0)); % za korijene kvantiziranog H(z)

r_mod=abs(r); r_ang=angle(r); % Odredi modul i kut svih korijena
rk_mod=abs(rk); rk_ang=angle(rk);
```

Slika 12.2-5 Položaj polova originalnog (+) i kvantiziranog (*) filtra za 30-ti okvir analize (sredina izgovora glasa 'a')

Obzirom da funkcija **roots** vraća korijene u proizvoljnom rasporedu, potrebno je izvršiti sortiranje po rastućim centralnim frekvencijama. Na osnovi sortiranog niza indeksa (**inr**, odnosno **inrk**), preslaguju se i moduli, da bi imali isti raspored kao i kutovi korijena.

```
% Sortiraj korjene po rastucim kutevima
[sr_ang, inr]=sort(r_ang); sr_mod=r_mod(inr);
[srk_ang, inrk]=sort(rk_ang); srk_mod=rk_mod(inrk);

% Prikazi tablicu modula i kuteva (prije i poslije kvant.)
disp ([1:5]' sr_ang sr_mod srk_ang srk_mod);
```

Tablica 12.2-1 Polovi LPC filtra prije i poslije kvantizacije **a**-koeficijenata

i	Polovi $H(z)$		Polovi $\hat{H}(z)$	
	$\angle(r_i)$	$ r_i $	$\angle(\hat{r}_i)$	$ \hat{r}_i $
1	0.5862	0.9583	0.5691	0.9437
2	0.8668	0.9582	0.8698	1.0077
3	1.3967	0.8492	1.4296	0.8500
4	2.0495	0.7898	2.0625	0.7860
5	2.7544	0.9457	2.7594	0.9598

U tablici 12.2-1 prikazani su parametri (kut i modul) polova prijenosnih funkcija $H(z)$ i $\hat{H}(z)$, koji su također i grafički prikazani na slici 12.2-5. U drugom redu tablice može se očitati da je radijus drugog pola uslijed kvantizacije porastao sa 0.9582 na 1.0077, čime je uzrokovao nestabilnost filtra. Usporedbom lijeve i desne strane vidljivo je na kvantizacija utječe i na modul i na kut pola. Obzirom da položaj polova određuje parametre formanata

koji su modelirani s LPC filtrom, pogodno je parametre polova ($\angle(r_i)$ i $|r_i|$) preračunati u parametre formanata, tj. centralnu frekvenciju f_i i širinu BW_i , kao što je opisano slijedećim izrazima:

$$f_i = \frac{f_s}{2\pi} \cdot \angle(r_i), \quad BW_i = -\frac{f_s}{\pi} \log(|r_i|) \quad (12.2-2)$$

```
% Preracunaj kuteve i module u cent. frekv. i sirine formanata
sr_f =sr_ang/pi*fs/2;    sr_bw =-log(sr_mod)/pi*fs;
srk_f=srk_ang/pi*fs/2;    srk_bw=-log(srk_mod)/pi*fs;

% Prikazi tablicu cent.frekv. i sirina formanata (prije i poslije kvant.)
disp ([1:5]' round([sr_f sr_bw srk_f srk_bw]));
```

Tablica 12.2-2 Formanti LPC filtra prije i poslije kvantizacije a-koeficijenata

i	Formanti $H(z)$		Formanti $\hat{H}(z)$	
	cent.f. [Hz]	BW, [Hz]	cent.f. [Hz]	BW, [Hz]
1	746	109	725	147
2	1104	109	1107	-20
3	1778	416	1820	414
4	2609	601	2626	613
5	3507	142	3513	105

Iz tablice je 12.2-2 vidi se da je najveća promjena upravo u širini drugog formanta koji je sa 109 Hz sužen na -20 Hz. Ova druga brojka nema nikakvog fizikalnog smisla, osim što obzirom na negativni predznak pokazuje da je filter nestabilan. Najveća promjena centralne frekvencije je za 3 formant (pomak za 42Hz u desno), no obzirom da taj formant ima relativno veliku širinu, ta se greška i nije značajno manifestirala na slikama 12.2-2 i 12.2-3.

U stvarnim primjenama dovoljno je ustanoviti da li kvantizacija uzrokuje nestabilnost ili ne. Broj nestabilnih konjugirano kompleksnih parova polova u vektoru rk nalazimo slijedećom naredbom:

```
unst=length(find(abs(rk)>1))/2;
```

Zadatak 2.

- Opisani postupak određivanja broja nestabilnih konjugirano kompleksnih parova polova treba provesti za sve okvire analize, $okv=1$ do bf , tako da se u stupac $unst$ sa bf redaka upiše broj nestabilnih formanata svakog pojedinog okvira. Napiši pripadni program u Matlab-u
- Izvrši ga i grafički prikaži rezultate (uz $b=4$).
- Za koje glasove je broj nestabilnih okvira najveći, a za koji najmanji?
- Potrebno je ustanoviti postotak nestabilnih LPC okvira prema izrazu:


```
post_unst=mean(unst>0)*100;
```
- Odrediti ovisnost $post_unst$ o broju bita b , te je prikazati grafički i tabelarno za b od 3 do 7
- Koliko ima nestabilnih okvira uz $b=7$, koji su to okviri i kojem glasu pripadaju?

12.3 Kvantizacija k-koeficijenata LPC filtra

Druga mogućnost je kvantizacija k-koeficijenata ili tzv. PARCOR koeficijenata, koji se mogu odrediti iz a-koeficijenata LPC filtra. Ova konverzija čak nije ni potrebna ako se primjenjuje autokorelacijski LPC postupak s Levinson-Durbin-ovom rekurzijom za određivanje prediktora, jer se tada k-koeficijenti određuju automatski u postupku izračunavanja a-koeficijenata. Kvantizacija se provodi na k-koeficijentima koji će za stabilne LPC filtre $H(z)$ uvijek biti u intervalu $k_i \in (-1, 1)$, za $i=1,2,\dots,p$.

U ovom slučaju parametre kvantizatora je mnogo lakše izračunati, jer je raspon svih ulaznih varijabli uvijek jednak $(-1, 1)$. Ako bi ovim koeficijentima pribrojili jedan i zatim sve skupa podijelili sa dva, dobiva se željen raspon skalirane ulazne realne varijable $(0, 1)$. Nakon provedene kvantizacije provodi se obrnut postupak, te se zatim tako dobiveni kvantizirani k-koeficijenti nazad transformiraju u a-koeficijente. Ako je odabirom skupa kvantizacijskih vrijednosti Q osigurano da i nakon kvantizacije svi k-koeficijenti ostanu u intervalu $(-1, 1)$, tada će i pripadni LPC filter sigurno biti stabilan.

Ako je polinom $A(z)$ u nazivniku prijenosne funkcije $H(z)$ definiran a-koeficijentima a_i , prema izrazu:

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_p z^{-p} \quad (12.3-1)$$

tada se koeficijenti a_i mogu se dobiti iz PARCOR koeficijenata k_i korištenjem rekurzije:

$$a_i^{(i)} = -k_i \quad (12.3-2)$$

$$a_j^{(i)} = a_j^{(i-1)} - k_i a_{i-j}^{(i-1)} \quad 1 \leq j \leq i-1 \quad (12.3-3)$$

Izrazi (12.3-2) i (12.3-3) izračunavaju za $i=1, \dots, p$, a koeficijenti polinoma $A(z)$ slijede iz zadnjeg koraka rekurzije prema izrazu:

$$a_j = a_j^{(p)} \quad 1 \leq j \leq p \quad (12.3-4)$$

Slično tome, skup PARCOR koeficijenata k_i može se dobiti iz skupa a-koeficijenata koristeći obrnutu rekurziju oblika:

$$k_i = -a_i^{(i)} \quad (12.3-5)$$

$$a_j^{(i-1)} = \frac{a_j^{(i)} - a_i^{(i)} a_{i-j}^{(i)}}{1 - k_i^2} \quad 1 \leq j \leq i-1 \quad (12.3-6)$$

gdje i ide od $p, p-1$, pa sve do 1 , a u početku se postavi

$$a_j^{(p)} = a_j \quad 1 \leq j \leq p \quad (12.3-7)$$

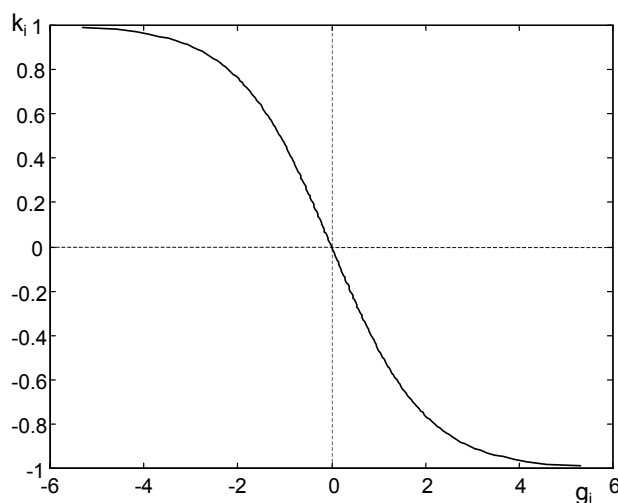
Funkcije koje provodi opisane konverzije već postoje i zovu se `a2k_new.m`, odnosno `k2a_new.m`. Te funkcije vrše pretvorbu jednog retka iz a u k koeficijente, odnosno obratno. Način pozivanja ovih funkcija proučite pozivom `help a2k_new`, odnosno `help k2a_new`.

Zadatak 3. Potrebno je ponoviti postupak kvantizacije koji je opisan u poglavlju 12.2, ali sada na skupu k-koeficijenata. Prvi korak je izračunavanje matrice k_m na osnovu a_m

koja je učitana iz datoteke **samo.mat** (PAZI: u funkciju **a2k_new** ne ulazi koeficijent a_0 koji se originalno nalazi u matrici **a_m**). Ponovi zadatak 1. u cijelosti, te prikaži rezultate iz stavke **1. d)** usporedno s onima za slučaj kvantizacije **a**-koeficijenata s istim brojem bita.

12.3.1 LAR (engl. Log Area Ratio) koeficijenti

Još se bolji rezultati mogu postići ako se prije kvantizacije provede još jedna nelinearna transformacija **k**-koeficijenata. Utvrđeno je da spektralna pogreška koju uzrokuje kvantizacija **k**-koeficijenata ovisi o njihovom apsolutnom iznosu, tj. opaženo je da kada je modul koeficijenta blizak jedinici, da tada i male kvantizacijske pogreške uzrokuju veliko spektralno izobličenje. Radi toga pogodno je provesti takvu transformaciju koja će prije kvantizacije 'razvući' područje u okolini -1 i 1.



Slika 12.3-1 Transformacija **k**-koeficijenata u LAR parametre

Jedna od dvije poznate transformacije tog tipa prikazana je na slici 12.3-1. Transformirana varijabla g_i se naziva logaritam omjera poprečnog presjeka (engl. log area ratio) ili kraće LAR-parametar, a definirana je slijedećim izrazom::

$$g_i = \log\left(\frac{A_{i+1}}{A_i}\right) = \log\left(\frac{1-k_i}{1+k_i}\right), \quad 1 \leq i \leq p \quad (12.3-8)$$

Parametri g_i su jednaki logaritmu omjera površina poprečnih presjeka susjednih segmenata kod modela s cijevima bez gubitaka, gdje taj model ima identičnu prijenosnu funkciju, kao i LPC model opisan parametrima k_i . Utvrđeno je da su g_i parametri vrlo pogodni za kvantizaciju, jer imaju relativno ravnu karakteristiku spektralne osjetljivosti, tj. neovisno o iznosu pojedinog parametra, spektralno izobličenje će biti proporcionalno kvadratu kvantizacijske pogreške tog parametra.

Iz g_i parametra je moguće inverznom transformacijom direktno dobiti k_i parametre:

$$k_i = \frac{1-e^{g_i}}{1+e^{g_i}}, \quad 1 \leq i \leq p \quad (12.3-9)$$

Zadatak 4) Modificiraj program iz zadatka **3)** i usporedi i ove rezultate s prethodnima.

12.4 Rješenja:

1.a)

```
for okv=1:bf,
    a=[1 a_m(okv,:)]; % Izdvoji originalne koeficijente tog okvira
    ak=[1 a_m_dek(okv,:)]; % Izdvoji kvatizirane koeficijente tog okvira

    h=20*log10(abs(freqz(1,a,512))); % Originalna frekv. karak. u dB
    hk=20*log10(abs(freqz(1,ak,512))); % Kvatizirana frekv. karak. u dB

    SD2(okv)=mean((h-hk).^2);
end;
```

1. b) vidi sliku 12.2-4

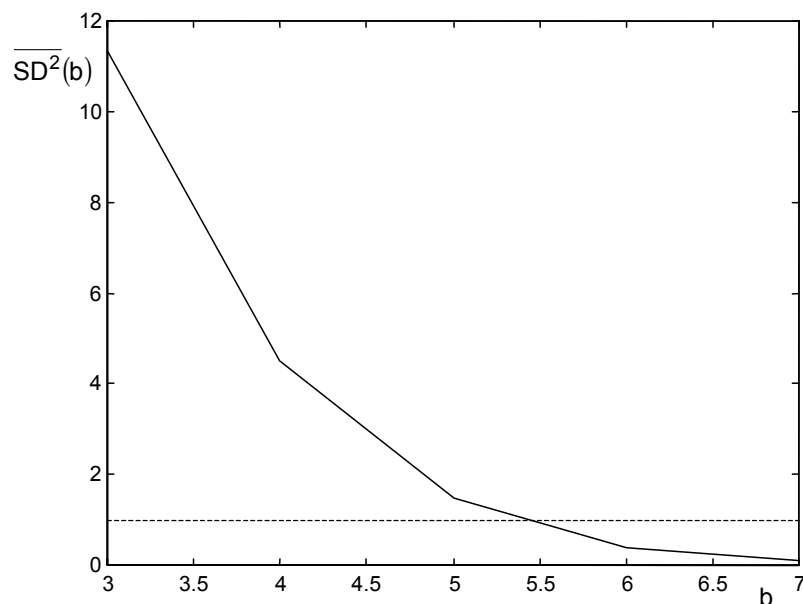
1. c) 4.5021 dB²

1. d)

Tablica 12.4-1 Ovisnost prosječne vrijednosti srednjeg kvadratnog spektralnog izobličenja o broju bita **b** za kvantizaciju

broj bita, b	3	4	5	6	7
$\overline{SD^2}$, [dB ²]	11.3143	4.5021	1.4617	0.3905	0.1013

U tablici 12.4-1 i slici 12.4-1, prikazana je ovisnost kvalitete kodera o broju bita **b** kojim se kvantizira svaki **a**-koeficijent filtra. Na slici 12.4-1 je crtkanom linijom označen i nivo od $\overline{SD^2} = 1 \text{ dB}^2$ uz koji se ostvaruje nečujnost utjecaja kvantizacije. Vidljivo je da se taj iznos spektralnog izobličenja ostvaruje uz cca 5.5 bita po koeficijentu, tj. 5 bita je malo premalo, dok je 6 bita malo previše.



Slika 12.4-1 Ovisnost prosječne vrijednosti srednjeg kvadrantog spektralnog izobličenja o broju bita **b** za kvantizaciju

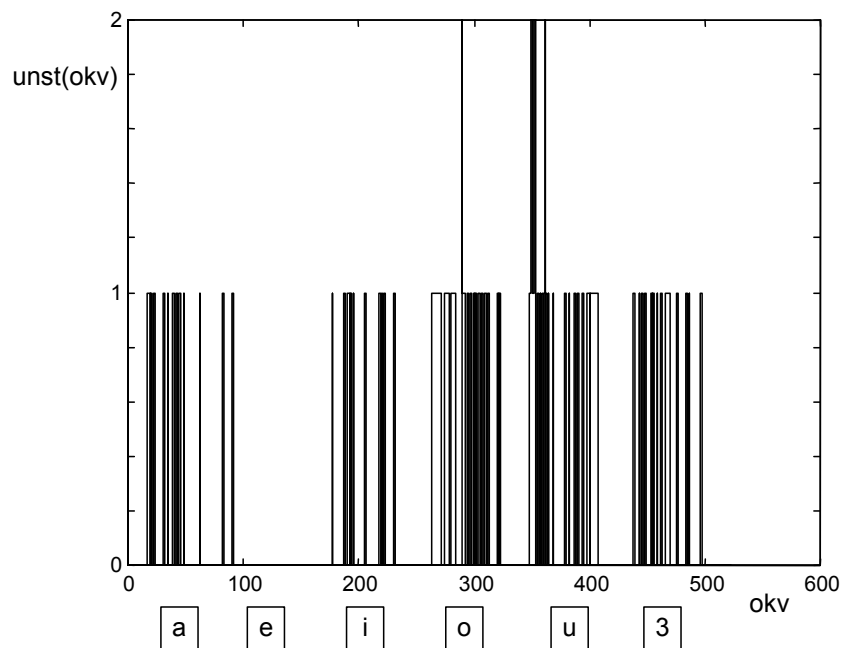
1. e) ... Odgovor je $b_{\min} = 6$ bita po koeficijentu

2.a)

```

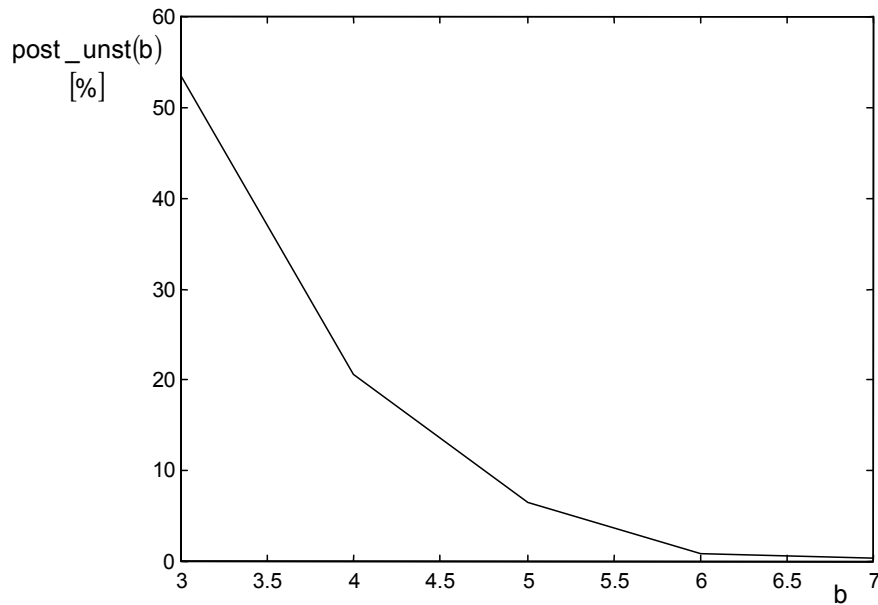
for okv=1:bf,
    ak=[1 a_m_dek(okv,:)]; % Izdvoji kvatizirane koeficijente tog okvira
    rk=roots(ak);
    unst(okv)=length(find(abs(rk)>1))/2;
end;

```

2. b) vidi sliku 12.4-2**Slika 12.4-2** Broj nestabilnih polova za sve okvire analize**2. c)** Najveći postotak nestabilnih okvira je za glasove 'o' i 'u', a najmanji za 'e'**2. d)** post_unst=20.6501 %**2. e)****Tablica 12.4-2** Ovisnost postotka nestabilnih LPC okvira o broju bita **b** za kvantizaciju **a**-koeficijenata

broj bita, b	3	4	5	6	7
post_unst, [%]	53.54	20.65	6.50	0.96	0.38

U tablici 12.4-2 i slici 12.4-3, prikazana je ovisnost postotka nestabilnih LPC okvira o broju bita **b** kojim se kvantizira svaki **a**-koeficijent filtra. Vidljivo je da se taj postotak vrlo brzo smanjuje s povećanjem **b**, ali čak i uz **b=7** još uvijek postoje dva okvira s nestabilnim filtrima i to su okviri okv=351 i okv=352 na početku glasa 'u', što je ujedno i odgovor na pitanje **2. f)**.



Slika 12.4-3 Ovisnost postotka nestabilnih LPC okvira o broju bita b za kvantizaciju a -koeficijenata

3.a) Program za kvantizaciju k -koeficijenata LPC filtra

```
clear
load samo                                % Ucitaj LPC parametre
[bf,p]=size(a_m);                        % broj okvira i red predikcije
p=p-1;

a_m=a_m(:,2:p+1);                        % Izbaci prvi stupac a0
k_m=0*a_m;                              % Matrica za pohranu k-koef.

for okv=1:bf,                            % Za sve LPC okvire
    k_m(okv,:)=a2k_new(a_m(okv,:));      % Izracunaj k-koef.
end;

k_m_norm=(k_m+1)/2;                      % Svedi na interval 0 do 1

for b=3:7,                               % Variraj broj bita od 3 do 7

    k_m_cijel=floor(k_m_norm*2^b);        % Kvantiziraj
    k_m_norm_dek=(k_m_cijel+0.5)/(2^b);   % Dekodiraj
    k_m_dek=k_m_norm_dek*2-1;            % Denormiraj

    a_m_dek=0*k_m_dek;                   % Matrica za pohranu dek. a-koef.

    for okv=1:bf,                         % Za sve okvire
        a_m_dek(okv,:)=k2a_new(k_m_dek(okv,:)); % Izracunaj a-koef.
    end;

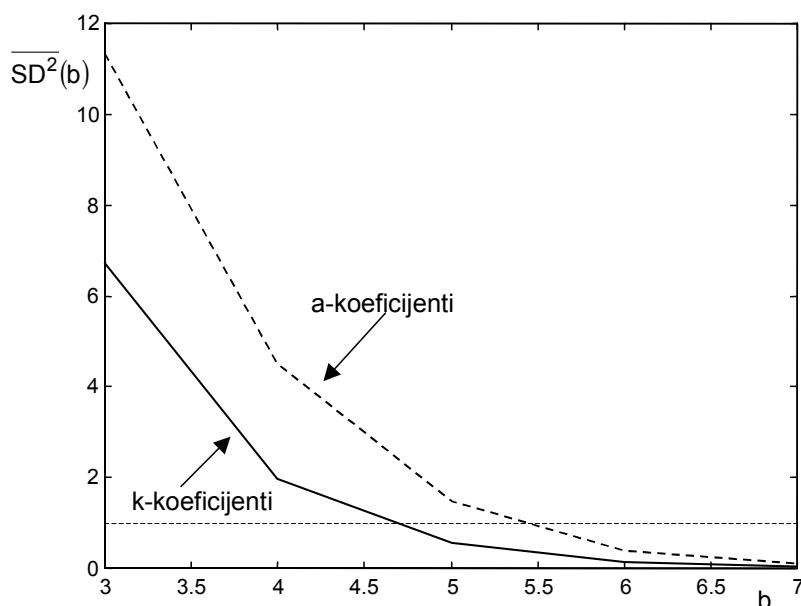
    for okv=1:bf, % Izracunja srednje kv. spek. izoblicenje za sve okvire
        h=20*log10(abs(freqz(1,[1 a_m(okv,:)]),512))); % Originalna karak.
        hk=20*log10(abs(freqz(1,[1 a_m_dek(okv,:)]),512))); % Kvatizirana kar.
        SD2(okv)=mean((h-hk).^2); % Srednje kv. spektralno izoblicenje
    end;

    SD2m(b)=mean(SD2); % Zapamti prosj. vrijed. sred. kv. spek. izoblicenja
end;
```

3.d) U tablici 12.4-3 i slici 12.4-4 usporedno su prikazani rezultati direktne kvantizacije **a**-koeficijenata i kvantizacije **k**-koeficijenata. Vidljivo je da upotreba **k**-koeficijenata štedi oko jedan bit po koeficijentu u odnosu na **a**-koeficijente, tj. prosječna vrijednost srednjeg kvadratnog izobličenja 1dB^2 postiže se već kod cca. 4.5 bita po koeficijentu (a ne 5.5).

Tablica 12.4-3 Ovisnost prosječne vrijednosti srednjeg kvadratnog spektralnog izobličenja o broju bita **b** za kvantizaciju **a** i **k** koeficijenata

broj bita, b		3	4	5	6	7
$\overline{SD^2}$, [dB ²]	a-koef	11.3143	4.5021	1.4617	0.3905	0.1013
	k-koef	6.7098	1.9770	0.5577	0.1303	0.0344

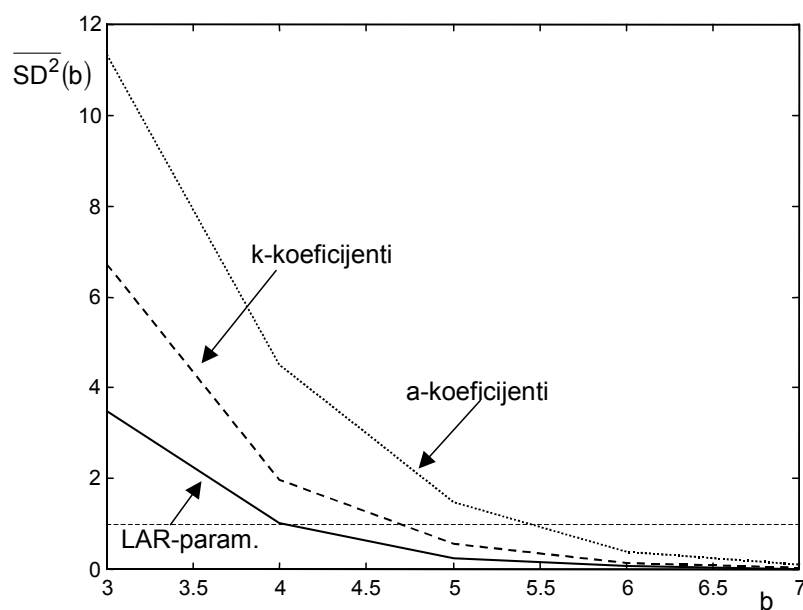


Slika 12.4-4 Ovisnost prosječne vrijednosti srednjeg kvadratnog spektralnog izobličenja o broju bita **b** za kvantizaciju **a** i **k** koeficijenata

4) U slučaju korištenja LAR parametara, uvjet na nečujnu kvantizaciju se ostvaruje već uz 4 bita po svakom LAR parametru, kao što se vidi u tablici 12.4-4 i 12.4-5, što predstavlja dodatnu uštedu od cca 0.5 bita u odnosu na **k**-koeficijente

Tablica 12.4-4 Ovisnost prosječne vrijednosti srednjeg kvadratnog spektralnog izobličenja o broju bita **b** za kvantizaciju **a** i LAR koeficijenata

broj bita, b		3	4	5	6	7
$\overline{SD^2}$, [dB ²]	a-koef	11.3143	4.5021	1.4617	0.3905	0.1013
	LAR	3.4978	1.0281	0.2485	0.0640	0.0159



Slika 12.4-5 Ovisnost prosječne vrijednosti srednjeg kvadrantog spektralnog izobličenja o broju bita b za kvantizaciju sva tri tipa LPC koeficijenata

13. HOMOMORFNA ANALIZA GOVORNOG SIGNALA

Student: Ermin Hasićević

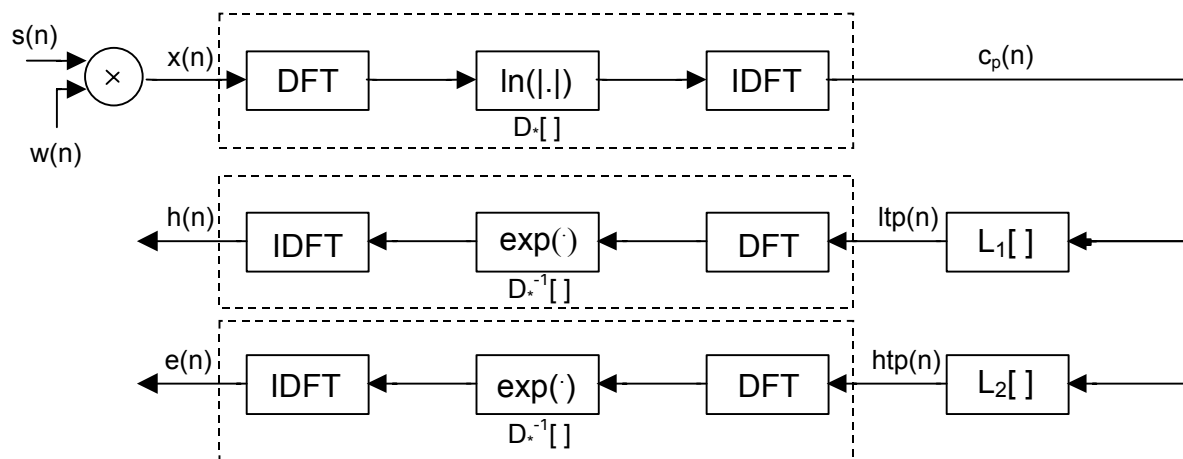
POČETAK

13.1 Uvodna razmatranja o homomorfnoj analizi

Ovim uvodnim razmatranjem ćemo zadati glavne zahtjeve koje treba udovoljiti sustav za homomorfnu analizu govornog signala te u skladu s tim zahtjevima oblikovati traženi sustav i dati njegovu blok-shemu. Sustav mora biti homomorfan s obzirom na konvoluciju. To svojstvo je dano izrazom (12.2-1), gdje operator $H[\]$ opisuje djelovanje sustava, $x(n)$ ulazni signal predstavljen konvolucijom signala $x_1(n)$ i $x_2(n)$, $y(n)$ izlazni signal predstavljen konvolucijom signala $y_1(n)$ i $y_2(n)$.

$$H[x(n)] = H[x_1(n) * x_2(n)] = H[x_1(n)] * H[x_2(n)] = y_1(n) * y_2(n) = y(n) \quad (13.1-1)$$

Sustav za homomorfnu obradu govora, koji zadovoljava postavljene uvjete sastojat će se od kaskade triju sustava homomorfnih za konvoluciju, dijela. To su: karakteristični sustav za dekonvoluciju ($D^*[\]$), linearnog operatora ($L_1[\]$, odnosno $L_2[\]$) i inverzni karakteristični sustav za dekonvoluciju ($D^{*-1}[\]$). $L_1[\]$ i $L_2[\]$ su međusobno komplementarni operatori.



Slika 13.1-1 Blok shema sustava za homomorfnu obradu govora

$s(n)$ – govorni signal

$w(n)$ – vremenski otvor

$x(n)$ – segment govornog signala izdvojen vremenskim otvorom (slika 13.2-2)

$c_p(n)$ – keprstar segmenta govornog signala (slika 13.2-4)

$ltp(n)$ – nisko-vremenski dio keprstar govornog signala (slika 13.2-5)

$hnp(n)$ – visoko-vremenski dio keprstar govornog signala (slika 13.2-6)

$h(n)$ – impulsni odziv sustava (slika 13.2-11)

$e(n)$ – pobuda sustava (slika 13.2-12)

13.2 Homomorfna analiza zvučnih glasova

U ovom poglavlju ćemo pokazati, na primjeru glasa 'a', pokazati homomorfnu analizu zvučnog glasa i rezultate pojedinih međukoraka. Zvučna datoteka iz koje ćemo uzeti uzorke za analizu biti će **sa2.wav**.

Zvučnu datoteku ćemo učitati sa:

```
[sig, fs]=wavread('sa2.wav');
```

, gdje je

sig - vektor koji sadrži uzorke govornog signala

fs – frekvencija uzorkovanja.

Broj uzoraka signala koje ćemo uzeti za analizu neka bude

$N=512$;

Frekvencijsku i vremensku os možemo pripremiti ovako:

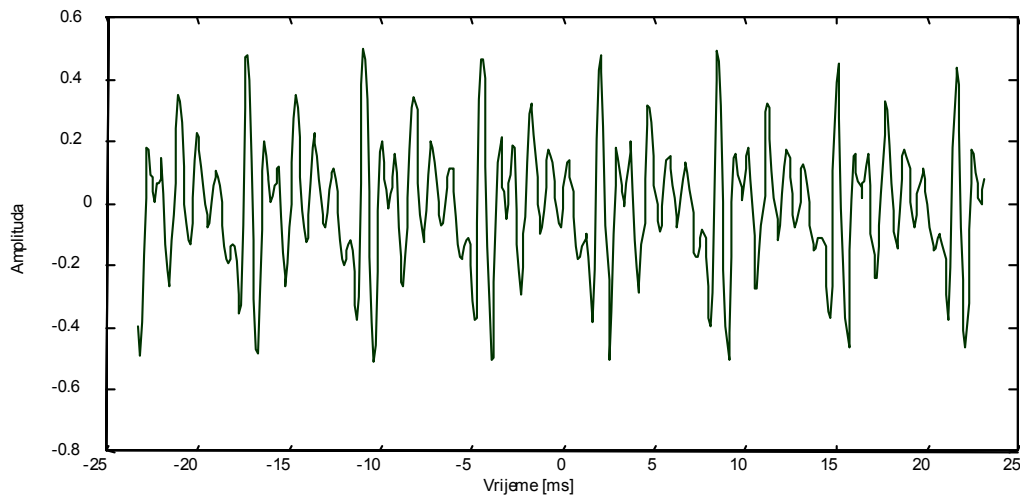
```
f = [-N/2:N/2-1]/N*fs;  
t = [-N/2:N/2-1]/fs*1000;
```

Naredbom `size(dat)` dolazimo do ukupnog broja uzoraka u signalu koji iznosi 13780. Na temelju tog podatka možemo odrediti od kojeg uzorka ćemo uzeti odsječak govornog signala. Neka to u našem primjeru bude 10000-ti uzorak. Od tog uzorka uzimamo slijedećih N uzoraka.

```
sig_part=sig(10000:10000+(N-1));
```

Izvučeni dio signala možemo prikazati sa:

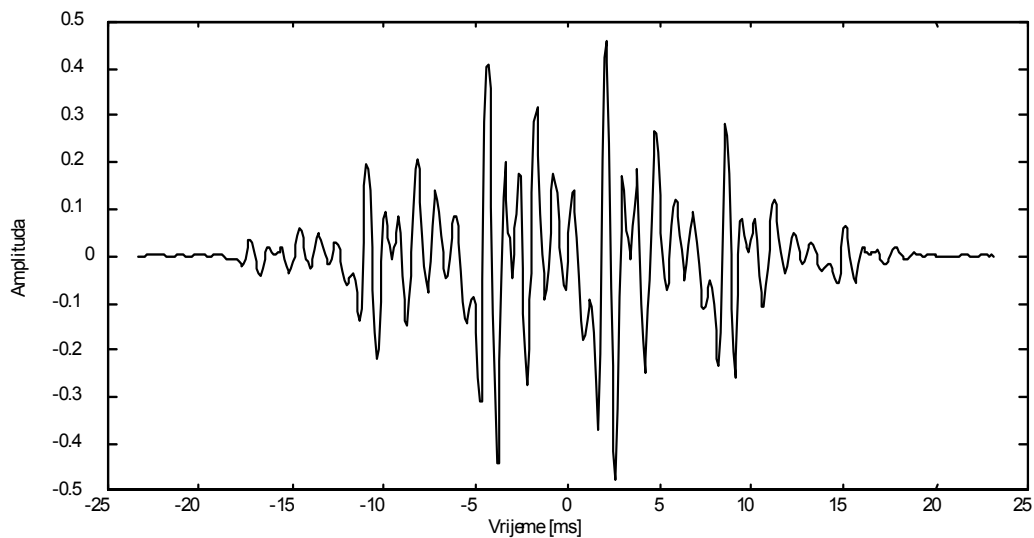
```
plot(t,sig_part);
```



Slika 13.2-1 Govorni signal

Potom izvučeni dio signala pomnožimo sa Blackman-ovim vremenskim otvorom i prikazemo ga.

```
sig_part=sig_part.*blackman(N);
plot(t,sig_part);
```



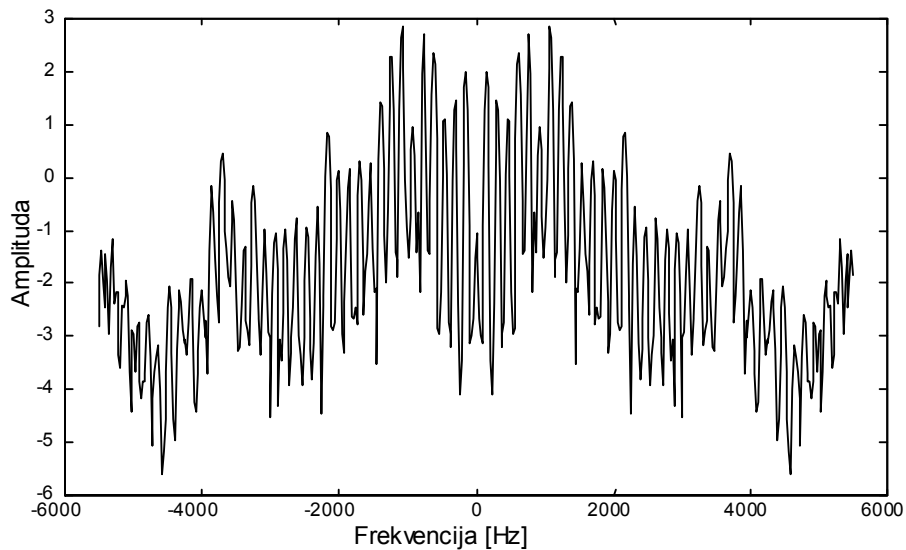
Slika 13.2-2 Segment zvučnog glasa izdvojenog vremenskim otvorom

Kao što smo i očekivali, izdvojeni dio signala, s obzirom da se radi o zvučnom glasu, je periodičan. Za potrebe homomorfne analize zvučnih glasova ćemo koristiti realni kepsstar, tj. možemo zanemariti faznu karakteristiku signala. Prvo ćemo na izdvojeni segment govornog signala primijeniti karakteristični sustav za homomorfnu konvoluciju ($D^*[]$). Dakle, slijedeći korak predstavlja nalaženje DFT-a signala u N točaka i logaritma modula spektra.

```
spect=fft(sig_part);
mod_spect=log(abs(spect));
```

Prikaz logaritma modula spektra:

```
plot(f, [mod_spect(N/2+1:N) mod_spect(1:N/2)]);
```



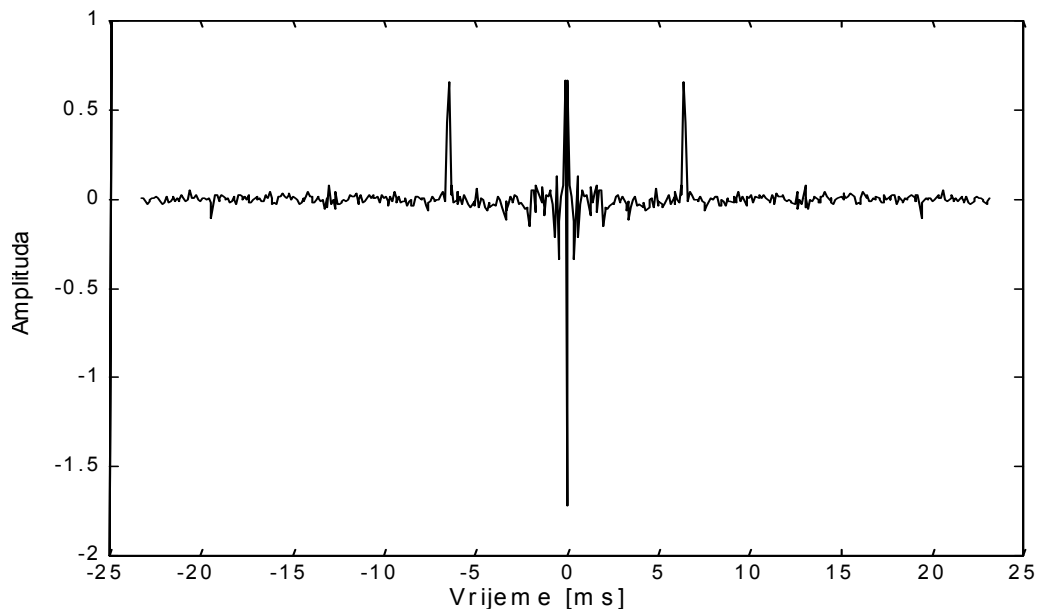
Slika 13.2-3 Prirodni logaritam modula spektra

Sada ćemo izračunati realni kepstar tako da primijenimo inverzni DFT nad prirodnim logaritmom modula spektra(mod_spect) signala koji promatramo. Sa slike 13.2-3 vidimo da je prirodni logaritam modula spektra simetričan te stoga kao rezultat inverznog DFT-a moramo dobiti realni signal. Kako to nije slučaj u MATLAB-u(zbog numeričkih pogrešaka koje unosi MATLAB) rezultat inverznog DFT-a ćemo na silu načiniti realnim.

```
cepst=ifft(mod_spect);  
cepst=real(cepst);
```

Prikaz kepstara:

```
plot(t, [cepst(N/2+1:N) cepst(1:N/2)]);
```



Slika 13.2-4 Kepstar govornog signala

U kepstaru možemo uočiti dva karakteristična dijela. To su nisko-vremenski dio(Low Time Part-LTP), koji odgovara središnjem dijelu kepstara, i visoko-vremenski dio(High Time

Part-HTP), koji odgovara rubnim dijelovima kepstra. LTP predstavlja pobudu sustava, dok HTP odgovara odzivu sustava. Mi želimo iz kepstra izdvojiti i LTP i HTP. To postizemo primjenom komplementarnih linearnih operatora $L1[]$ i $L2[]$. Pri tome ćemo uzeti određeni vremenski prag t_0 . Neka on bude 2.5 ms:

```
t0=2.5e-3;
```

Sada moramo izračunati indeks onog uzorka kepstra koji se nalazi na granici određenoj vremenskim pragom t_0 :

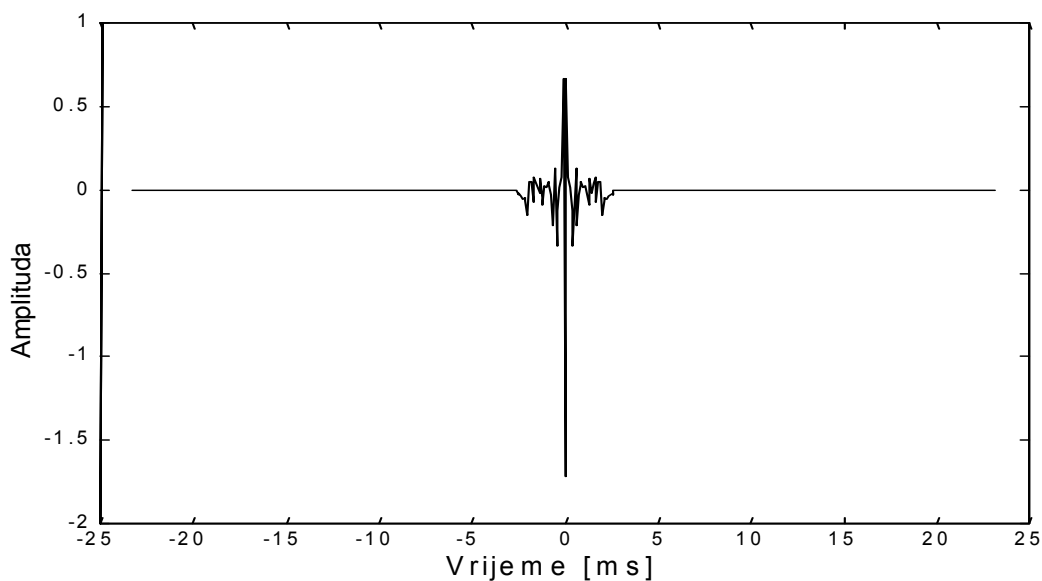
```
t0_ind=round(t0*fs);
```

Na temelju tako određenog indeksa možemo izdvojiti LTP i HTP kepstra:

```
ltp=cepst;
ltp(in_t0+2:N-in_t0)=zeros(1,N-2*in_t0-1);
htp=cepst-ltp;
```

Prikaz LTP-a:

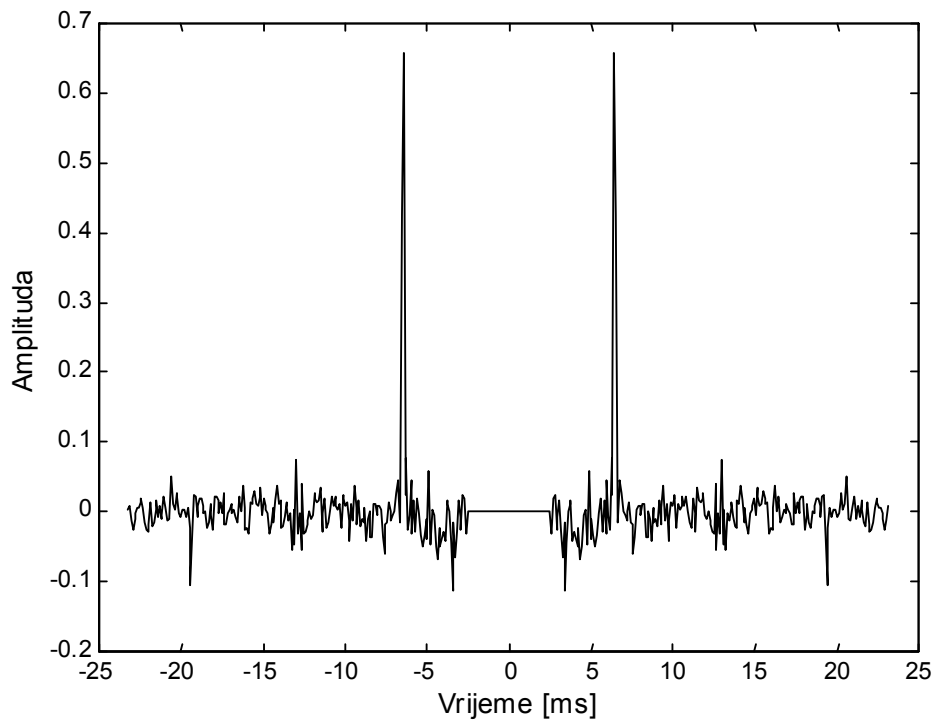
```
plot(t,[ltp(N/2+1:N) ltp(1:N/2)]);
```



Slika 13.2-5 Nisko-vremenski dio signala

Prikaz HTP-a:

```
plot(t,[ltp(N/2+1:N) ltp(1:N/2)]);
```



Slika 13.2-6 Visoko-vremenski dio signala

Tako pronađene LTP i HTP treba propustiti kroz inverzni karakteristični sustav $D^{-1}[\]$ za dekonvoluciju kako bi pronašli pobudu sustava $e(n)$ i impulsni odziv sustava $h(n)$. Prvo ćemo naći DFT LTP-a i HTP-a:

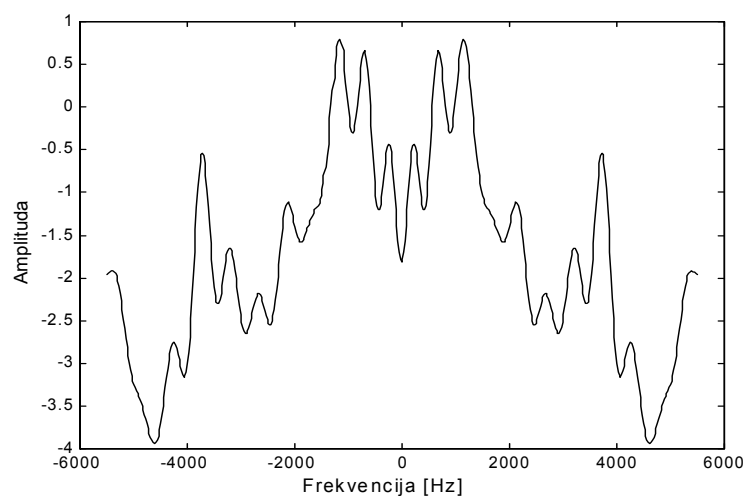
```
ltp_spect=fft(ltp);
htp_spect=fft(htp);
```

Pošto su LTP i HTP parni signali(simetrični oko nule) njihovi spektri trebaju također biti parni i imati samo realni dio. Zbog numeričkih grešaka koje unosi MATLAB-u spektre moramo učiniti na "silu" realnima:

```
ltp_spect=real(ltp_spect);
htp_spect=real(htp_spect);
```

Prikaz spektra LTP-a:

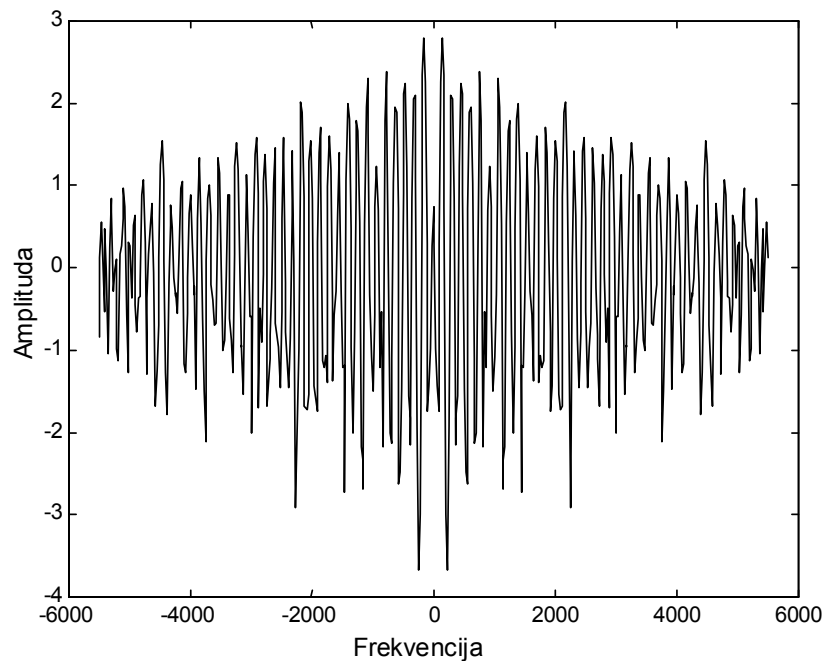
```
plot(f, [ltp_spect(N/2+1:N) ltp_spect(1:N/2)]);
```



Slika 13.2-7 DFT od LTP-a

Prikaz spektra HTP-a:

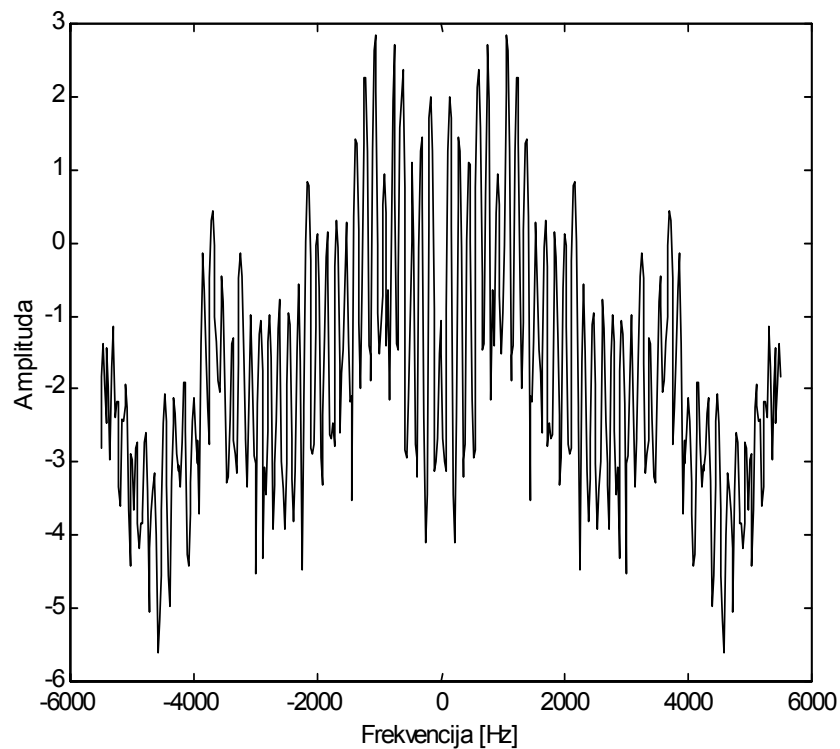
```
plot(f, [htp_spect(N/2+1:N) htp_spect(1:N/2)]);
```



Slika 13.2-8 DFT od HTP-a

Suma spektra LTP-a i HTP-a u stvari predstavlja logaritam modula spektra signala prikazanog na slici 13.2-3.

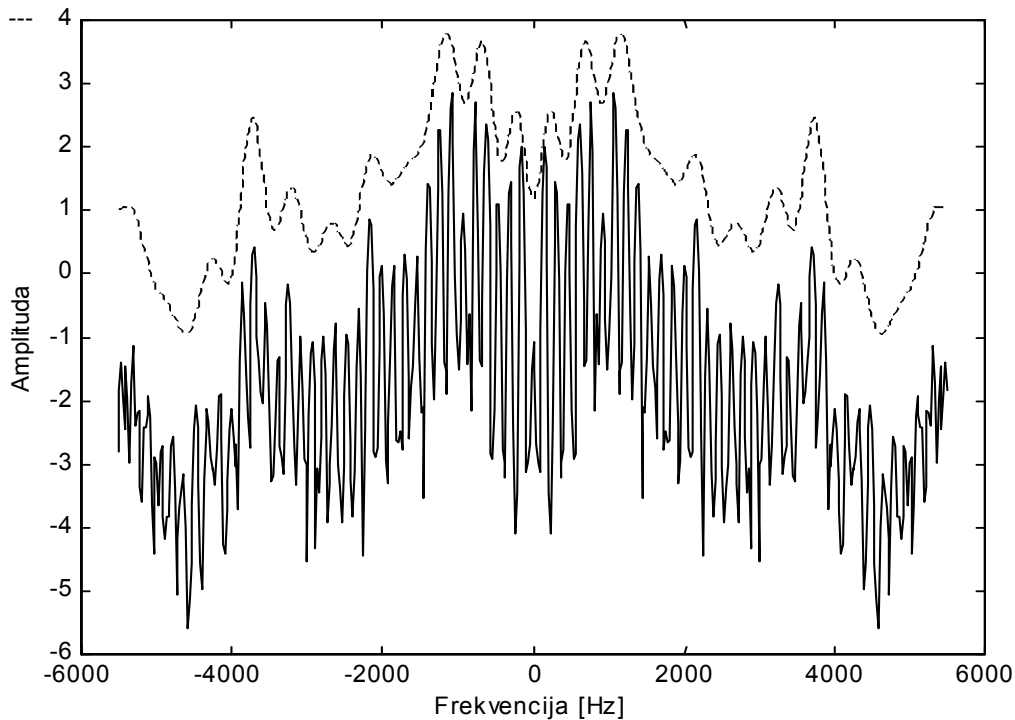
```
plot(f, [htp_spect(N/2+1:N)+ltp_spect(N/2+1:N)
        htp_spect(1:N/2)+ltp_spect(1:N/2)]);
```



Slika 13.2-9 Suma DFT-a LTP-a i DFT-a HTP-a

Spektar LTP-a opisuje sporo promjenljivu anvelopu logaritma modula spektra što možemo uočiti na slici 13.2-10 gdje su prikazani zajedno.

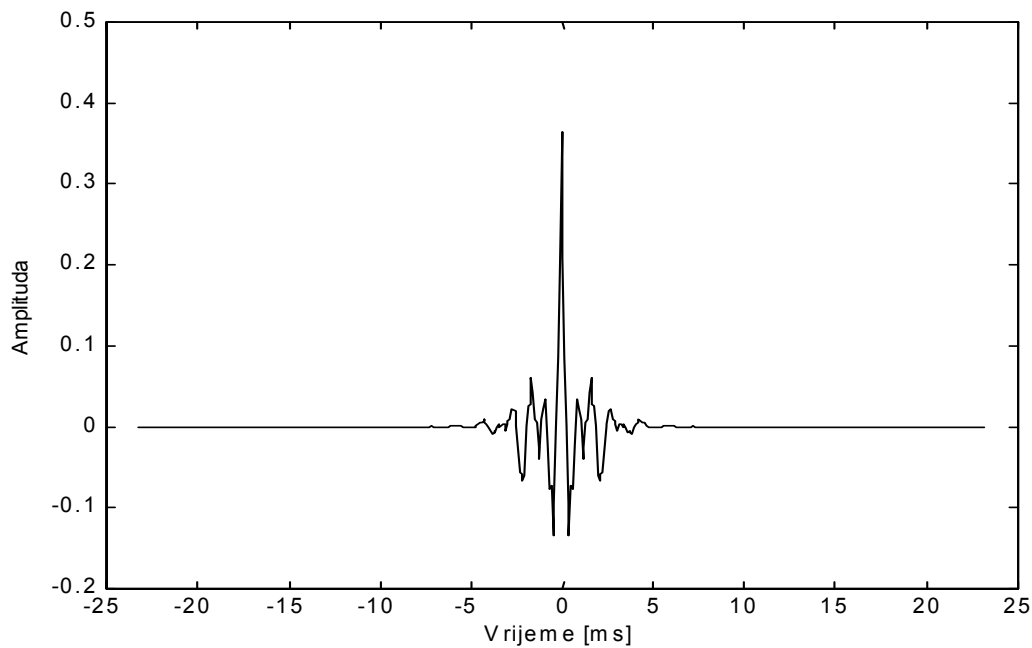
```
plot(f, [mod_spect(N/2+1:N) mod_spect(1:N/2)],  
f, [ltp_spect(N/2+1:N)+3 ltp_spect(1:N/2)+3]);
```



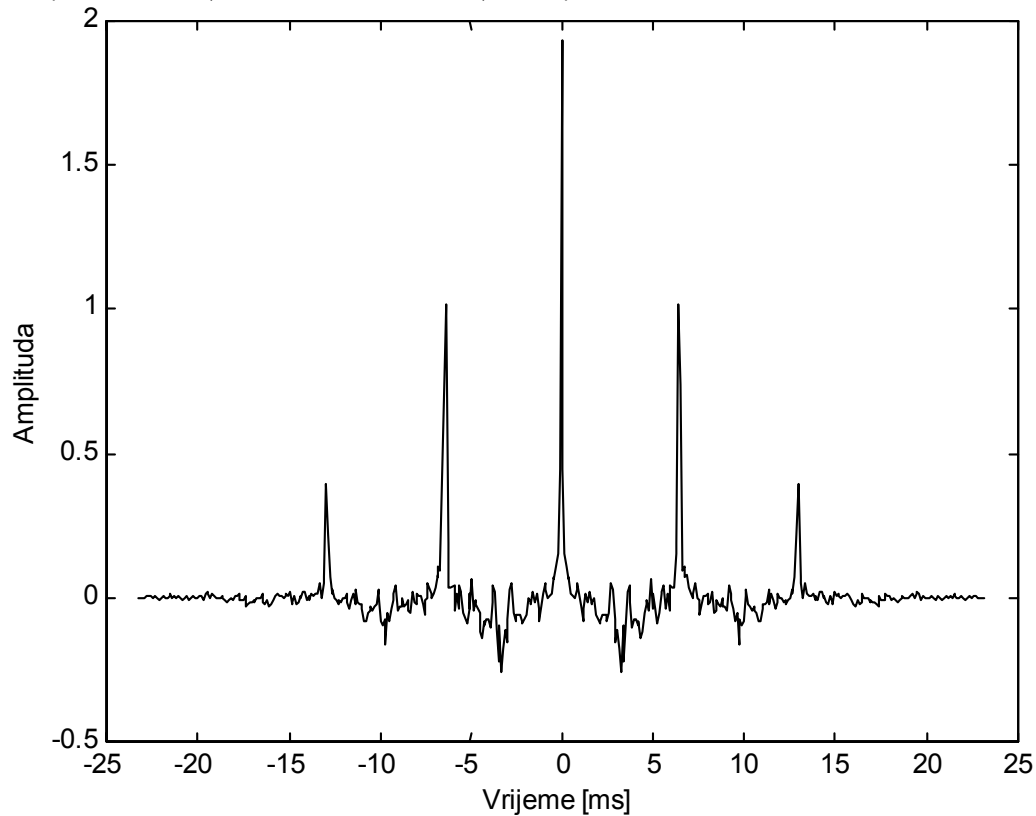
Slika 13.2-10 Logaritam modula DFT-a i DFT od LTP-a zvučnog glasa

Na kraju nam preostaje da ekponenciramo spektre LTP-a i HTP-a, te nad tako ekponencirane spektre primjenimo inverzni DFT. Kao rezultat tog niza radnji nad spektrom LTP-a dobivamo pobudni signal $e(n)$, dok je rezultat istog niza radnji nad spektrom HTP-a impulsni odziv sustava $h(n)$.

```
imp_resp=real(ifft(exp(ltp_spect)));  
excit=real(ifft(exp(htp_spect)));  
plot(t, [imp_resp(N/2+1:N) imp_resp(1:N/2)]);
```

Slika 13.2-11 Impulsni odziv sustava, $h(n)$

```
plot(t, [excit(N/2+1:N) excit(1:N/2)]);
```

Slika 13.2-12 Pobuda sustav, $e(n)$

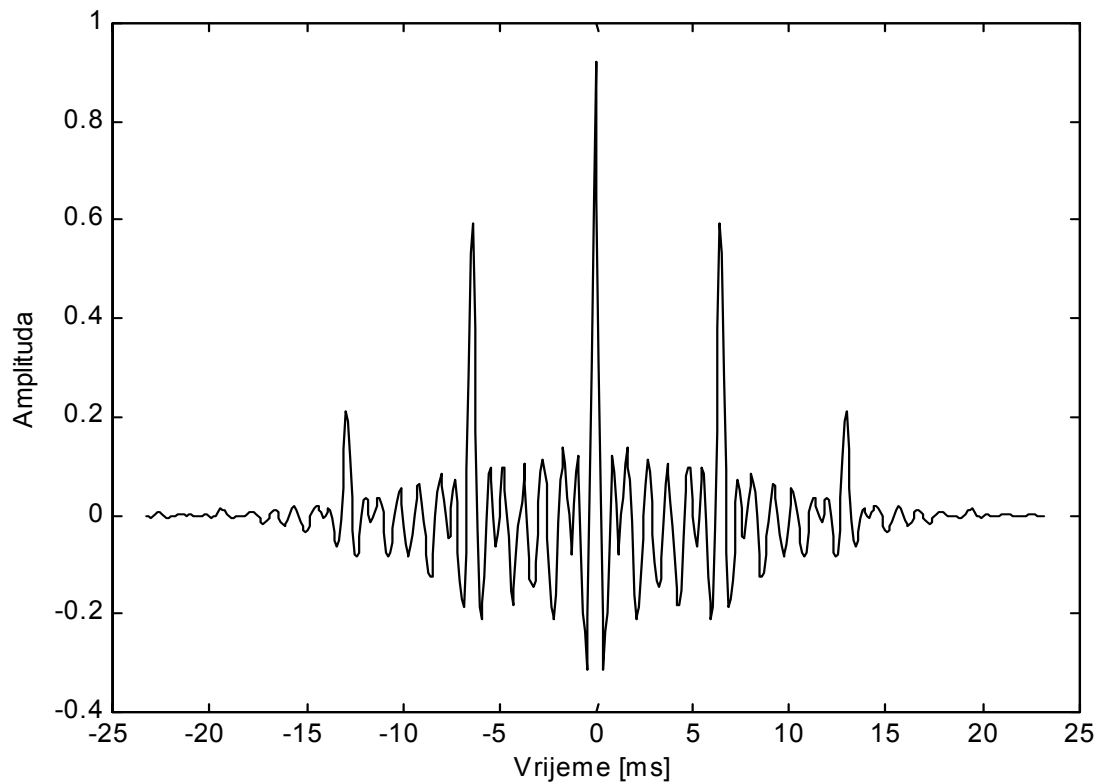
Vidimo da šiljci pobudnog signala prate oblik vremenskog otvora koji smo primjenili da bi izdvojili segment govornog signala za potrebe analize. Konvolucija pobudnog signala $e(n)$

i impulsnog odziva $h(n)$ bi na kraju trebala dati originalni segment govornog signala izdvojenog vremenskim otvorom. No, ako usporedimo slike 13.2-2 i 13.2-13, vidimo da to nije slučaj pošto smo radili samo sa modulom spektra tj. zanemarili smo njegovu fazu. Računanje DFT-ova signala dobivenog homomorfnom analizom i segmenta signala izdvojenog vremenskim otvorom bi pokazalo da su oni jednaki po modulu, ali da im se faze razlikuju.

```
imp_resp=[imp_resp(N/2+1:N) imp_resp(1:N/2)];
excit=[excit(N/2+1:N) excit(1:N/2)];

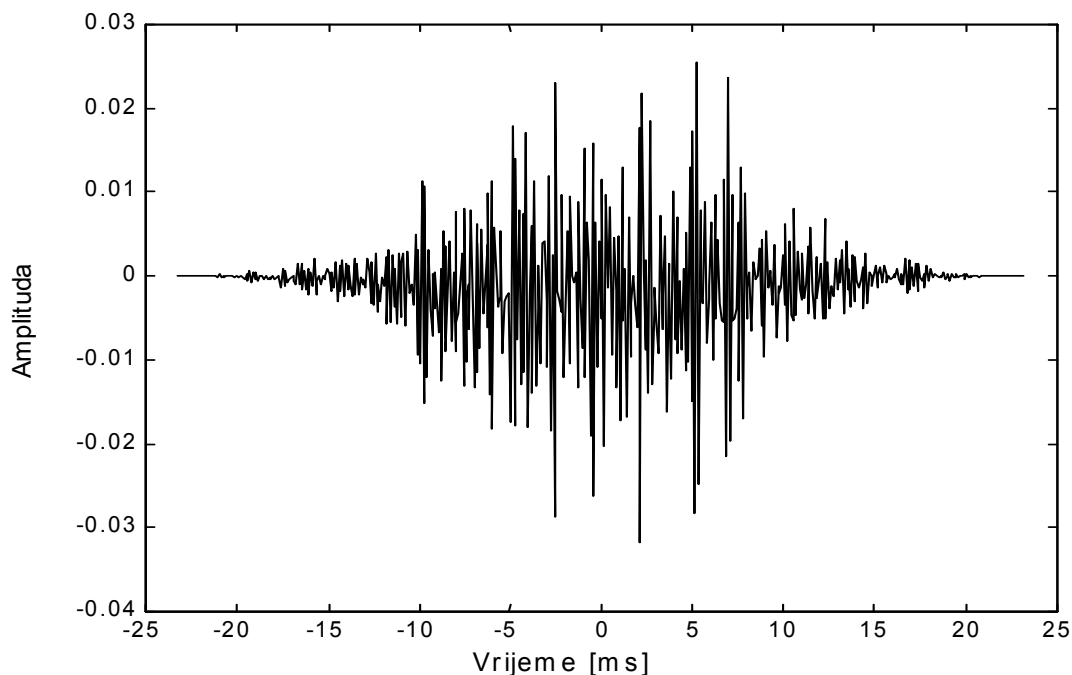
res=conv(imp_resp, excit).
res=res(N/2+[1:N]);           %uzimamo zanimljivi dio

plot(t, res);
```



Slika 13.2-13 Konvolucija signala $e(n)$ i $h(n)$

13.3 Homomorfna analiza bezvučnih glasova



Slika 13.3-1 Segment bezvučnog glasa izdvojenog vremenskim otvorom

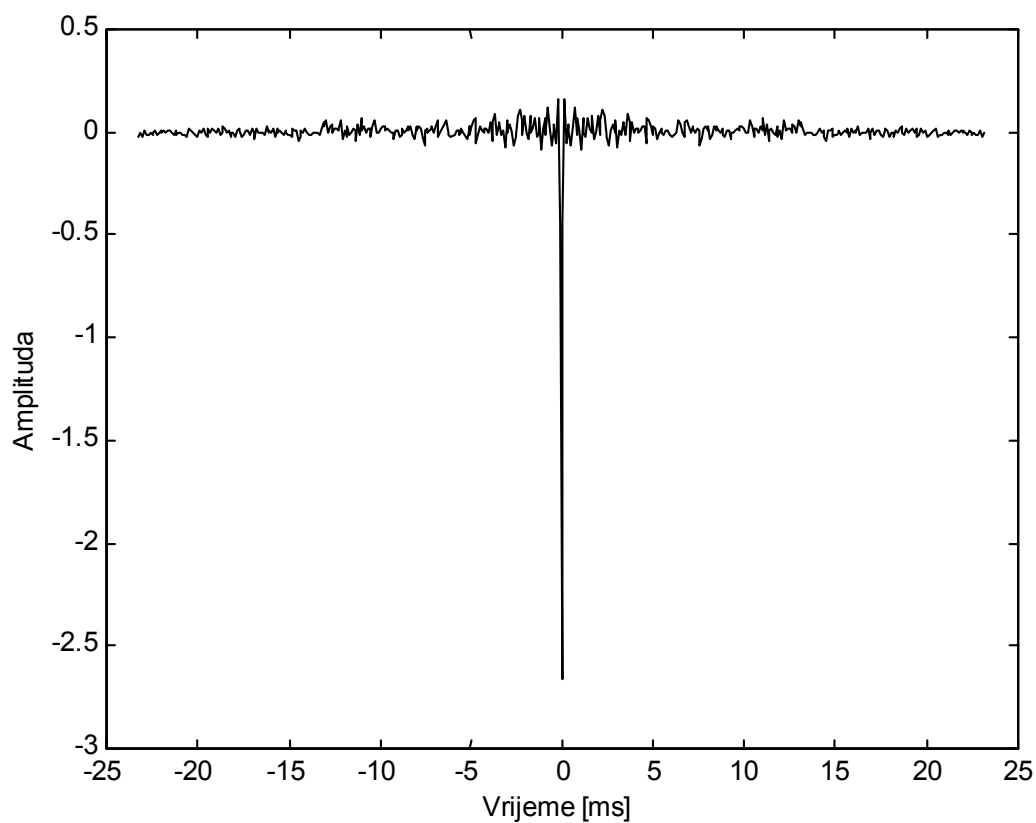
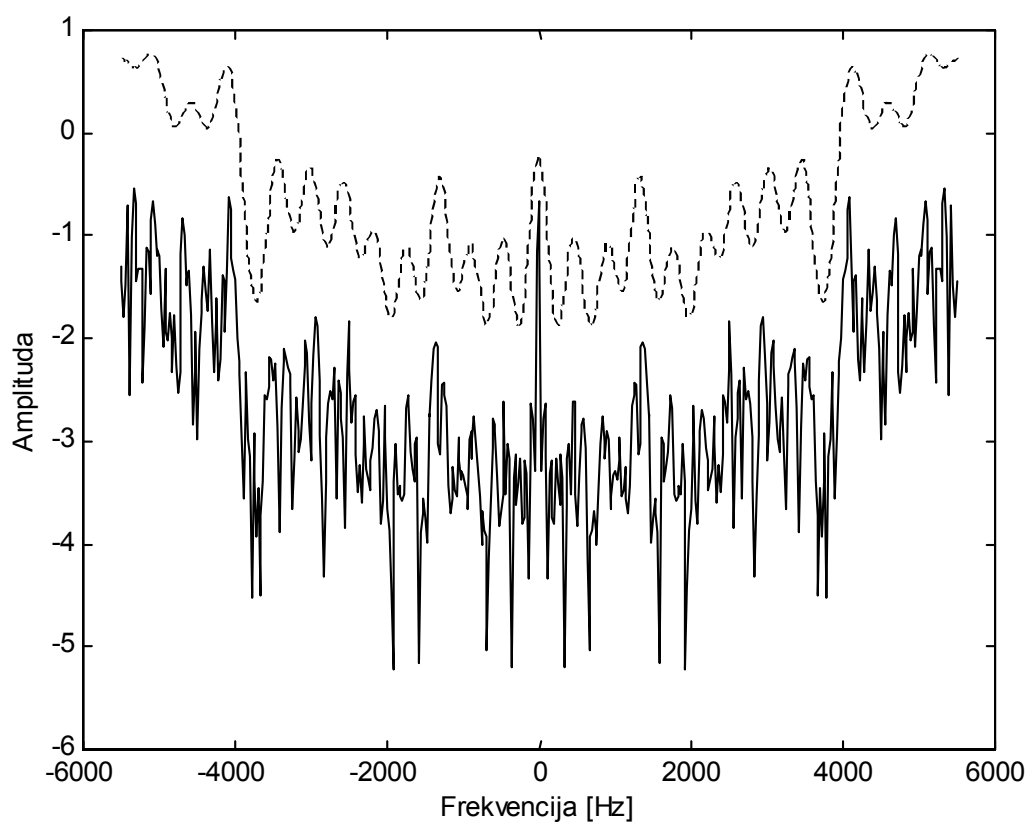
Sada ćemo na primjeru glasa 's' pokazati homomorfnu obradu bezvučnih glasova. Ponovno za potrebe primjera uzimamo zvučnu datoteku **sa2.wav**. Broj uzoraka N , koje uzimamo za analizu, neka bude kao i u prethodnom poglavlju jednak 512. Početak okvira analize neka bude 4000-ti uzorak.

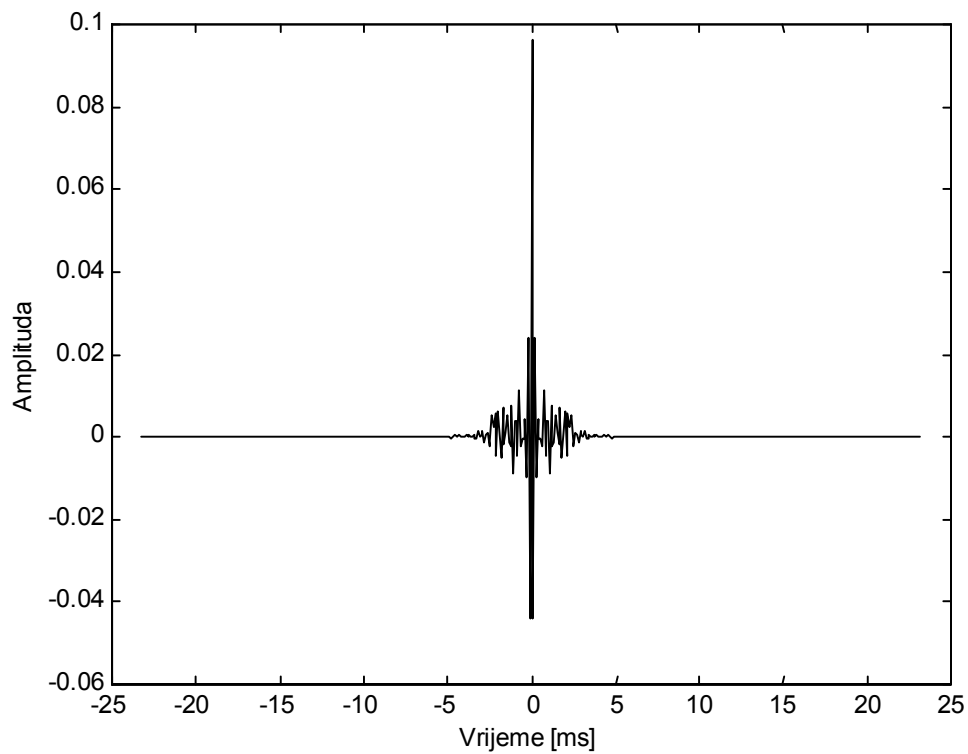
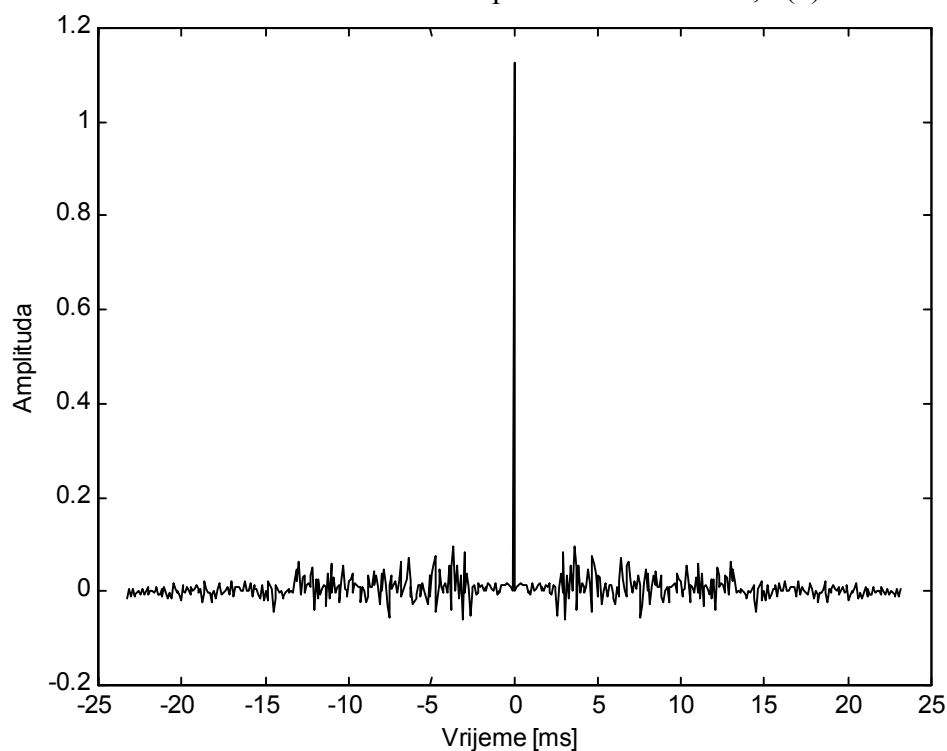
Postupak analize bit će jednak onome koji je bio prikazan u prethodnom poglavlju. Ovdje ćemo komentirati samo anomalije koje se javljaju prilikom homomorfne analize bezvučnih glasova, te ćemo pokazati način na koji otklanjamo te anomalije.

Segment signala izdvojen vremenski otvorom je u skladu s našim očekivanjima tj. on je aperiodski i šumovit kao što možemo vidjeti na slici 13.3-1.

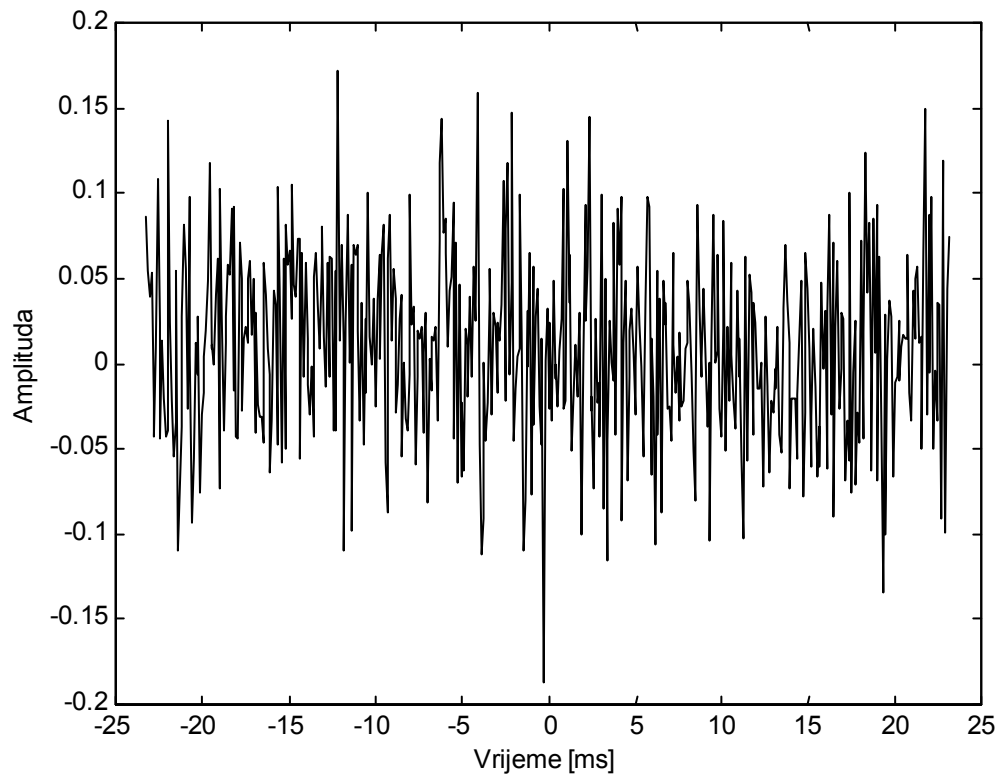
Daljnjom obradom dolazimo do kepstra bezvučnog glasa. Na slici možemo uočiti da imamo istaknutu samo jednu kepstralnu komponentu u LTP dijelu kepstra ($n=0$), dok HTP dio kepstra ima šumovit karakter bez izraženih maksimuma.

Nadalje, ukoliko promotrimo logaritam modula DFT-a bezvučnog signala i prikažemo zajedno s DFT-om LTP-a onda uočavamo da i ovdje DFT LTP-a predstavlja anvelopu logaritma modula DFT-a bezvučnog signala koji pak predstavlja formantnu strukturu govornog signala. To vidimo na slici 13.3-3. Nadalje, kako je to zacrtano blok shemom sustava za homomorfnu obradu govornog signala na slici 13.1-1, dobivamo impulsni odziv sustava $h(n)$ i njegovu pobudu $e(n)$. Oni su dani na slikama 13.3-4 i 13.3-5.

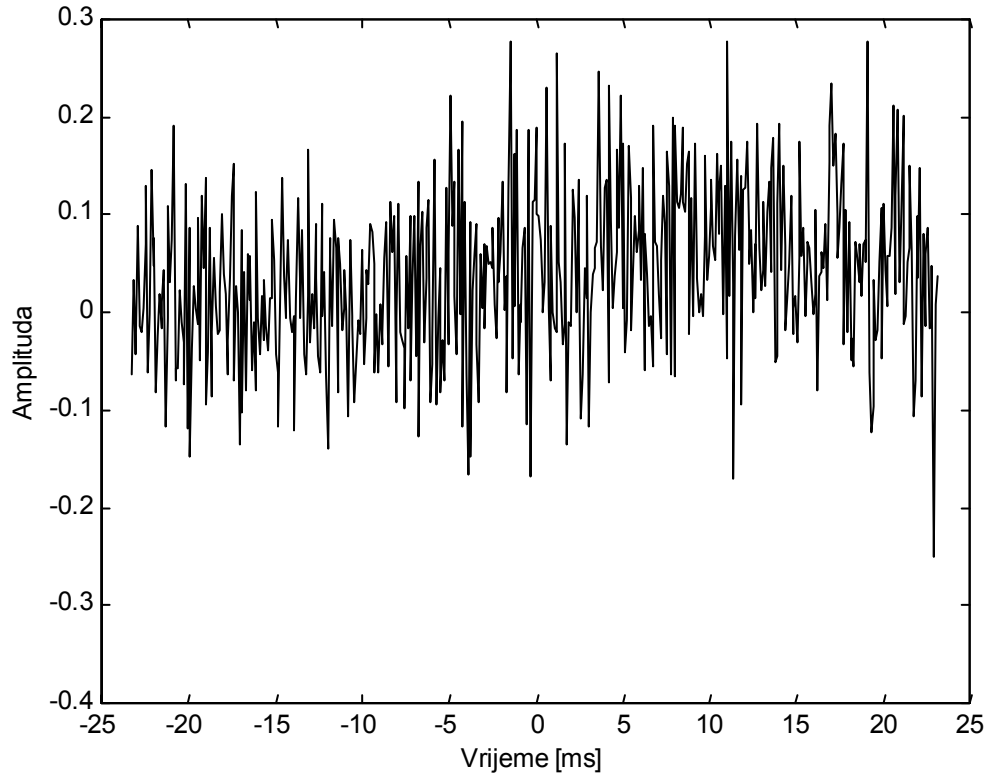
**Slika 13.3-2** Kepstar bezvučnog glasa**Slika 13.3-3** Logaritam modula DFT-a i DFT od LTP-a bezvučnog glasa

**Slika 13.3-4** Impulsni odziv sustava, $h(n)$ **Slika 13.3-5** Pobuda sustava, $e(n)$

Pobuda sustava nije u skladu s našim očekivanjima. Ovdje smo očekivali šumovit signal kao pobudu a dobili smo pobudu u obliku šiljka za $n=0$. To je posljedica toga što radimo s realnim kepstrom pa su faze svih komponenti DFT-a HTP-a jednake nuli.



Slika 13.3-6 Pobuda sustava sa slučajnom fazom, $e(n)$



Slika 13.3-7 Konvolucija signala $e(n)$ i $h(n)$

Stoga ćemo za svaku komponentu (osim za komponentu u $k=0$ i $k=N/2$) odrediti fazu na principu slučajnog odabira. Za komponente čije indeks k zadovoljava $0 < k < N/2$ faza se odabere slučajno, a pošto je zahtjev da faza bude antisimetrična (kako bi IDFT dao realni signal) za komponente čiji indeks k zadovoljava $N/2 < k < N$ uzmemo faze komponenti čiji indeks k zadovoljava $0 < k < N/2$ u obrnutom poretku i s obrnutim predznakom. Pobuda koju dobivamo iz DFT-a HTP-a s ovako odabranom fazom zadovoljava naša očekivanja, što se i vidi na slici 13.3-6. Radnje koje je potrebno izvršiti u MATLAB-u da bi dobili slučajnu fazu su sljedeće:

```
faz=rand(1,N/2-1)*2*pi;  
fazc=[0 faz 0 -faz(N/2-1:-1:1)];  
imp_resp=real(ifft(exp(htp_spect)*exp(sqrt(-1)*fazc)));
```

Rezultat konvolucije $h(n)$ i $e(n)$ je dan na slici 13.3-7. Vidimo da se on razlikuje od originalnog signala na slici 13.3-1, ali se pokazuje da moduli DFT-ova obaju signala su jednaki.

KRAJ

14. ODREĐIVANJE ZVUČNOSTI I OSNOVNE FREKVENCije TITRANJA GLASNICA

Student: Maja Adam

POČETAK

14.1 Kepstar i osnovna frekvencija titranja glasnica

Govorni signal je kvaziperiodičan za zvučne glasove. Ako je period takvog signala T , tada se njegov spektar sastoji od harmonika međusobno razmaknutih T^{-1} Hz (slika 14.1-1). Zbog toga možemo reći da je spektar zvučne sekvence govora "periodičan" duž frekvencijske osi sa "periodom" jednakim recipročnoj vrijednosti perioda govornog signala koji se analizira. Određivanje tog "perioda" spektra može se izvesti računanjem Fourierove transformacije spektra koja će onda imati šiljak na mjestu koje će odgovarati periodu titranja glasnica (engl. pitch period). Taj spektar kvadrata modula spektra poznatiji je pod imenom autokorelacija (engl. autocorrelation function):

$$r(\tau) \equiv \text{FT} \left[|F(\omega)|^2 \right], \quad \text{gdje je} \quad F(\omega) = \text{FT}[f(t)] \quad (14.1-1)$$

Ovo se također može zapisati kao:

$$r(\tau) = \text{FT} \left[|S(\omega)|^2 |H(\omega)|^2 \right] = \text{FT} \left[|S(\omega)|^2 \right] * \text{FT} \left[|H(\omega)|^2 \right] = r_s(\tau) * r_h(\tau) \quad (14.1-2)$$

pri čemu $*$ označava konvoluciju. Vidi se da su doprinosi vokalnog trakta i pobude međusobno konvoluirani u autokorelacijskim funkcijama. To će uzrokovati široke šiljke, a u nekim slučajevima i višestruke šiljke u autokorelacijskoj funkciji. Zbog toga autokorelacija nije dobra za određivanje perioda titranja glasnica (engl. pitch detection).

Rješenje je stvoriti novu funkciju u kojoj će doprinosi pobude i vokalnog trakta biti gotovo nezavisni i lako razlučljivi. Takva funkcija je logaritam kvadrata modula spektra:

$$\log |F(\omega)|^2 = \log \left[|S(\omega)|^2 |H(\omega)|^2 \right] = \log |S(\omega)|^2 + \log |H(\omega)|^2 \quad (14.1-3)$$

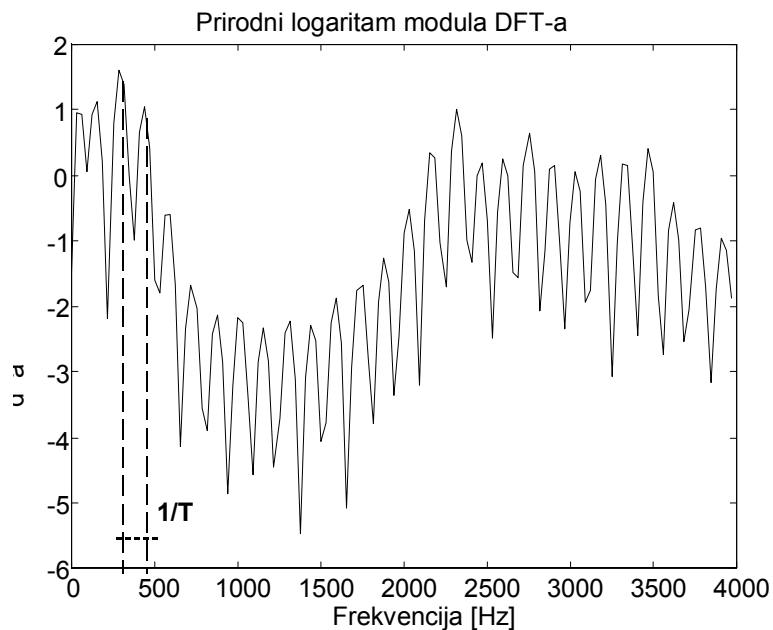
Fourierova transformacija ove funkcije ili keprstar (engl. cepstrum) čuva tu nezavisnost:

$$\text{FT} \left[\log |F(\omega)|^2 \right] = \text{FT} \left[\log |S(\omega)|^2 \right] + \text{FT} \left[\log |H(\omega)|^2 \right] \quad (14.1-4)$$

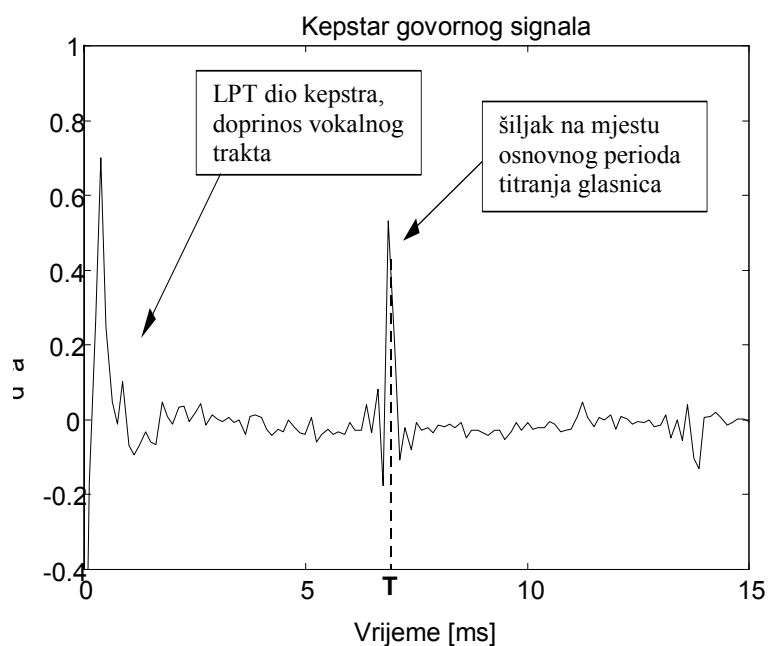
Doprinos vokalnog trakta stvorit će niskofrekvencijski dio keprstar (engl. low-time part ili LTP), a periodičnost titranja glasnica će se vidjeti kao visokofrekventni šiljci (engl. high-time part ili HTP), kako je prikazano na slici 14.1-2.

Za slučaj zvučne govorne sekvence (kada glasnice titraju) keprstar će imati šiljke na mjestu osnovnog perioda titranja glasnica i njegovim višekratnicima. Za bezvučne glasove keprstar nema šiljke, jer pri njihovom izgovoru glasnice ne titraju pa se ne može govoriti o periodu. Upravo se ovo svojstvo može iskoristiti kao osnova za određivanje zvučnosti govornog segmenta kao i za određivanje osnovnog perioda zvučnih govornih segmenata.

U postupku se nakon izračunavanja keprstar, u okolini očekivanog osnovnog perioda u keprstru traži šiljak. Ako je veličina šiljka iznad prethodno postavljenog nivoa, ulazni govorni segment je vjerojatno zvučan, pri čemu pozicija šiljka predstavlja dobru procjenu osnovnog perioda. Ako je veličina šiljka ispod nivoa, ulazni govorni segment je vjerojatno bezvučan.



Slika 14.1-1 Primjer logaritma modula spektra za zvučnu sekvencu govora



Slika 14.1-2 Kepstar zvučnog glasa, Fourierova transformacija signala sa slike 14.1-1

14.2 Kepstralna analiza

Program za ovu vježbu provodi vremenski kratkotrajnu kepstralnu analizu govornog signala, tako da provodi analizu u blokovima. U tom postupku, funkcija `noll_new` će na osnovu HTP dijela kepstara odrediti da li se radi o zvučnom segmentu ili bezvučnom segmentu. Ako se radi o zvučnom segmentu, tada će odrediti fundamentalnu frekvenciju tj. osnovni period titranja glasnica. Postupak se sastoji od pronalaženja najveće komponente

(najvišeg šiljka) u HTP dijelu kepstra, te njegove analize kako bi se utvrdilo da li on predstavlja zvučni segment i njegov osnovni period.

14.2.1 Funkcije i parametri

Parametri analize su definirani na sljedeći način:

```
param(1)=512;      % Broj uzoraka frame-a
param(2)=110;      % korak frame-ova izrazen u broju uzoraka
param(3)=80;       % Donja granicna frekvencija u Hz (-3 dB)
param(4)=0;        % faktor za pre-emphasis ili 0 ako je bez
param(5)=3e-3;     % minimalni ocekivani pitch period
param(6)=15e-3;    % maksimalni ocekivani pitch period
param(7)=0.175;    % v/uv prag
param(8)=1;        % HTP Cepstrum interpolation (1=Yes)
param(9)=1;        % Median-Linear Pitch filtering (1=Yes)
```

Parametri param(5) i param(6) određuju interval u kojem će se tražiti lokalni ekstremi u HTP dijelu kepstra. U našem primjeru gore taj period je 3-15 ms. Param(7) je prag za određivanje zvučnosti. Ukoliko se param(8) postavi na 1 tada će se period titranja glasnica izračunavati interpolacijom između dvije susjedne cjelobrojne vrijednosti, inače će period biti cijeli broj koji odgovara poziciji maksimuma. Za param(9) jednak 1 se nakon određivanja perioda titranja glasnica provodi još i završna MEDIAN-LINEAR filtracija koja će ukloniti lažne skokove u krivulji perioda titranja glasnica.

Nakon postavljanja parametara poziva se funkcija `noll_poz` sa parametrima:

```
noll_poz('suput.wav','suput.mat',param,1);
```

Ulazni podaci funkcije su stringovi sa nazivom ulaznog wav-file-a i izlaznog mat-file-a, vektor sa potrebnim parametrima analize, te da li želimo da se crtaju slike. Izlaz funkcije biti će izlazna .mat datoteka u kojoj su pohranjene sljedeće varijable:

```
% 1. wavefile .... String s nazivom ulaznog wav file-a
% 2. param ..... vektor parametara koristenih za analizu
% 3. fs ..... frekvencija otipkavanja
% 4. pp ..... pitch period
% 5. v ..... v/uv (1/0)
% 6. en ..... Cepstralni koef C0
```

Za pohranu ulaznog signala koristi se globalna varijabla tako da se izbjegne prenošenje te ogromne varijable pri pozivu funkcije `noll_new`:

```
global dat;
```

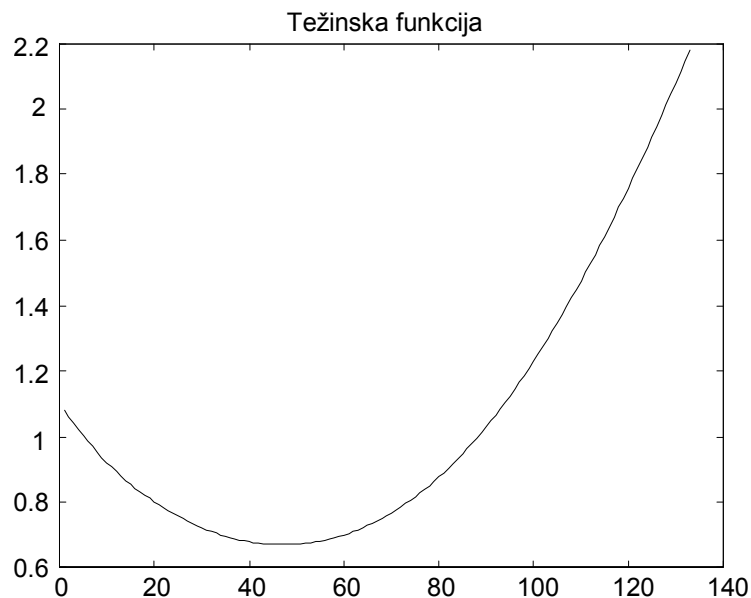
Ulazni govorni signal filtrira se prvo Butterworthovim visokopropusnim filtrom 8. reda za otklanjanje niskofrekvencijskih komponenti, a zatim se poziva funkcija `noll_new`:

```
[en,v,pp]=noll_new(N,fs,kor,pp_mn,pp_mx,tr,koef,interp);

% ... ulaz:  dat ..... signal, kao GLOBALNA varijabla
%           N ..... broj tocaka za FFT
%           fs ..... sampling rate
%           kor .... korak za novi Frame
```

```
%
    pp_mn .. minimalni pitch period (ms)
%
    pp_mx .. maksimalni pitch period (ms)
%
    tr ..... prag za voiced/unvoiced
%
    koef ... koeficijenti polimona za HTP weighting
%
    interp . ako je interp==1 radi se Cepstralna
              interpolacija kod traženja maksimuma
% ... izlaz: v ..... binarni vektor (1=voiced, 0=unvoiced)
%
    pp ..... pitch period (u broju uzoraka)
%
    en ..... nulti Cepstralni uzorak (energija)
```

Tada se za sve okvire analize u signalu računa kepstar i pohranjuje se informacija o energiji okvira (nulti koeficijent kepstra). Kod ‘visoko-kvofrentnih’ komponenata (‘rahmonika’) kepstra može se javiti problem s obzirom da se njihova amplituda bitno smanjuje i za zvučne segmente može se naći ispod praga za određivanje zvučnosti. Zbog toga se HTP dio kepstra množi težinskom funkcijom koja će te komponente izjednačiti. U vježbi se koristi težinska funkcija sa slike 14.2-1.



Slika 14.2-1 Težinska funkcija za kepstralne komponente

14.2.2 Interpolacija kod traženja maksimuma

Signal koji se koristi u analizi je diskretan, što ostavlja mogućnost pogreške pri određivanju perioda titranja glasnica veličine vremenskog razmaka između dva uzorka. Zato se pri postavljanju parametara analize može odabrati interpolacija kod traženja maksimuma u kepstru tj. perioda. Pretpostavka je da period nije višekratnik perioda uzoraka, nego da leži između neka dva uzorka. Uspoređujući neposredne susjede uzorka moguće je iz njihove veličine zaključiti u koju stranu treba pomaknuti mjesto perioda tj. da li period treba smanjiti ili povećati. Ako je lijevi susjed veći od desnog, period je manji od onog što ga određuje centralni uzorak, odnosno potrebno ga je pomaknuti u lijevo. Ukoliko je veći desni susjed, točan period je veći od onog određenog centralnim uzorkom.

Ako se pronađeni maksimum ne nalazi na samom rubu kepstra, uspoređuju se njegov desni i lijevi susjedni uzorak. Veći se susjed pamti i radi se interpolacija između centralnog uzorka i tog susjeda. Veći susjed i centralni uzorak spremaju se u dvije varijable, računa se

faktor za interpolaciju ovisno o amplitudi uzoraka, te se interpolacijom dobije novi iznos perioda i nova amplituda. Izvedba u Matlabu izgleda ovako:

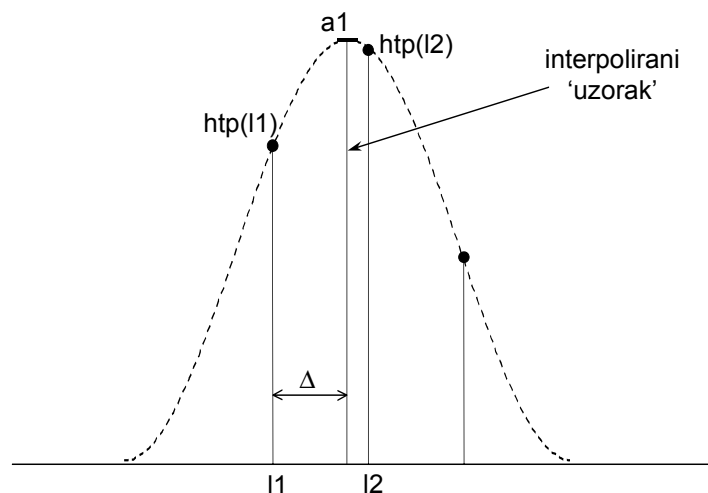
```
if (htp(pitch-1)>htp(pitch+1)),
    % ako je to lijevi susjed tada, prvi je lijevi, a drugi je centralni
    l1=pitch-1;
% inace prvi je centralni, a drugi je desni
else,
    l1=pitch;
end;
l2=l1+1;

% nadji faktor delta prema formuli za interpolaciju
alfa=htp(l2)/htp(l1);
delta=alfa/(1+alfa);

% Izraz za pravi quefreny najviseg harmonika
% ... u broju uzoraka, a ne u milisec
quef1=(l1+pp_min-1+delta);

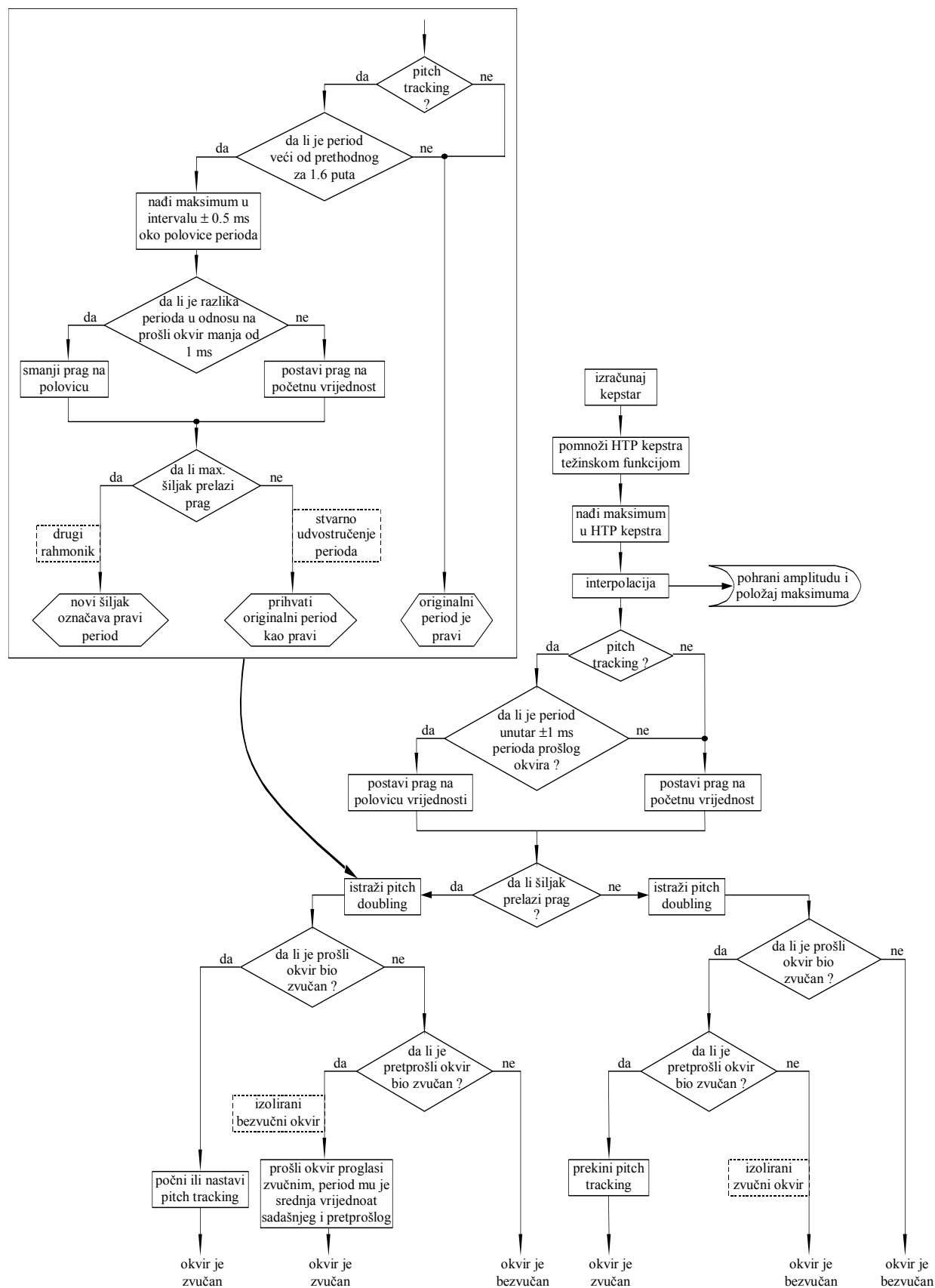
% ako je prvi veci od drugog po amplitudi
if (htp(l1)>htp(l2)),
    % tada koristi prvog za odredjivanje amplitude
    a1=pi*delta*htp(l1)/abs(sin(pi*delta));
% inace koristi drugog
else,
    a1=pi*(1-delta)*htp(l2)/abs(sin(pi*(1-delta)));
end;
```

Na ovaj način će period biti smanjen ili povećan, ovisno da li mu je veći lijevi ili desni susjed.



Slika 14.2-2 Primjer interpolacije za slučaj kada je veći lijevi susjed

Provjeravaju se još i sub-rahmonici, jer postoji mogućnost da postupkom nije odabran osnovni već neki viši rahmonik. Uzimaju se intervali koji se nalaze na npr. trećini, polovini i dvije trećine perioda pa se u njima ponavlja gornji postupak: pronađe se lokalni maksimum i izvrši se interpolacija. Za period se na kraju proglaši onaj šiljak koji ima najveću amplitudu.



Slika 14.2-3 Dijagram toka algoritma određivanja zvučnosti i pitch perioda

14.2.3 Dodatni problemi pri određivanju osnovnog perioda titranja glasnica

Problemi nastaju na krajevima zvučnih segmenata, pri pojavi izoliranih šiljaka ili kada je amplituda drugog rahmonika veća od amplitude prvog.

Na krajevima zvučnih segmenata kepstralni šiljci obično imaju smanjenu amplitudu i može se desiti da padnu ispod praga. Zbog toga se prag smanjuje za neki faktor (u našem slučaju prag se smanjuje na polovicu) ako je razlika perioda prema prošlom okviru analize manja ili jednaka 1 ms. Nakon završetka zvučnog segmenta, prag se vraća na normalnu vrijednost.

Također se može desiti jedan izolirani šiljak u kepstru koji će prijeći prag i time dati lažnu informaciju o zvučnom segmentu. Kako bi se to izbjeglo, provjeravaju se podaci o zvučnosti prošlog i pretprešlog okvira analize. U slučaju da trenutni šiljak prelazi prag (pretpostavka da je segment zvučan), a prošli je segment bio bezvučan, provjerava se pretprešli te ako je i on bezvučan zaključuje se da je započeo zvučni segment. Ako je prošli segment bio bezvučan, a pretprešli zvučan, tada je ovaj prošli jedan izolirani bezvučni segment između dva zvučna, pa se tada i on proglašuje zvučnim segmentom, a kao period se odabere srednja vrijednost trenutnog i pretprešlog segmenta.

Ukoliko se šiljak nalazi ispod praga (pretpostavka da je segment bezvučan), a prošli segment je bio zvučan, provjerava se pretprešli segment. Ako je on također zvučan, onda se nakon dva zvučna segmenta pojavio jedan bezvučan, pa je započeo bezvučni segment. Ako je pretprešli segment bio bezvučan, a prošli zvučan, tada se i taj prošli segment pretvori u bezvučan s obzirom da se radilo o izoliranom šiljku. Na ovaj se način svaki šiljak zapravo provjerava dvaput, budući da u sljedećem koraku odluka o njegovoj zvučnosti može biti promijenjena ako se utvrdi da je bio samo izolirani šiljak ili nedostatak šiljka.

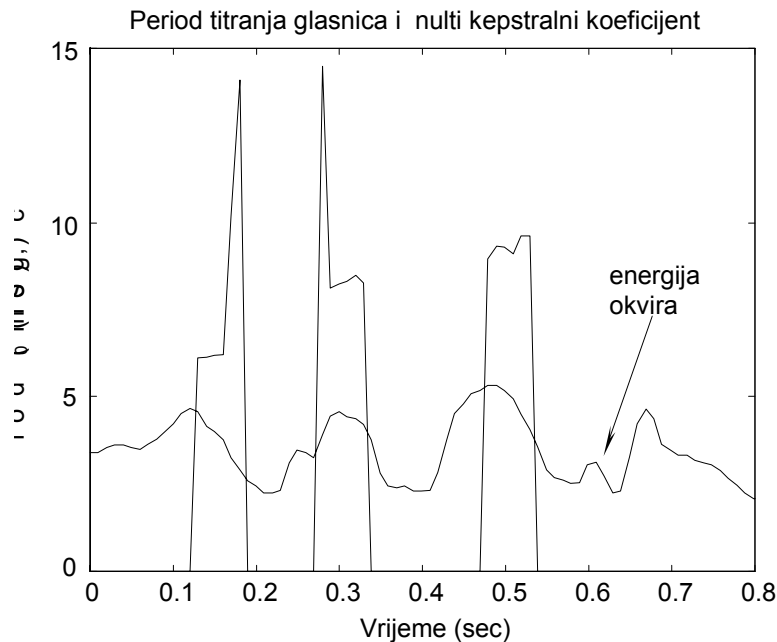
Ponekad se događa da drugi rahmonik kepstralnog šiljka bude veći od osnovnog te tako on bude izabran kao osnovni u postupku određivanja perioda. To je lažno udvostručenje perioda (engl. pitch doubling). Postupak mora dopuštati stvarno udvostručenje perioda, a eliminirati lažno. U slučaju lažnog udvostručenja perioda postoji još jedan šiljak (pravi) na polovici tog određenog (lažnog) perioda. Provjerava se da li je trenutni period titranja glasnica veći od prethodnog za više od 1.6 puta. Ukoliko jest, treba provjeriti da li je došlo do lažnog udvostručavanja perioda. Definira se podinterval ± 0.5 ms oko polovice perioda i u njemu se odredi lokalni maksimum te se gleda da li prelazi prag zvučnosti. Za slučaj da je razlika novog perioda i perioda prethodnog okvira manja ili jednaka 1 ms, prag se spušta na polovicu. Ako je novi šiljak iznad praga, on se proglašava pravim periodom.

Primjer

Na slici 14.3-1 nalazi se primjer perioda titranja glasnica kao funkcije vremena za riječ *suputnik*. U riječi suputnik glasovi *u* i *i* su zvučni, dok su glasovi *s, p, t, n* i *k* bezvučni. Jasno se vide tri zvučna segmenta i bezvučni segmenti između njih (dijelovi za koje je period namjerno postavljen na nulu). Prikazana je i energija okvira (nulti kepstralni koeficijent), pa je vidljivo kako zvučni segmenti imaju veću energiju od bezvučnih.

14.3 MEDIAN-LINEAR filtracija

Pogledamo li sliku 14.3-1 odmah se uočavaju dva šiljka u funkciji perioda titranja glasnica. Ti šiljci nisu stvarni iznos perioda već nastaju zbog nesavršenosti algoritma i potrebno ih je ukloniti tj. potrebno je cijelu dobivenu funkciju izgladiti. Budući da je praksa pokazala kako linearni filtri nisu dobri za takvu namjenu jer razmazuju i oštre skokove u funkciji koji tu moraju biti na prijelazima iz zvučnih u bezvučne segmente, koristi se nelinearan filter. Nelinearan filter koji se koristi je kombinacija median filtra i linearnog filtra.



Slika 14.3-1 Period titranja glasnica kroz vrijeme i energija okvira izražena preko nultog kepstralnog koeficijenta

Kod median filtriranja (engl. median filtering) izlazna vrijednost točke je jednaka medianu točaka sadržanih unutar prozora W . Median skupa brojeva se izračuna tako da se brojevi poredaju po veličini te se kao rezultat odabere onaj u sredini. Prozor W potrebno je odabrati tako da broj uzoraka koje obuhvaća bude neparan kako bi postojao srednji element. U Matlabu je on izveden na sljedeći način:

```
N_med=5; % Samo neparni median dozvoljen
for i=1:bf-(N_med-1), % bf je broj uzoraka okvira
    blok=pp(i:i+(N_med-1)); % pp je ulazni vektor sa pitch periodom
    blok=sort(blok);
    mf_pp(i)=blok((N_med+1)/2); % mf_pp je izlazni vektor sa izglađenim
end; % vrijednostima
```

Za nelinearno filtriranje pogodno je signale rastaviti ovisno o tome da li su glatki ili slični šumu. Tada se signal može $x(n)$ zapisati kao:

$$x(n) = S[x(n)] + R[x(n)] \quad (14.3-1)$$

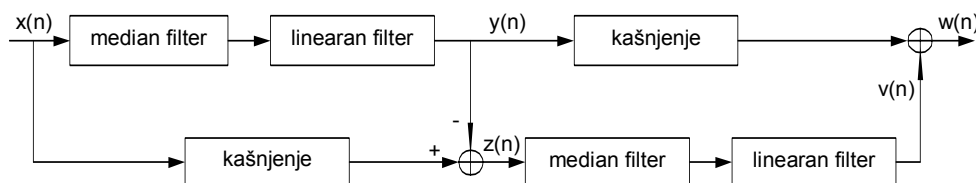
gdje je $S[x(n)]$ glatki dio signala $x(n)$, a $R[x(n)]$ njegov šumoviti dio. Za izdvajanje $S[x(n)]$ iz $x(n)$ koristi se median filtriranje.

Slika 14.3-2 prikazuje blok dijagram algoritma koji će izglati našu funkciju perioda titranja glasnica. Signal $y(n)$ je aproksimacija signala $S[x(n)]$, ali budući da to filtriranje nije idealno dodaje se još jedna kombinacija mediana i linearnog filtra koja će unijeti korekciju. Budući da vrijedi:

$$z(n) = x(n) - y(n) \approx R[x(n)], \quad \text{jer je} \quad y(n) \approx S[x(n)] \quad (14.3-2)$$

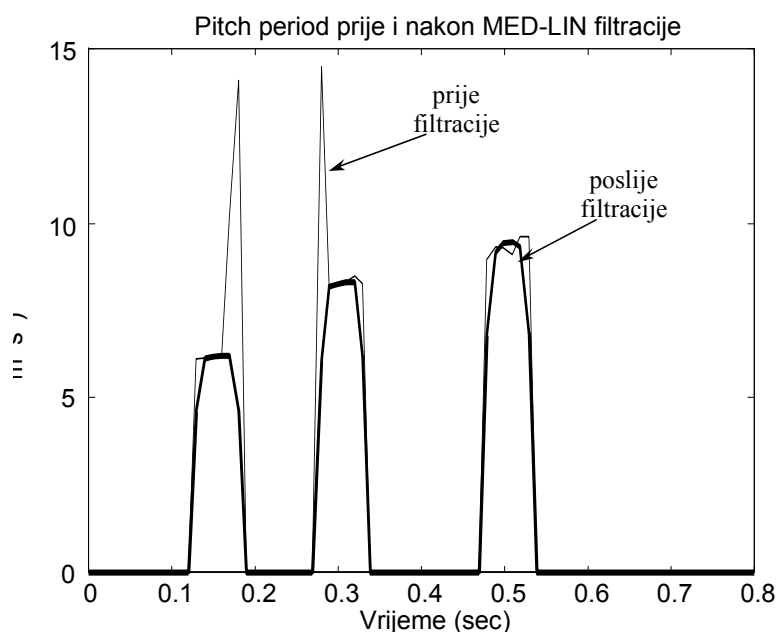
dodatno glaćenje $z(n)$ dat će korekciju koja se onda dodaje signalu $y(n)$ kako bi dobili $w(n)$, novu aproksimaciju $S[x(n)]$:

$$w(n) \approx S[x(n)] + S[R[x(n)]] \quad (14.3-3)$$



Slika 14.3-2 Blok dijagram MEDIAN-LINEAR filtracije

Na slici 14.3-3 nalazi se primjer MEDIAN-LINEAR filtracije za funkciju perioda titranja glasnica sa slike 14.3-1. Vidi se kako je nelinearan filter uklonio neželjene šiljke. Nedostatak ovakvog filtriranja je to što ipak razmazuje nagle prijelaze. Vidi se da je krivulja perioda na mjestu prijelaza iz zvučnog u bezvučni segment zaobljena umjesto da ima oštre skokove.



Slika 14.3-3 Funkcija osnovnog perioda titranja glasnica prije i poslije MEDIAN-LINEAR filtracije

14.4 Zadaci

Zadatak 1.: Izvesti postupak određivanja zvučnosti i perioda titranja glasnica za govorni zapis glava.wav. Odrediti periode titranja glasnica za prvi i drugi glas *a*.

Zadatak 2.: Snimiti vlastiti govor tako da za neki zvučni glas (npr. samoglasnik) frekvencija titranja glasnica varira.

PAŽNJA! Pri snimanju govora frekvencija otipkavanja mora biti postavljena na 11025 Hz. Ako se zapis snimi sa nekom drugom frekvencijom otipkavanja, program neće raditi pravilno budući da su svi ostali parametri postavljeni baš za tu frekvenciju.

KRAJ

15. PRIMJER KOMPLETNOG KODERA GOVORNOG SIGNALA