SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1642

IZVEDBA SUSTAVA ZA DETEKCIJU GOVORA

Nikola Petrak

Zagreb, siječanj 2011

Sadržaj:

1.Uvod

Detekcija govornog signala (u literaturi poznata po skraćenici VAD od engleskog Voice Activity Detection) je postupak korišten u obradi govora pri kojem sustav prepoznaje prisutnost odnosno odsutnost ljudskog govora u zvučnom zapisu. Tri glavna područja upotrebe VAD-a su automatsko prepoznavanje govora, kodiranje i diskontinuirani prijenos govora te pri smanjenju pozadinskih smetnji okoline i poništavanja jeke. Pri kodiranju govora VAD služi za izbjegavanje nepotrebnog kodiranja tihih predjela govora u pauzama između riječi što smanjuje propusnost potrebnu za prenošenja govora putem mreže (VoIP). Kod automatskog prepoznavanja govora VAD povećava točnost raspoznavanja govornih uzoraka. Pošto nalazi široku primjenu, razvijene su brojne inačice VAD algoritama koje pružaju brojne mogućnosti i kompromise između latencije, osjetljivosti, preciznosti i potrebnih računalnih resursa.

Razvoj detekcije govornog signala tekao je paralelno sa otkrićima u polju digitalne obrade govora, točnije automatskog prepoznavanja govora (eng. ASR- Automated speech recognition), čiji početci sežu u pedesete i šezdesete godine prošlog stoljeća. Danas, kada imamo 3.5 generaciju sustava za obradu govora, detekcija govora i dalje predstavlja izazov te većina VAD algoritama zakazuje kada se smanjuje omjer signala i buke (SNR- eng. Signal to Noise Ratio) ili kada buka nadglašava govorni signal. Tijekom prošlog desetljeća brojni istraživači razvili su različite robusne algoritme za detektiranje govora u bučnim uvjetima, a temelje se na kombinacijama ranijih saznanja te će ukratko biti izložena u radu. Uspješnije VAD metode uključuju one bazirane na energetskim pragovima, detekciji visine tona (eng. pitch detection), spektralnoj analizi, broju prijelaza nule (eng. zero crossing rate), mjeri periodičnosti (eng. periodicity measure) i statistike višeg reda u rezidualnoj domeni linearno prediktivnog kodiranja (LPC- eng. Linear Predictive coding).

U radu će se prvo napraviti kratki povjesni pregled razvoja detekcije i obrade govora, a potom radi boljeg razumijevanja materije biti će opisan način nastajanja i percepcije govora, potom izvori i tipovi smetnji koji se javljaju pri obradi govora. Nakon toga dan je opis nekoliko algoritama s obzirom na način

njihove izvedbe. Na kraju slijedi detaljan opis odabranog algoritma za implementaciju u Matlabu i na blackfin dsp sustavu te kratki opisi alata i sustava na kojem je algoritam ostvaren.

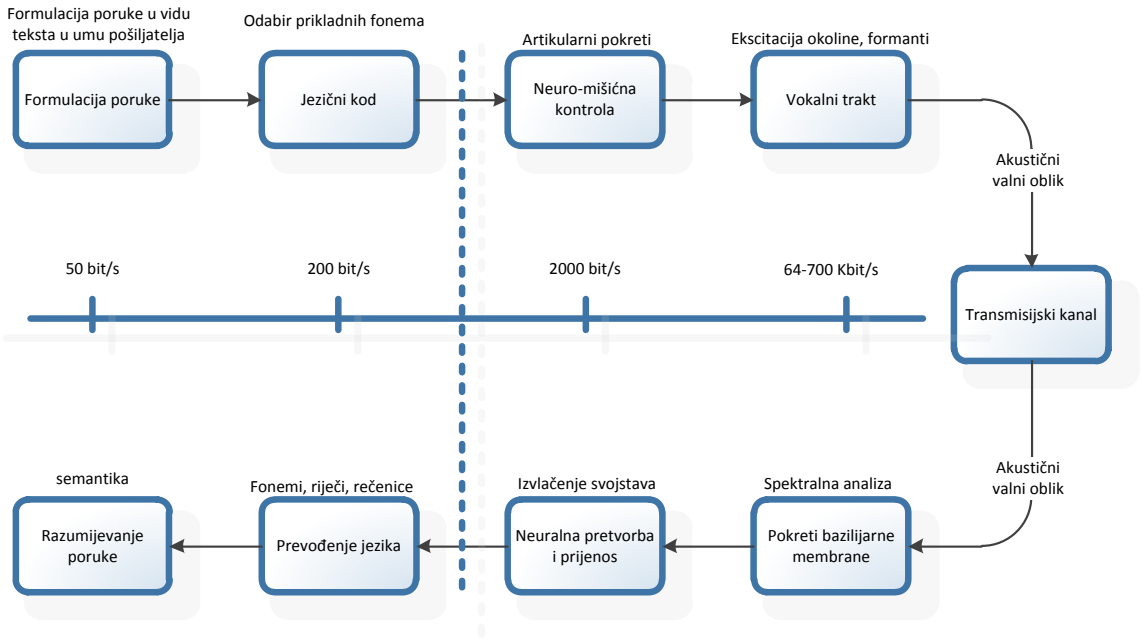
2. Povijesna crtica

Prvi pravi početci detekcije govora započinju u šezdesetim godinama 20. stoljeća kada su istraživači u Bellovim laboratorijima pokušali iskoristiti osnovne ideje akustične fonetike te uspjeli konstruirati sustav za prepoznavanje pojedinih znamenki brojeva za jednog govornika. Sustav nazvan TASI, bazirao se na mjerenju spektralnih rezonancija pri izgovaranju samoglasnika. 1963. Nagata i djelatnici NEC-ovih laboratorija konstruiraju hardversko prepoznavanje govora bazirano na segmentatoru govora kombinirano sa nul-prelaznom analizom. Martin je razvio set osnovnih vremensko-normalizirajućih metoda, zasnovanih na sposobnosti detektiranja početka i kraja govora. U isto vrijeme u Sovjetskom Savezu razvijana je metoda dinamičkog programiranja za vremensko poravnavanje para govornih iskaza. U istom desetljeću imamo i nekolicinu Reddyevih otkrića. 70ih godina razvoj prepoznavanja govora je uvelike napredovao u smjeru raspoznavanja izoliranih riječi, zasnovanih na Velichkovim i Zagoruykovim studijama. U SAD-u Itakura razvija metodu linearnog prediktivnog kodiranja koje je uspješno korišteno u niskobitovnim kodiranjima govora. U AT&T Bell laboratorijma vodila su se istraživanja prepoznavanja govora neovisno o govorniku. Kao što je prepoznavanje izoliranih riječi bio cilj u 1970' problem prepoznavanja spojene riječi, naime bilo je potrebno kreirati robusni sustav koji prepoznaje kontinuirano govoren niz riječi na osnovi spajanja uzoraka pojedinih riječi te su u tu svrhu razvijeni brojni algoritmi. Tijekom 80' godina istraživanje govora karakterizirano je promjenom tehnologije sa one bazirane na predlošcima riječi na statistički modelirane metode, posebice skrivenih Markovljevih modela (HMM). U to isto vrijeme počinje korištenje neuralnih mreža u prepoznavanju govora.

3. Općenito o govoru i mediju kroz koji se govor prenosi

3.1Nastanak govornog signala

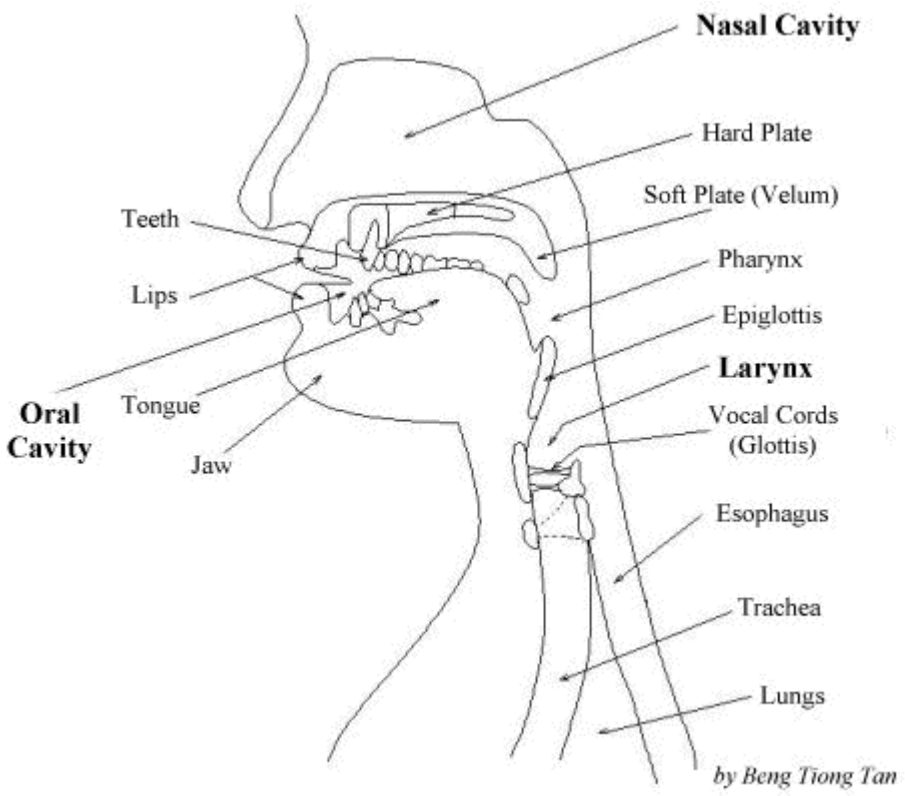
Ljudski je govor poseban u odnosu na druge zvukove s obzirom na to kako nastaje i kako ga percipiramo. Upravo se te posebnosti mogu iskoristiti za obradu govora računalom. Slika 1 prikazuje cjelokupni proces nastanka i percipiranja govora od stvaranja poruke preko prijenosa signala do razumijevanja poruke na slušateljevoj strani. Navedenu formulaciju Denes i Pinson nazivaju govorni lanac te je na slici prikazano koliku količinu informacije se prenosi u kojem segmentu.



Slika 1. Govorni lanac

Nakon formulacije poruke u mozgu pošiljatelja i odabira prikladnih fonema za odašiljanje poruke, artikulacijskim procesima oblikuje se glas. Ljudski glas, kao nositelj poruke, nastaje prolaskom kroz govorne organe koji formiraju zvuk, a raspon mu je od 60 Hz do 8 kHz s dinamičkim rasponom od 40 db. Proizvodnja glasa počinje istiskivanjem zraka iz pluća kroz dušnik na kraju kojeg se nalaze glasnice koje titraju pod strujanjem zraka te ovisno o svom položaju, proizvode zvuk različite frekvencije. Glasnice se ponašaju kao mehanički oscilator, koji prelazi u stanje relaksacijskih oscilacija uslijed struje zraka koja kroz njih prolazi. Nastali zvuk dalje se formira u grlu te u nosnoj i usnoj šupljini. U usnoj šupljini na njega utječemo položajem i pokretima jezika, stražnjeg (mekog), srednjeg i

prednjeg nepca, zubiju te usana. Prema nosnoj šupljini imamo velum ili resicu koja zatvara usnu šupljinu prema nosnoj i samu šupljinu koja završava s nosnicama. Na slici 2 prikazan je glasovni trakt. Nastali zvukovi proizvedeni u vokalnom traktu, kao što ćemo vidjeti, imaju specifičan oblik prilikom vremenske i frekvencijske analize. Naime vokalni trakt proizvodi određene frekvencije koje karakteriziraju pojedine glasove, a te se rezonantne frekvencije zovu formanti. Na primjer glas „a“ ima formante F1 na 660 Hz, F2 na 1700 Hz i F3 na 2400Hz .



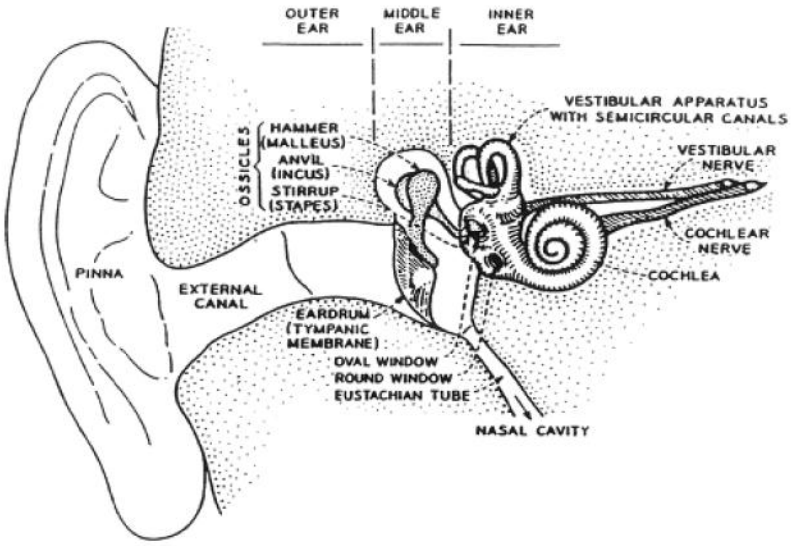
Slika 2. Govorni trakt

Po osnovi konfiguracije i otvora vokalnog trakta, glasove u hrvatskom jeziku možemo podijeliti na tri osnovne grupe, otvorni glasovi (samoglasnici, vokali), poluotvorni glasovi (glasnici, sonanti) i zatvorni glasovi (suglasnici ili konsonanti). Vokali ili samoglasnici su glasovi najveće energije, kod kojih je vokalni trakt većim dijelom otvoren, a tokom cijelog trajanja izgovora glasnice titraju. Glasnici su također zvučni glasovi jer pri njihovom izgovoru glasnice titraju ali se uslijed približavanja ili dodirivanja pojedinih organa (artikulatora) u vokalnom traktu, otvor za prolaz zraka sužava ili djelomično zatvara. Kod tih glasova amplituda je veća i uočavamo pravilnost odnosno periodičnost signala. Kod se zatvornih glasova to nije uvijek slučaj i dijelimo ih na zvučne i bezvučne te kod

bezvučnih ne uočavamo pravilnosti i imaju nižu energiju jer se prolaz zračnoj struji potpuno zatvara te je s toga izlaz manji.

3.2 Percepcija govornog signala

Model percepcije govora sastoji se od nekoliko koraka od kojih je prvi efektivna konverzija akustičnog valnog oblika koji putuje zrakom u spektralnu reprezentaciju istog. To se događa u unutarnjem uhu na bazilijarnoj membrani koja se ponaša kao nejednoliki spektralni analizator koji prostorno razdvaja spektralne komponente dolaznog govornog signala. Sljedeći korak je prijenos spektralnih svojstava primljenog signala kroz mrežu živčanih stanica do centara u mozgu koji dekodiraju odnosno prepoznaju zvučna svojstva iz spektralnih svojstava te potom iz tih zvučnih svojstava prepoznati foneme, i riječi.



Slika 3. Slušni kanal

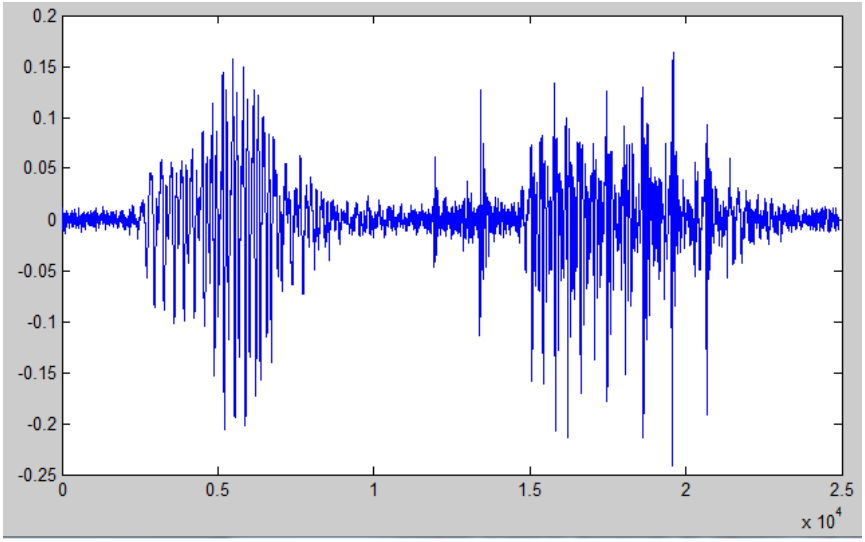
3.3 Buka i smetnje u govornom kanalu

Transmisijski dio govornog lanca nije idealan te se u stvarnom svijetu javljaju smetnje koje izmjenjuju poruku te potrebno uvesti pojam buke u komunikacijskom kanalu. Buku definiramo kao nepoželjni signal koji ometa komunikaciju, mjerenje ili obradu drugog, korisnog, signala koji nosi informaciju. Kao takva, buka je glavna prepreka koju svaki VAD algoritam mora premostiti. Izvora buke ima mnogo i variraju, od audio frekvencijske akustične buke koja dolazi iz gibajućih ili vibrirajućih izvora kao što su razni strojevi, vozila u kretanju, vjetar kiša i tako dalje, pa sve do radio-frekvencijske elektromagnetske buke koja ometa prijenos i

primitak glasa odnosno podataka. Buku prema izvorima nastajanja možemo klasificirati u nekoliko kategorija: akustičnu, elektromagnetsku, elektrostatsku, kanalsko izobličenje (jeka i degradacija signala), buka pri obradi (A/D D/A pretvorba). Nadalje ovisno frekvencijskim (vremenskim) karakteristikama buku dijelimo na: uskopojasnu (50-60 Hz; primjer zujanje električnog napajanja), bijelu buku(slučajna buka koja ima ravan spektar snage; teoretski sadrži sve frekvencije u podjednakom intenzitetu, pojasno ograničenu bijelu buku(slično kao bijela buka samo ima ograničen spektar), obojenu buku(čiji spektar snage ima neravan oblik), impulsnu buku (kratkotrajni pulsevi slučajne amplitude i trajanja) te na tranzijentne impulse buke.

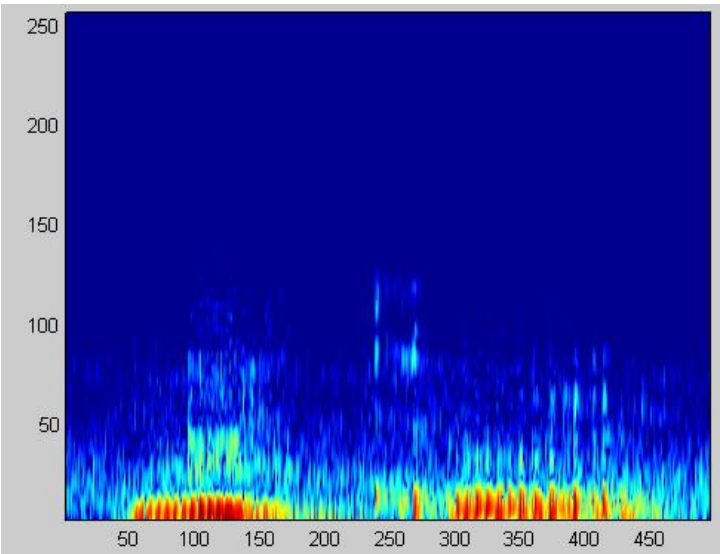
3.4 Obrada govornog signala

Na temelju dosadašnjih razmatranja zaključujemo da se sustav procesa nastanka govora kao i njegova percipiranja može modelirati i matematički kao niz diskretnih i kontinuiranih sustava, a samim time moguća je implementacija na računalu.U ovom radu, zbog opširnosti područja, bit ce opisani samo oni matematički modeli bitni za razumijevanje osnovnih principa detekcije govorne aktivnosti. Započinjemo sa valnim oblikom govora te njegovom frekvencijskom analizom. Na slici 4 prikazan je valni oblik riječi "Nikola" snimljen pomoću programskog paketa audacity uzorkovan brzinom 44 kHz, a prikazan je u matlabu:



Slika 4. Valni oblik riječi "Nikola"

Spektrogram danog signala, kao jedan od najvažnijih alata za analizu govora, a prikazuje promjenu spektralne gustoće kroz vrijeme, dobiven je postupkom kratko-vremene Fourierove transformacije (STFT) za svaki pojedini okvir signala konačne (kratke) širine i prikazan je na slici 5.



Slika 5. Spektrogram riječi "Nikola"

3.3.1 LPC analiza

LPC analiza je metoda obrade govora koja je najprikladnija za određivanje njegovih osnovnih parametara uz minimalnu numeričku složenost i veliku brzinu izračunavanja. Osnovni princip linearne predikcije je taj da se uzorak signala može aproksimirati linearnom kombinacijom prethodnih uzoraka. Skup koeficijenata prediktora određuje se minimizacijom sume kvadrata razlike stvarnih govornih uzoraka i uzoraka dobivenih linearnom predikcijom na ograničenom vremenskom intervalu. Postupak linearne predikcije je sljedeći. Signal otipkan nekom frekvencijom, predikciju uzorka s(n) izračunavamo kao linearnu kombinaciju uzorka s(n-1), s(n-2),...,s(n-p), gdje je p stupanj odnosno red prediktora. Koeficijente linearne predikcije označavamo sa α. Linearnu predikciju možemo iskazati formulom:

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k) \quad (1)$$

Koeficijenti su stalni za pojedinu predikciju na nekom segmentu otipkanog signala te ih je potrebno odabrati da pogreška predikcije bude što manja. Za signale kod

kojih je energija predikcijske pogreške manja od energije polaznog signala kažemo da su korelirani, odnosno da imaju spektar sa izraženim maksimumima što LPC analizu čini pogodnom za obradu govora.

Srednja vrijednost pogreške predikcije na segmentu $s_n(m)$ dana je izrazom :

$$E_n = \sum_m e_R^2(m) = \sum_m \left(s_n(m) - \tilde{s}_n(m)\right)^2 \quad (2)$$

Izračunavanje predikcijskih koeficijenata provodi se nad vremenskim segmentom konačnog trajanja tako da se parcijalne diferencijalne jednadžbe od E_n po nepoznatim koeficijentima izjednače s 0.

$$\frac{\partial E_n}{\partial \alpha_i} = 0, i = 0,1, \dots, p \quad (3)$$

te tako dobivamo sustav od p linearnih parcijalnih diferencijalnih jednadžbi:

$$\sum_m s_n(m-i)s_n(m) = \sum_{k=1}^{p} \alpha_k \sum_m s_n(m-i)s_n(m-k) \quad (4)$$

Sređivanjem jednadžbe i supstitucijom sa $\phi_n(i,k) = \sum_n s_n(m-i)s_n(m-k)$ dobivamo izraz:

$$\sum_{k=1}^{P} \alpha_k \phi_n(i,k) = \phi_n(i,0) \quad (5)$$

Iz kojeg izračunavamo skup optimalnih koeficijenata prediktora koji minimaliziraju srednju kvadratnu pogrešku predikcije. Raspon sumacije po indeksu m nije fiksan no zbog vremenske promjenjivosti spektralnih svojstava signala on mora biti ograničen s toga se uvode postupci predikcije koji ga ograničavaju. Jedan od njih je i autokorelacijski postupak.

3.3.2 Autokorelacijska metoda

Jedna od formulacija linearne predikcije jest autokorelacijska metoda te ona ograničava granice sumacije pogreške predikcije tako da pretpostavlja da je segment signala $S_n(m)$ izvan intervala $k \in \ 0 \leq n \leq N-1$ jednak nuli. To zapisujemo diferencijskom jednadžbom $s_n(m) = s(m+n)w(m)$ gdje je w(m) vremenski otvor jednak nuli izvan intervala k. Vremenski otvor koji se koristi mora

prigušavati signal pri krajevima vremenskog signala, npr. Hammingov otvor, zbog pogreške velikog iznosa jer se pokušava provesti predikcija signala na temelju uzoraka jednakih nuli (na početku) odnosno predikcija signala koja je jednaka nuli na temelju uzoraka različitih od nule (na kraju). Zbog toga se na kraju dobiva izraz sa sljedećim rasponom indeksa:

$$\phi_h(i,k) = \sum_{m=0}^{N-1-(i-k)} S_n(m)s_n(m_{+i} - k) \quad , 1 \leq i \leq p, 1 \leq k \leq p \tag{6}$$

što je istovjetno kratkotrajnoj autokorelacijskoj funkciji:

$$\phi_n(i,k) = R_n(i-k) \tag{7}$$

te dobivamo izraz

$$\sum_{k=1}^{P} \alpha_k R_n(|i-k|) = R_n(i) \tag{8}$$

gdje su α_k traženi koeficijenti optimalnog prediktora kojeg želimo odrediti. Napišemo li izraz u matričnom obliku dobivamo matricu autokorelacijskih vrijednosti Toeplitzovog oblika dimenzije pxp. Autokorelacijska metoda uvijek rezultira stabilnim prediktorom. Svaki kompleksni par polova određuje jednu rezonantnu karakteristiku koja se poklapa s jednim od fomanada govornog signala.

3.3.3 Levinson-Durbinova metoda dobivanja autokorelacijskih koeficijenata

Algoritam ove metode je jednostavan, naime set slijedećih jednadžbi rješava se rekurzivno:

$$k_i = \left(R(i) - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right) / E^{(i-1)} \tag{9}$$

$$\alpha_i^{(i)} = k_i \tag{10}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \tag{11}$$

$$E^{(i)} = (1 - k_i^2) E^{(i-1)} \tag{12}$$

Tako dobivamo konačno rješenje dano u obliku $\alpha_j = \alpha_j^{(p)}, \quad 1 \leq j \leq p$

3.3.4 LSF-linijske spektralne frekvencije

U literaturi pojam poznat kao i linijski spektralni parovi (LSP) koriste se za reprezentiranje koeficijenata linearne predikcije za prijenos putem komunikacijskog kanala. LSF imaju nekoliko dobrih svojstava kao što su smanjena osjetljivost na kvantizacijske pogreške koje ih čine boljima od direktne kvantizacije linearnih predikcijskih koeficijenata. Izračun LSF-a počinje od z transformacije odaziva filtra predikcijske pogreške sa P koeficijenata:

$$A(z) = 1 - \sum_{k=1}^{P} \alpha(k) z^{-k} \tag{13}$$

Gornji polinom možemo razdvojiti na dva polinoma od kojih je jedan simetričan, $F_1(z) = A(z) + z^{-(P+1)} A(z^{-1})$, a drugi antisimetričan $F_2(z) = A(z) - z^{-(P+1)} A(z^{-1})$. Korijeni ova dva pomoćna polinoma određuju linijske spektralne frekvencije. Soong i Juang pokazali su da ako je A(Z) minimalna faza tada su ti korijeni na jediničnoj kružnici i pravilno su raspoređeni, te su LSF zapravo kutne pozicije korijena u rasponu od 0 do π. Metode određivanja LSF-a su brojne, neke od važnijih su diskretnom kosinusnom transformacijom, autokorelacijskom funkcijom koeficijenata derivacije od podfunkcija u frekvencijskoj domeni te potom izračunom spektra snage čiji su minimumi zapravo LSF koeficijenti i na kraju rekurzivnim razvojem Čebiševljevih polinoma.

4. Algoritmi za detekciju govorne aktivnosti

4.1. VAD algoritam koji koristi Itakura LPC mjeru udaljenosti

Kod ovog algoritma koristi se prosječna mjerena vrijednost spektra signala za svake od 3 signalne klase (tišinu, buku, glas). LPC udaljenosti se koriste kao mjera sličnosti testnog i referentnog uzorka. K tome se još računa i energetska udaljenost testnog i referentnog uzorka te se nelinearnom kombinacijom energetske i LPC udaljenosti dobiva konačna odluka kamo klasificirati testni signal. Blokovska reprezentacija algoritma prikazana je slikom 6. Prednosti ovog algoritma su da se sva informacija dobivena iz spektra koristi u klasifikacijskom

algoritmu i to da izračun LPC udaljenosti neuniformno koristi spektar pri ocjenjivanju sličnosti testnog i referentnog primjerka.

Opis algoritma:

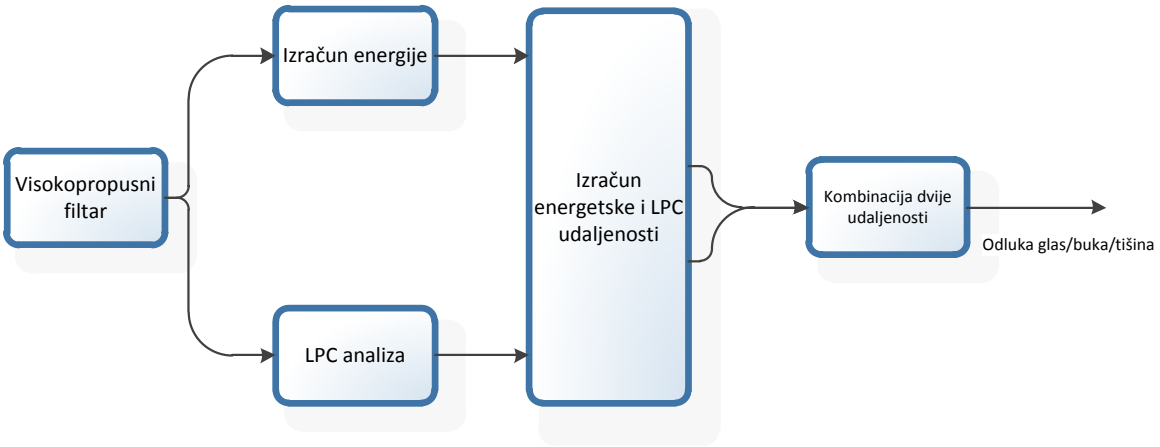
Ulazni signal uzorkuje se frekvencijom 6.67 kHz i propušta kroz visokopropusni filtar aproksimativno 200Hz da se uklone niskofrekvencijski šum ili komponente buke koje mogu biti prisutne u signalu. 8-polovska LPC analiza obavlja se svakih 15 ms (100 uzoraka) dijela signala koristeći kovarijacijsku metodu analize. Ukupno 67 analiza po sekundi se obavlja. LPC skup za i-ti okvir zapisujemo kao

$$a_i = \left(a_i(1), a_i(2), \dots, a_i(8)\right) \tag{14}$$

te logaritamsku energiju i-tog okvira

$$E_i = 10 \log_{10} \left[\sum_{n=n_0}^{n_0+149} x^2(n) \right] \tag{15}$$

Gdje je x(n) signal propusten kroz visokopropusni filter, n_0 je indeks uzorka u i-tom okviru vremena.



Slika 6. Blok dijagram algoritma 1

Energetska udaljenost računa se kao normalizirana Euklidska udaljenost oblika :

$$D_E(j) = \left| \frac{E_i - \bar{E}(j)}{\sigma_E(j)} \right| \tag{16}$$

Gdje j=1,2,3 predstavlja tišinu, bezvučni govor i zvučni govor tim redoslijedom. $\bar{E}(j)$ je prosječna logaritamska energija (pribavljena iz podataka za uvježbavanje

algoritma) j-te signalne klase, a $\sigma_E(j)$ standarnda devijacija j-te signalne klase. LPC udaljenost osnovana ja na mjeri koju je predložio Ikatura 1975. Godine, i ona je oblika:

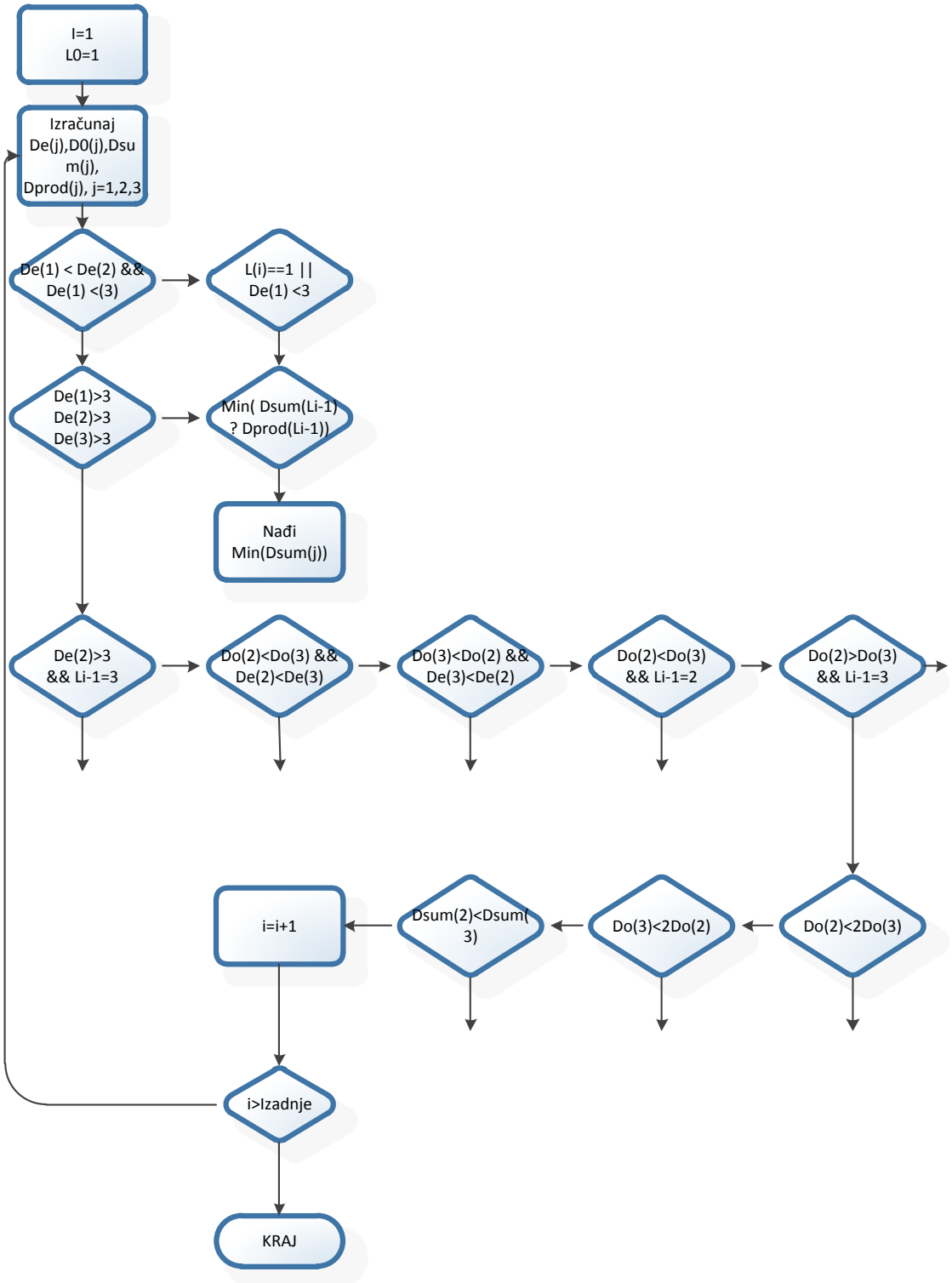
$$D_a(j) = \frac{(a-m_j)(\phi)(a-m_j)'}{(a\phi a')} \tag{17}$$

i zapravo je kovarijanca koja ocjenjuje LPC koeficijente. m_j (mean) vektor LPC koeficijenata dobiven iz podataka treniranja za j-tu signalnu klasu, dok je ϕ matrica korelacija za trenutni okvir uzorkovanja. Nazivnik $(a\phi a')$ je rezidualna pogreška LPC analize.

Na osnovu $D_E(j)$ i $D_a(j)$ te uz pomoć osnovnog seta logike algoritam stvara klasifikaciju uzorkovanog signala. Pošto je najvarijabilniji od tri signala tišina algoritam prvo odlučuje da li je signal tišina ili ne na osnovi energetskih udaljenosti i 1 okvira memorije. Ako minimum energetske distance nije tišina prema treniranim vrijednostima algoritma provjerava se da li je $D_E(j) \geq 3$ za sve j te ako jest za odluku klasifikacije koristi se prethodni okvir memorije ili kombinirane distance dane izrazima:

$$D_{sum}(j) = D_E(j) + D_a(j) \tag{18}$$

$$D_{prod}(j) = D_E(j) + D_a(j) \tag{19}$$



Slika 7. blok izračuna energetskih i LPC udaljenosti

4.2 Algoritam baziran na frekvenciji prijelaza nule i iznosu kratkotrajne energije.

Ovaj jednostavan algoritam za detekciju početka i kraja govora baziran je na dvije na dvije mjere govora: frekvenciji prijelaza nule i „kratko-vremenskoj“ energiji i može razaznati govor od pozadinske buke sa SNR-om od minimalno 30 dB. Algoritam posjeduje svojstvo koje je samo-adaptirajuće akustičnom okruženju iz kojeg preuzima sve potrebne pragove (eng. Thresholds) za kriterij detekcije početne i krajnje točke govora.

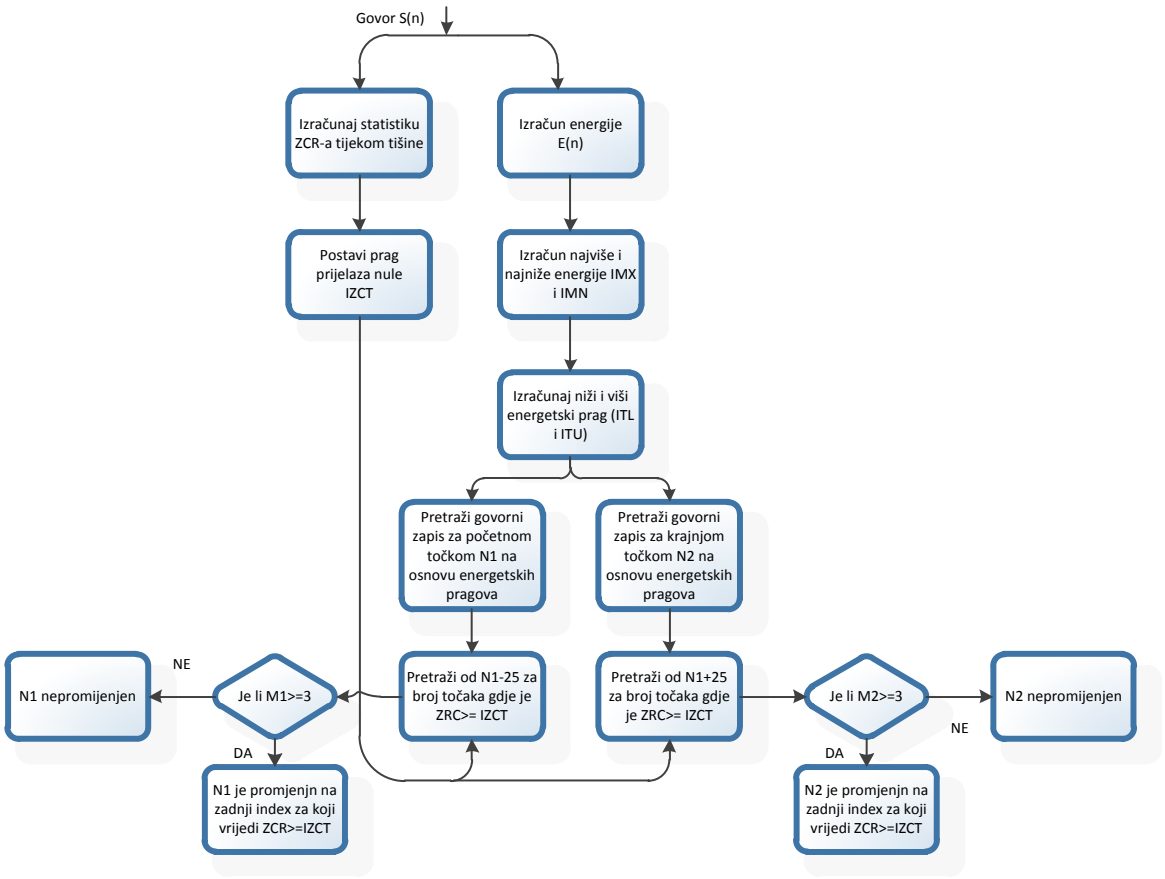
Opis algoritma:

Definiramo „kratkotrajnu energiju“ govora: E(n) –sumu amplituda u 10 ms govora centrirana na mjerenom intervalu.

$$E(n) = \sum_{i=-50}^{50} |S(n+i)| \tag{20}$$

gdje su s(n) govorni uzorci otipkani frekvencijom 10 kHz.

Frekvenciju prelaženja nula (eng. Zero crossing rate) z(n) definiramo kao broj prolaska funkcije govora kroz nulu(zadani nivo) u 10-ms intervalu. Iako je ZCR osjetljiv na 60-hz šum, dc odmak i td. U većini slučajeva je dovoljno dobra mjera za detekciju prisutnosti govora.



Slika 8. Blok dijagram algoritma 2

Slika 8 prikazuje dijagram tijeka navedenog algoritma. Ulazni signal je filtrian prije uzorkovanja (frekvencijom 10 kHz), pojasno-propusnim filtrom sa granicama 100 Hz i 4 kHz, sa prigušenjem od 48dB po oktavi. Pretpostavlja se da govor nije pristutan u prvih 100 ms zapisa jer se u to vrijeme mjeri statistika pozadinske tišine. Ta mjerenja uključuju prosječnu i standardnu devijaciju ZCR-a i prosječne energije. Ako je bilo koji od dva faktora prevelik algoritam se zaustavlja. Pragovi odnosno nivoi energije izračunavaju se po sljedećim izrazima:

$$IZCT = MIN(IF, \overline{IZC} + 2\sigma_{IZC}) \quad (21)$$

$$I1 = 0.03 * (IMX - IMN) \quad (22)$$

$$I2 = 4 * IMN \quad (23)$$

$$ITL = MIN\ (I1, I2) \quad (24)$$

$$ITU = 5 * ITL \quad (25)$$

Gdje je IZCT –prag prijelaska funkcije kroz nulu, IF-fiksni prag za bezvučni govor (25 prijelaza u 10 ms), $\overline{IZC}$ – suma srednje vrijednosti ZCR tijekom tišine, a σ_{IZC} njezina standardna devijacija. IMX izraza 2 je vršna vrijednost energije, a IMN energija za vrijeme tišine. Izraz 2 pokazuje da je I1 nivo u kojem se 3 posto vršne energije (namještene za „razinu energije tišine“), dok izraz 3 pokazuje da je I2 četiri puta veći od energije tišine . Algoritam počinje pretragu od početka intervala sve dok ne naiđe na razinu koja prelazi najniži prag. Ova točka se preliminarno označuje kao početak govora osim ako energija padne ispod ITL praga prije prelaska iznad ITU praga. Tada se nalazi nova točka kao prva sljedeća koja prijeđe ITL prag. Daljnji tijek algoritma za razliku od do sad opisanog zasniva se na frekvenciji prijelaska nule, jer su do sad određeni pragovi energije prilično grubi te ne daju dovoljno točan rezultat. S toga algoritam počinje provjeravati okolinu na prijelaze nule dosad određenih krajnjih točki govora i to u intervalu od N_1 do $N_1 - 25$ odnosno 250 ms interval prije početne točke govora. Ukoliko je IZCT odnosno broj prijelaska nule 3 ili veći nova granica govora postavlja se na prvi prijelazak nule, inače se ostavlja tamo gdje je bila i prije provjere. Na analogan način provjerava se i krajnja točka govora odnosno interval N_1 do $N_1 + 25$.

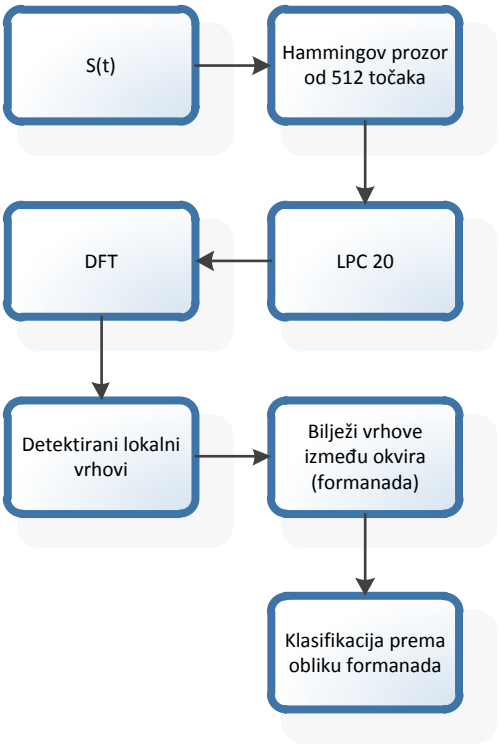
4.3 Algoritam baziran na pretraživanju oblika formanta

Sljedeći algoritam predložili su Hoyt i Wechsler 1994. godine nakon što su uočili da svi govorni signali poprimaju konveksne i konkavne oblike formanata u frekvencijskom pojasu između 400 Hz i 4KHz. Konstruirani algoritam bio je učinkovit u raspoznavanju govora od pozadinske strukturirane buke (npr. vjetar, glazba, prometna buka,..) jer nije baziran na procjeni energije signala ili autokorelacijskim metodama, što je dosadašnjim algoritmima onemogućavalo razaznavanje govora od na primjer instrumentalne glazbe.

Opis algoritma.

Ovaj algoritam analizira oblik formanata na uzorku trajanja od jedne do tri sekunde te ih klasificira na temelju konkavnosti i konveksnosti. U prvom koraku radi se zaravnavanje spektrograma da bi algoritam mogao jednostavno prepoznati formante. Primjenjuje se 20 polovski LPC na vremenski red podataka. Potom se LPC kodirana datoteka transformira diskretnom Fourierovom transformacijom (DFT). Vršne vrijednosti dobivenih podataka detektira algoritam te bilježi i

uspoređuje između vremenskih okvira pomoću algoritma lančanog koda te na temelju rezultata donosi odluku. Dijagram toka algoritma prikazan je slikom



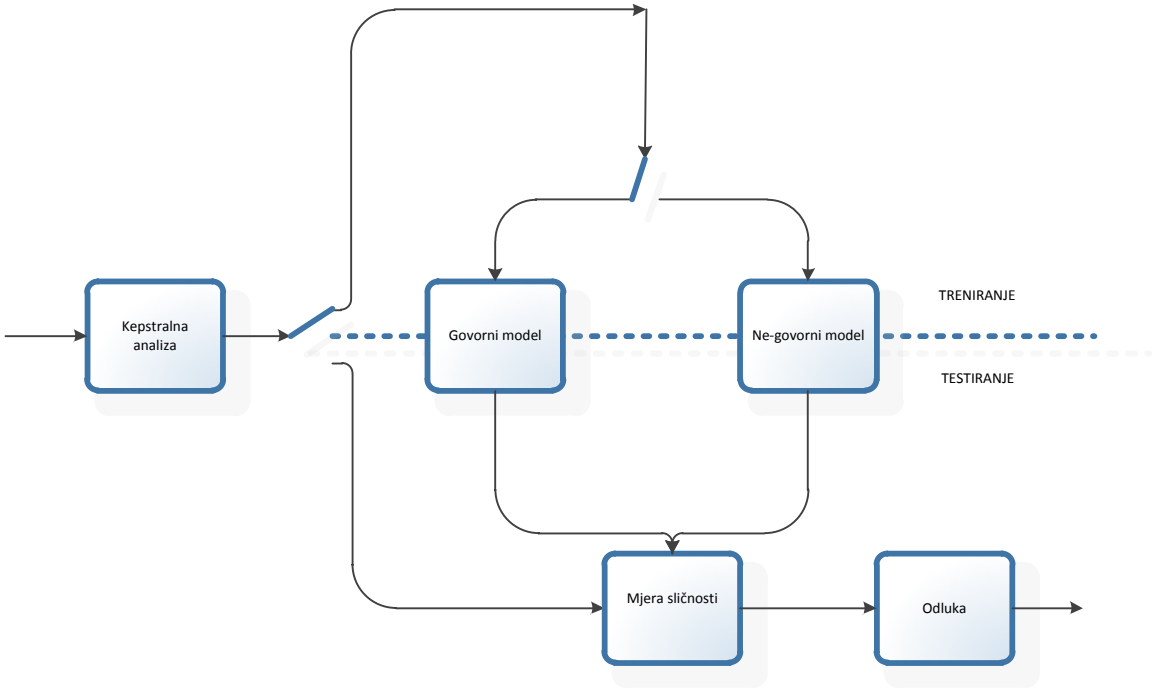
Slika 9. Blok dijagram algoritma 3

4.4 Algoritam zasnovan na kepstralnoj analizi

Ovaj algoritam baziran je na eksplicitnom (statičkom) modeliranju govora i svega što nije govor (buka, tišina etc.) , donošenje odluka radi se na svakom pristiglom („preklopljenom“) kepstralnom okviru u ovisnosti o koeficijentu sličnosti baziranom na referentnom modelu. Kepstralna analiza omogućava veliku nezavisnost tako da je svako namještanje parametara i postavljanje pragova za odlučivanje potpuno nepotrebno. Točni početci govora postignuti su čak i sa SNR od 0 db.

Opis algoritma

Algoritam možemo podijeliti na dvije pod faze: fazu treniranja govora i fazu testiranja. Prilikom izvođenja faze treniranja algoritma kreiraju se odvojeni modeli za govor i (ne govor). Ovi modeli su reprezentativni za svoju klasu.



Slika 10. Blok dijagram algoritma 4

U fazi testiranja ulazni signal obrađen kepstralnom analizom uspoređuje se s modelima okvir po okvir te se za svaki ocjenjuje razlika od treniranog modela. Pretpostavljajući da su oba modela prethodno normalizirana, jednostavni omjer dvije distorzije (iskrivljenja), prolaskom kroz fiksni prag mogu se koristiti za odluku ima li govora ili nema. Slika 10 prikazuje blok dijagram spomenutog algoritma. Dva modela iz slike 10 ne moraju sadržavati nikakvu vremensku informaciju, jer algoritam zahtjeva samo jednostavnu mjeru od svakoga okvira. Posljedica toga je korištenje vektorske kvantizacije, točnije „codebook“ pristup. Uz korištenje „codebook“ modela potrebna je eksplicitna mjera sličnosti, a najjednostavnija ujedno i najuspješnija je vektorska kvantizacija distorzije iz standardne euklidske distance :

$$d = \sum_{i=1}^{P} (C_i - C_i')^2 \qquad (26)$$

Gdje je p red kepstralne analize, a C_i i C_i' i-ti elementi dva kepstralna vektora.

4.5 Algoritam za detekciju govora uz adaptivno modeliranje buke

Kao što je već spomenuto, svaki izvor buke možemo promatrati kao aditivni, odnosno nadodan na korisni signal. Ovaj algoritam koristi adaptivno regresivno modeliranje buke kako bi umanjio utjecaj signala buke te uz to koristi uspoređivanje spektralne gustoće buke sa spektralnom gustoćom govornog signala. FIR (eng. finite impulse response) filtri korišteni u autoregresivnoj analizi (AR) ugađaju se LMS (najmanji srednji kvadrat) algoritmom tijekom intervala kad nema govornog signala. Prema dobivenim eksperimentalnim rezultatima algoritam uspješno detektira kraj i početak govora na vrlo niskim SNR omjerima, pa čak i kad je amplituda buke veća od amplitude govora do -6 SNR.

Opis algoritma

1. Autoregresivna (AR) analiza

Pretpostavimo da se buka n(i) može opisati AR procesom reda M tako da zadovoljava sljedeću jednadžbu:

$$H_A(z)N(z) = w(z) \tag{27}$$

Gdje su $N(z)$ i $w(z)$ z transformacije buke i procesa bijele buke dok je $H_A(z)$ autoregresivni filtar definiran izrazom :

$$H_A(z) = 1 + \sum_{k=1}^{M} a_k z^{-k} \tag{28}$$

Ukoliko je buka donekle stacionarna, njen autoregresivni filtar $H_A(z)$ procjenjuje se u ne-govornim intervalima i može se koristiti za povećanje energetskog razmaka (odstojanja) između buke i bučnog govornog signala. Kako je govor intrinzički ne statičan signal i ima komponente u razmatranom pojasu (250 Hz-3.2KHz), njegova spektralna gustoća je sve više i više različita od one koju ima buka, čak iako je je buka korelirana i koncentrirana u niskim frekvencijama (ispod 1KHz). Zbog toga je atenuacija uzrokovana $H_A(z)$-om će biti u prosjeku manja za govorni signal nego za signal buke. $H_A(z)$ je transverzalan ili ima konačan impulsni odaziv te se njegovi koeficijenti mogu odrediti korištenjem klasičnog LMS algoritma. Ako se koeficijenti a_k zamijene sa c_i gdje je $c_i = -a_i$ adaptacija težine dana je izrazom:

$$C_k(i+1) = C_k(i) + \eta n(i-k)e(i) \quad (29)$$

Gdje je η koeficijent učenja, $e(i)$ predikcija pogreške dana izrazom:

$$e(i) = n(i) - \sum_{k=1}^{M} c_k n(i-k) \quad (30)$$

2. Uspoređivanje gustoće spektra

Spektralna estimacija se radi sa 14-kanalnom Mel-filtarskom bankom, bez primjene logaritamske kompresije i normalizacije. Spektralna gustoća (SD(i)) za okvir vremena i određuje se izrazom u euklidskoj metrici:

$$SD(i) = 20 \cdot \log\left(\frac{\sqrt{\sum_{k=1}^{14}\left(E_k^n - E_{i,k}\right)^2}}{\sqrt{\Sigma_{k=1}^{14}\left(E_k^n\right)^2}}\right) \quad (31)$$

Gdje $E_k^n, E_{i,k}$ označavaju izlazne energije Mel- filtra. Spektralna estimacija buke izračunava se iz spektra unutar 10 ne-govornih okvira.

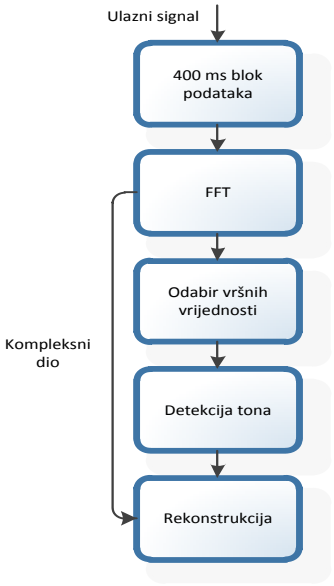
4.6 Algoritam za detekciju govora koji koristi mjere periodičnosti

Ovaj algoritam kao svoj mehanizam odluke „govor - ne govor“ koristi određivanje periodičnosti najmanjeg kvadrata. (eng. LSPE least-square periodicity estimate). Cilj algoritma nije pronaći točne granice početka i kraja govora te je zbog toga najpodobniji za korištenje u aplikacijama koje snimaju govor za kasniji pregled te imaju manju potrebu za visokom preciznošću detekcije početka i kraja govora. Lažno okidanje zbog periodičnosti nekih ne govornih signala kod ovog pristupa je dosta često te je potrebno pretprocesirati signal i postprocesirati dobivene rezultate. Algoritam je sposoban detektirati govor u jako niskim SNR omjerima, od 0 do -5db. LSPE izračun koristi nepreklapajuće 25 ms okvire na pojasu signala od 200-1000 HZ. Uski pojas se koristi da se minimalizira vjerojatnost unutar-pojasnog (eng. inband) interferirajućeg signala, a opet da se postigne učinkovita detekcija periodičnosti. Signal se uzorkuje dva puta na frekvenciji od 4kHz da bi se postigla dodatna rezolucija detekcije periodičnosti. U algoritam je uključen i energetski detektor da bi se izbjegla detekcija signala jako niskih amplituda u prisutnosti signala veće amplitude.

Opis algoritma

1. preprocesiranje.

Preprocesor implementiran za algoritam mora biti u mogućnosti detektirati i ako je to moguće, ukloniti sve očekivane tipove smetnji. U opisanom primjeru detektor uklanja sve podatke i tonove sa konzistentnim pojavljivanjem vrhova u konačnom vremenu. Pošto govor također ima konstantno pojavljivanje vrhova potrebna je harmonijska pretraga kako se slučajno ne bi uklonio govor. Blok dijagram algoritma prikazan je slikom 11.



Slika 11. Blok dijagram algoritma 11

Ulazni signal se formira u 400ms djelove, svaki dio dijeli se u 25 okvira dugih 32 ms sa preklapanjem 16ms. Svaki okvir prolazi kroz Hammingov otvor. Koristeći 400 ms dijelove formira se 25 FFT okvira (brzom Fourierovom transformacijom rezolucije 31.25 Hz sa dva izlaza, kompleksnim zapisom transformacije i logaritamske amplitude kompleksne transformacije) i 16 LSPE okvira da budu procesuirani u istom intervalu. Odabir vršnih vrijednosti radi se nad svakim FFT okvirom, neovisno o susjednim okvirima, radi se na temelju jednog od izlaza FFT-a (logaritma amplitude kompleksne transformacije podataka). Svaka vršna vrijednost sadrži u okolini 5 vrijenosti frekvencije koje zadovoljavaju sljedeće izraze:

$$p_{i-2} < p_{i-1} < p_i \tag{32}$$

$$p_{i+2} < p_{i+1} < p_i \tag{33}$$

Detektor tona pregledava 3 uzastopna 400ms bloka i označava sve frekvencijske točke koje su vršne vrijednosti u više od pet okvira svakog bloka. Sve takve vrhove preprocesor treba izbaciti, dok preostale označene frekvencije mogle bi biti ili interferencija ili vršna vrijednost govornog signala. Kako bi se napravila razlika između njih radi se harmonijska pretraga bloka, sa vršnim vrijednostima najmanje udaljenim 400 Hz. Ako se harmonici sa sličnim razmakom nađu dva put unutar tri uzastopna okvira, označena frekvencija se smatra govorom, ostali se smatraju interferencijom. Podaci bez interferencije se rekonstruiraju tako da se sve FFT točke pronađene interferencije postave u nulu.

2. LSPE izračun.

Izračunavanje LSPE izbacuje vrijednost za svaki 25 ms okvir ulaznog signala s(i). Nek je

$$s(i) = s_0(i) + n(i) \tag{34}$$

Za i = 1, 2, ..., N(N=100 za 25 ms okvir) gdje je $s_0(i)$ periodična komponenta ulaznog signala, $n(i)$ njegova neperiodična komponenta. $s_0(i) = s_0(i + kP_0)$ za cjelobrojni višekratnik perioda P_0. Neka je $\hat{P}_0$ estimacija P_0 i $\hat{s}_0(i, \hat{P}_0)$ estimacija periodične komponente koja se dobiva iz izraza

$$\hat{s}_0(i) \equiv \sum_{h=0}^{k_0} \frac{s(i+I_n\hat{P}_0)}{k_0} \tag{35}$$

Cilj metode najmanjih kvadrata je naći period vršne vrijednosti $\hat{P}_0$ koji minimizira srednju kvadratičnu pogrešku ($\sum_{i=1}^{N}\left[S(i) - \hat{S}_0(i)\right]^2$) za svaki analizirani okvir.

Da bi se to postiglo koristi se normalizirana mjera periodičnosti (Friedman).

$$R_1(\hat{P}_0) = \frac{I_0(\hat{P}_0) - I_1(\hat{P}_0)}{\sum_{i=1}^{n} s^2(i) - I_1(\hat{P}_0)} \tag{36}$$

Gdje je $I_1(\hat{P}_0) = \sum_{i=1}^{P}\sum_{h=0}^{K_0} \frac{s(i+h\hat{P}_0)^2}{K_0}$ i $\dot{I}_0(\hat{P}_0) = \sum_{i=1}^{N} \hat{S}_0^2(i)$ za svaki 25 ms okvir.

Periodičnosti oko 0.5 obično su izračunate iz bijele buke pa ih detektor odvaja i

postavlja negativne vrijednosti na nula. Potom sumira periodičnosti šest uzastopnih okvira i daje optimalnu performansu. Prag od 0.46 daje lažno okidanje u buci jednom u svakih 10 minuta što je zapravo dobar kompromis između osjetljivosti i točnosti za većinu aplikacija.

3. SSQE postprocesor

U prvom koraku pretražujemo $R_1(\hat{P}_0)$ za sve bitne vršne vrijednosti. Za govorni signal vršne vrijednosti će biti harmonički raspoređene sa frekvencijom koja se nalazi unutar normalnog govornog tonskog pojasa od 11-56 uzoraka(71 Hz- 363 Hz). Za interferirajuće signale vršne vrijednosti će biti jako blizu jedna drugoj, dok za buku vršne vrijednosti neće biti harmonički raspoređene. Iz navedenog vidimo da bi bilo korisno izvući dva parametra sumu svih vrhova koji su u harmonijskom odnosu sa govornim tonskim frekvencijama s_v, a drugi parametar r_{wn} je omjer ove sume ovog sume i sume svih drugih vrhova. Iz ova dva parametra možemo izračunati mjeru vjerojatnosti pojavljivanja govora $p_v = ws_v$ gdje je *w* težinski faktor dan izrazima:

$$w = 0, r_{wn} \leq 1 \tag{37}$$

$$w = 0.66(r_{wn} - 1), 1 \leq r_{wn} \leq 2.5 \tag{38}$$

$$w = 1, r_{wn} \leq 2.5 \tag{39}$$

Maksimalna vrijednost w_{max} koja je mjera stabilnosti tona preko 4 uzastopna okvira uzeta u kombinaciji sa p_v daje SSQE (eng. sum squared error) izlaz iz postprocesora.

4.7 Algoritam za detekciju govora baziran na raspoznavanju uzoraka

Arhitektura ovog algoritma je zasnovana na raspoznavanju uzoraka, točnije sastoji se od pretprocesoskog dijela, dijela za izvlačenje bitnih parametara, dijela za spajanje (matching phase), postprocesorskog dijela i bloka za odlučivanje. Specifičnost ovog algoritma nalazi se unutar faze za spajanje gdje se koristi poseban alat za učenje FuGeNeSys. Riječ je o alatu koji koristi neuronske mreže, genetske algoritme i fuzzy logiku. Navedene metodologije koje spadaju u područje umjetne inteligencije nazivaju se „Soft Computing“ i imaju puno bolje performanse nego tradicionalne metode.

Opis algoritma

Pretproceiranje govornog signala sastoji se od 140 Hz visokopropusnog filtra kako bi se uklonila neželjena niskofrekvencijska komponenta. Kako bi se zajamčilo robusno prepoznavanje granica riječi u prisutnosti visoke razine buke, koriste se 4 parametra za klasifikaciju buka/govor izračunata iz 10-ms okvira sa sempliranjem od 8kHz i to su: puno-pojasna energijska razlika ΔE_f, niskopojasna energetska razlika, razlika prijelaza nule ΔZC i spektralna distorzija ΔS. Isti parametri koriste se i u G729 standardu, i u algoritmu koji je implementiran na Blackfin okruženju. Za fazu ugađanja koristi se set od šest fuzzy pravila koja se automatski ekstrahiraju pomoću hibridnog programskog paketa za učenje FuGeNeSys. Kao primjer navodim 6 pravila dobivenih iz baze podataka za uhodavanje algoritma koja se sastoji od 342 minute čistih govornih signala i bučnih sekvenci.

Pravilo 1 *: IF (DS is medium-low) THEN (Y is active)*

Pravilo 2 : *IF (DEf is very high) THEN (Y is inactive)*

Pravilo 3 : *IF (DEl is low) AND (DS is very low) AND (DZC is high) THEN (Y is active)*

Pravilo 4: *IF (DEl is low) AND (DS is high) AND (DZC is medium) THEN (Y is active)*

Pravilo 5 : *IF (DEl is high) AND (DS is very low) AND (DZC is low) THEN (Y is active)*

Pravilo 6 : *IF (DEI is high) AND (DS is not low) AND (DZC is very high) THEN (Y is active)*

Fuzzy sustav ima ulogu mapiranja uzorka ulaznih parametara u skalarnoj vrijednosti između 0 i 1, koja indiciraju razinu pripadnosti području unutar klasifikacija govor/ne govor.

Kako bi se smanjile oštre varijacije izlaza iz fuzzy sustava, izlaz se post procesira sa „median“ filtrom 7. Reda. Na kraju unutra bloka za odlučivanje, pomoću uspoređivanja određenih pragova izbacuje se krajnja i početna točka govora.

Kako bi se izbjeglo lažno okidanje zbog nestacionarne buke, mogućih pauza unutar riječi ili bezglasnih zvukova. Koristi se kontrolni mehanizam koji se sastoji od dva pomična prozora w_1i w_2 jednkih širina. Da bi se odredila početna točka govora, jedan od pomičnih prozora prolazi kroz postprocesirani signal. Ako prvi broj elemenata veći od odabranog praga T jest veći od izraza k*w_1 tada je početak riječi dobar. Analogan postupak ponavlja se za kraj riječi, a koristi se prozor w_2. k je eksperimentalno dobivena bezdimenzionalna konstanta vrijednosti 0.8.

8. Algoritam baziran na višestrukim statističkim metodama

Istraživanja su pokazala da se DFT koeficijenti čistog govora i buke mogu efektvnije opisati pomoću funkcija gustoće vjerojatnosti kao što su Gaussova, Gamma i Laplaceova distribucija. Svaki parametarski model evaluiran sa Kolmogorov-Smirnov GOF testom (eng. Goodnes Of Fit) koji mjeri koliko su udaljeni pretpostavljeni modeli od empirijske distribucije. Pokazalo se da su Gamma i Laplaceova distribucija najprikladnije za modeliranje buke u govornom signalu.

Opis algoritama :

Statistički modeli za govorne signale

Pretpostavimo da je signal buke n dodan na govorni signal s, sa njihovom sumom koja je x. Dane su dvije hipoteze H_0i H_1 koje redom predstavljaju nepostojanje odnosno postojanje govornog signala.

$$H_0 \text{... X(t) = N(t)} \tag{40}$$

$$H_0 \text{... X(t) = N(t)+S(t)} \tag{41}$$

Gdje su X(t)=.....; N(t)=.....; S(t)..... redom DFT koeficijenti za okvir t govorni signal s bukom, buka i čisti govor. Gausova funkcija gustoće spektralnih komponenata za bučni signal predočene su izrazima:

$$P_G(X_k|H_0) = \frac{1}{\pi\lambda_{n,k}} \exp\left\{-\frac{x_k^2}{\lambda_{n,k}}\right\} \tag{42}$$

$$P_G(X_k|H_1) = \frac{1}{\pi[\lambda_{n,k}+\lambda_{s,k}]} \exp\left\{-\frac{|x_k^2|}{[\lambda_{n,k}+\lambda_{s,k}]}\right\} \tag{43}$$

Druga distribucija je Laplaceova. Realni i imagirnarni dijelovi svakog koeficijenta se distribuiraju prema laplaceovoj funkciji gustoće vjerojatnosti. Neka su $X_{k(R)}$ i $X_{k(I)}$ redom realni i imaginarni dijelovi DFT koeficijenta X_k. Ako su oba realni i imaginarni dijelovi iste varijance njihove distribucije su dane sa

$$P_L\left(X_{k(R)}\right) = \frac{1}{\sigma_x} \exp\left\{-\frac{2|X_{k(R)}|}{\sigma_x}\right\} \tag{44}$$

$$P_L\left(X_{k(I)}\right) = \frac{1}{\sigma_x}\exp\left\{-\frac{2|X_{k(I)}}{\sigma_x}\right\} \tag{45}$$

Gdje je σ_x varijanca od X_k.

Uvjetne vjerojatnosti dane su izrazima:

$$P_L(x_k|H_0) = \frac{1}{\lambda_{n,k}}\exp\left\{-\frac{2(|x_{k(R)}|+|x_{k(I)}|)}{\sqrt{\lambda_{n,k}}}\right\} \tag{46}$$

$$P_L(X_k|H_1) = \frac{1}{\lambda_{n,k}+\lambda_{S,k}}\exp\left\{-\frac{2(|X_{k(R)}|+|X_{k(I)}|)}{\sqrt{\lambda_{n,k}+\lambda_{S,k}}}\right\} \tag{47}$$

Zadnji model baziran je na gamma kompleksnoj distribuciji koja se može opisati sljedećim izrazima:

$$p_m\left(X_{k(R)}\right) = \left(\frac{\sqrt{6}}{8\pi\cdot\sigma_x|x_{k(R)}|}\right)^{0.5}\exp\left\{-\frac{\sqrt{3}|X_{k(R)}|}{\sqrt{2}\sigma_x}\right\} \tag{48}$$

$$p_m\left(X_{k(R)}\right) = \left(\frac{\sqrt{6}}{8\pi\cdot\sigma_x|x_{k(I)}|}\right)^{0.5}\exp\left\{-\frac{\sqrt{3}|X_{k(I)}|}{\sqrt{2}\sigma_x}\right\} \tag{49}$$

Uz primjenu dvaju hipoteza dobivamo izraze:

$$p_M(x_k|H_0) = \frac{\sqrt{6}}{8\pi\sqrt{\lambda_{\mathrm{n,K}}}|x_{k(R)}|^{0.5}|x_{k(1)}|^{0.5}}\exp\left\{-\frac{\sqrt{3}(|x_{k(n)}|+|x_{k(I)}|)}{\sqrt{2}\sqrt{\lambda_{\mathrm{S},k}}}\right\} \tag{50}$$

$$p_M(x_k|H_1) = \frac{\sqrt{6}}{8\pi\sqrt{\lambda_{n_F}+\lambda_{S,k}}|x_{k(R)}|^{0.5}|x_{k(1)}|^{0.5}}\exp\left\{-\frac{\sqrt{3}(|x_{k(n)}|+|x_{k(I)}|)}{\sqrt{2}\sqrt{\lambda_{\mathrm{S},k}+\lambda_{S,k}}}\right\} \tag{51}$$

GOF testovi:

Za uspješno detektiranje govora moramo odabrati model koji daje najbolje rezultate uz spektar govora sa aditivnom bukom. S toga se izvršava KS test sa uvjetovanim H0 i H1 hipotezama. Prednost ovog statističkog testa je ta da distribucija testa po sebi ne ovisi o kumulativnoj distribuciji funkcije koja se testira te da je iznimno precizan dok primjerice hi kvadrat test ovisi o veličini uzorkovane aproksimacije da bi bio validan. Govorni materijal od 64 sekunde trajanja od 4 muška i 4 ženska govornika, koristi se zajedno sa bijelom bukom, bukom vozila i žamorom govora.

Trapezoidalni prozor duljine 13 ms primjenjuje se na ulazni signal svakih 10 ms. Svaka rečenica je uzorkovana frekvencijom od 8kHz. Svaki okvir prozora signala se potom transformira kroz 128 brzu Fourierovu transformaciju (FFT) nakon (zero padding). Od sakupljenih podataka, srednja vrijednost uzoraka i varijanca je izračunata te je potom primijenjena parametarska distribucija. Neka je DFT koeficijent izračunat iz bučnog signala koji se sastoji od svih frekvecijskih linija koje su jednako razmaknute (eng. Frequency bins) Potom KS test usproeđuje empirijski dobivenu distribuciju funckije F koja se definira formulom:

$$F_x(z) = \begin{cases} 0, z < x_{(1)} \\ \frac{n}{N}, x_{(N)} \leq z < x_{(n+1)}, n = 1,2, \dots, N-1 \\ 1, z \geq x_{(N)} \end{cases} \quad (52)$$

Gdje je $X_{(n)}, n = 1, \dots, N$ red statistike za podatak **X**. Kako bi izračunali red statistike spritramo elemente od **X** tako da je $X_{(1)}$ najmanji, a $X_{(n)}$ najveći. Udaljenost između empirijskog modela i određenih distribucija računa se po formuli:

$$T(x) = \max_i |F_x(x_i) - F(x_i)| \quad (53)$$

Većina provedenih pokusa i provedenih KS testova pokazuje da je dobro primjeniti različite statističeke modele za DFT spektar govora s bukom ovisno o tipovima buke i SNR razinama.

Omjer vjerojatnosti (LRT eng. Likehood ratio test) za VAD

Za detekciju govora baziranu na prethodno opisanim statističkim modelima potrebno je definirati statistiku po kojoj se odlučuje za svaku liniju FFT jednake udaljenosti prema izrazu:

$$\Lambda_k \equiv \frac{p(x_k|H_1)}{p(x_k|H_0)} \quad (54)$$

U kojem Λ_k pretstavlja LR za k-ti frekvencijsku rezoluciju (frequency bin). Pravilo odlučivanja za VAD se izračunava kao geometrijska srednja vrijednost omjera vjerojatnosti (likehood ratio). tako da je

$$\log \Lambda = \frac{1}{M} \sum_{k=0}^{M-1} \log \Lambda_k <> \text{Ni} \quad (55)$$

pri čemu je ***M*** ukupan broj frekvencijskih linija koje su jednako razmaknute a Ni prag detekcije.

Određivanje SNR-a na temelju najveće vjerojatnosti(ML), estimacije upravljane odlučivanjem(DD) i prediktivne estimacije (PD)

DD pristup estimacije apriori SNR-a osniva se na heurističkoj detekciji. DD esetimator se također koristi kod algoritama za poboljšavanje govora (eng. speech enhancement). DD metoda opisana izrazom:

$$\hat{\xi}^{DD}(t) = \alpha \frac{|\hat{x}_k(t-1)|^2}{\hat{\lambda}_{n,k}(t-1)} + (1-\alpha)u[\hat{\gamma}_k(t) - 1] \tag{56}$$

omogućuje bolju estimaciju apriori SNR-a nego primjerice ML metoda opisana izrazom $\hat{\xi}_k^{ML} = \gamma_k - 1$ te posljedično smanjuje neđeljene fluktuacije povišenog LR tijekom perioda u kojima se ne govori. Međutim nedostak ove metode je kašnjenje kod perioda u kojima se govori, točnije a priori SNR se može protumačiti kao aposteriori SNR tijekom tih perioda. S toga da bi izbjegli nedostatke DD metode koristimo PD metodu baziranu na soft-decision metodi za a priori estimaciju SNR-a izraženu formulom:

$$\hat{\lambda}_{s,k}(t+1) = \zeta \hat{\lambda}_{s,k}(t) - (1-\zeta_s)E[\delta_k(t)^2 x_k(t)] \tag{57}$$

5. Odabrani algoritam za detekciju govorne aktivnosti

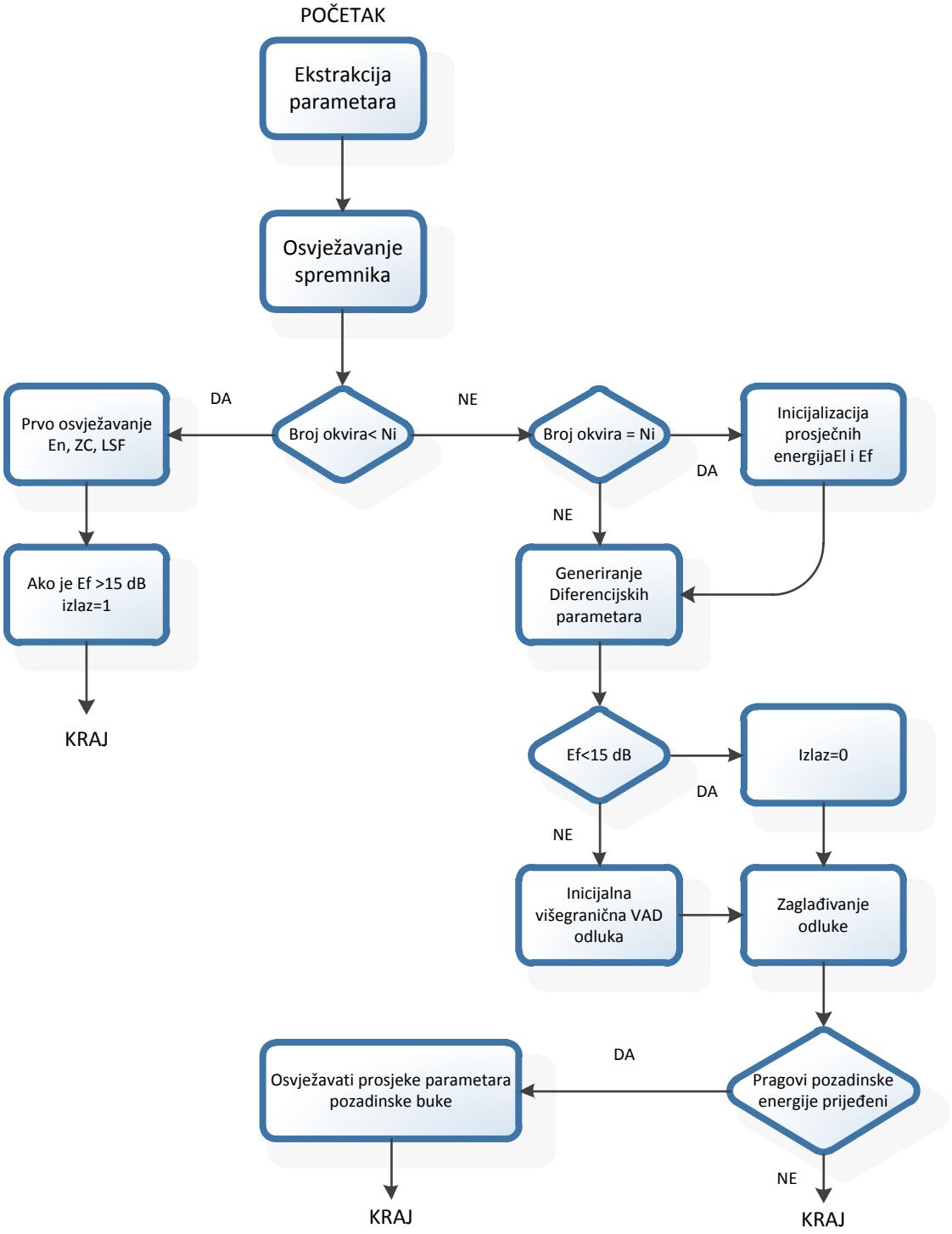
Zbog mogućnosti jednostavne implementacije na dsp-u odabran je algoritam koji je dio G.729 standarda, točnije dodatak B. G.729 definirala je ITU-T (internacionalna telekomunikacijska unija- Telekomunikacijski sektor za standarde) radna grupa (1993-1996), a dodatak b uključuje detektor govorne aktivnosti (VAD) i generator utješne buke(eng. Comfort noise generator) i segment za diskontinuirani prijenos (DTX). Ovi algoritmi se koriste da bi se smanjila količina podataka koje treba prenjeti komunikacijskim sustavom ukoliko nastupi dio komunikacije gdje se ne govori. VAD komponenta donosi odluku govor-ne govor svakih 10 ms u odnosu na veličinu okvira G.729 govornog kodera. Set diferencijskih parametara se izvlači iz ulaznih podataka i dobiva se odluka. Diferencijski parametri su puno-pojasna energija, niskopojasna energija, frekvencija prijelaza nule i spektralno izobličenje. Dugoročni prosjeci parametara tijekom segmenata govorne neaktivnost slijede promjenu pozadinske buke te se uspoređuju s trenutnim parametrima danog okvira. Inicijalna odluka se donosi pomoću djelomične linearne granice odluke između svakog para diferencijalnih parametara. Zaključna odluka o govornoj aktovnisti dobiva se nakon zaglađivanja inicijalne odluke.

Opis algoritma:

U prvom djelu algoritma četri parametra se izvlače iz ulaznog signala. Samo izvlačenje parametara se djeli sa koderom aktivnog govora i neaktivnog govora da bi se postigla veća učinkovitost. Ukoliko broj ovira manji od N_i inicijalizira se stadij za dugoročno procjenjivanje enegrije i detekcija govorne aktivnosti na izlazu daje 1 ako je okvir energije sa LPC analize iznad 15 dB. Inače detekcija govora daje 0 na izlazu. Ako je broj okvira jednak N_i inicijalizira se stadij za karakteristične enegije pozadinske buke. U sljedećem dijelu algoritma izračunava se set diferencijacijskih parametara. Ovaj set se definira kao razlika mjerenja između parametara trenutnig okvira i prosjeka karakteristike pozadinske buke.

Četri diferencijske mjere koje se računaju su: spektralna distorzija, potpuna energijska razlika, niskopojasna energijska razlika i razlika prijelaza nule.

Inicijalna odluka govorne aktivnosti se donosi u sljedećem koraku, koristeći višestruko ograničene predjele odluke u prostoru sa 4 diferencijske mjere. Odluka o govornoj aktivnosti se donosi kao unija svih četiri prostora diferencijskih mjera, dok se odluka o neaktivnosti donosi kao komplement prethodnoj logičnoj odluci. Razmatranje enegije zajedno sa prethodnim odlukama koristi se za „zaglađivanje olduke“.



1. Ekstrakcija parametara.

Za svaki okvir iz ulaznog govornog signala izvači se set od 4 parametra koji su zapravo autokorelacijski koeficijenti koji se zapisuju kao:

$$\{R(\dot{I})\}_{i=0}^{q} \tag{58}$$

q je u ovom slučaju 12.

Prvi parametar koji se izvlači je spektralna distorzija izračunata iz LSF (eng. Line Spectral Frequencies). Set linearnih prediktivnih koeficijenata se izračunava iz autokorelacije i set iz {LSFi}p i=1 gdje je p=10, je izračunat iz =>pogledati opis. Sljedeći parametar je cjelopojasna E_f energija, koja logaritam normaliziranog prvog autokorelacijskog koeficijenta, a računa se po formuli:

$$E_f = 10\log_{10}\left[\frac{1}{N}R(0)\right] \tag{59}$$

gdje je N= 240 veličina prozora LPC analize u govornim uzorcima.

Niskopojasna energija E_l mjerena od 0 do frekvencije F računa se po formuli:

$$E_l = 10\log_{10}\left[\frac{1}{N}h^T Rh\right] \tag{60}$$

Gdje je H impulsni odaziv FIR filtra sa cutoff frekvencijom na F[Hz], R Toeplitzova autokorelacijska matrica sa autokorelacijskim koeficijentima na svakoj diagonali.

Posljednji parametar, frekvencija prijelaza nule za svaki okvir računa se po formuli:

$$z_C = \frac{1}{2M}\sum_{i=0}^{M-1}[|\mathrm{sgn}[x(i)] - \mathrm{sgn}[x(i-1)]|] \tag{61}$$

Inicijalizacija prosjeka

Za prvih Ni okvira spektralni parametri pozdainske buke, označeni kao $\{\overline{LSF_i}\}_{i=1}^{P}$ se postavljaju kao prosijek $\{LSF_i\}_{i=1}^{P}$ vrijednosti za te okvire. Prosijek prijelaza nule pozadinske buke zapisujemo kao $\overline{ZC}$ dok $\overline{E_n}$ definira prosjek pune energije E_f za Ni okvira. Navedeni klizni prosijeci sadrže samo one okvire kojima je energija E veća od 15 db, a postupak same inicijalizacije izgleda ovako:

ako $\overline{E_n}$ *< 671088640 (T1) tad je*

E_f = $\overline{E_n}$

E_l = $\overline{E_n}$ *-53687091*

inače, ako je 671088640<$\overline{E_n}$*<738197504*

E_f = $\overline{E_n}$ *-67108864*

E_l = $\overline{E_n}$ *-93952410*

inače,

E_f = $\overline{E_n}$ *-134217728*

E_l = $\overline{E_n}$ *-161061274*

Generiranje parametra "dugotrajne" minimalne energije

Dugotrajna minimalna energija računa se kao minimum E_f na N0 predhodnih okvira. Kako je N0 relativno velik minimalna enrgija se računa koristeći spremljene vrijednosti E_f-a

Generiranje diferencijskih parametara

Četiri diferencijska parametra generiju se iz vrijednosti dobivenih izvlačenjem iz trenutnog okvira i kliznih prosjeka pozadinske buke. Prvi parametar je spektralna distorzija ΔS koja se dobiva kao suma kvadrata razlike između vrijednosti $\{LSF_i\}_{i=1}^{P}$ i klizećeg prosijeka spektra pozadinske buke $\{\overline{LSF_i}\}_{i=1}^{P}$ .To možemo zapisati izrazom:

$$\Delta S = \sum_{i=1}^{P}(LSF_i - \overline{LSF_i})^2 \tag{62}$$

Zatim imamo diferencijski parametar puno-pojasne energije, koa se izvodi kao razlika između energije trenutnog okvira i one klizećeg prosijeka:

$$\Delta E_f = \overline{E_f} - E_f \tag{63}$$

Analogno tome imamo i izraze za diferencijske parametre prijelaska nule i nisko-pojasne energije:

$$\Delta E_l = \overline{E_l} - E_l \tag{64}$$

$$\Delta ZC = \overline{ZC} - ZC \tag{65}$$

Prvotno, višegranično donošenje odluke detekcije govora

Odluke za detekciju govorne aktivnosti donose se na temelju četrnaest pravila unutar četverodimenzionalnog prostora odluke za četiri parametra svaka dimenzija su slijedeća:

1) ako DS > a_1 ×DZC + b_1 then I_{VD} = 1
2) ako DS > a_2 ×DZC + b_2 onda I_{VD} = 1
3) ako DE_f < a_3 ×DZC + b_3 onda I_{VD} = 1
4) ako DE_f < a_4 × DZC + b_4 onda I_{VD} = 1
5) ako DE_f < b_5 onda I_{VD} = 1
6) ako DE_f < a_6 ×DS + b_6 onda I_{VD} = 1
7) ako DS > b_7 onda I_{VD} = 1
8) ako DE_I < a_8 × DZC + b_8 onda I_{VD} = 1
9) ako DE_I < a_9 × DZC + b_9 onda I_{VD} = 1
10) ako DE_I < b_{10} onda I_{VD} = 1
11) ako DE_I < a_{11} × DS + b_{11} onda I_{VD} = 1
12) ako DE_I > a_{12} × DE_f + b_{12} onda I_{VD} = 1
13) ako DE_I < a_{13} × DE_f + b_{13} onda I_{VD} = 1
14) ako DE_I < a_{14} × DE_f + b_{14} onda I_{VD} = 1

Gdje je I_{VD} zastavica odnosno marker detekcije govora, a a_i i b_i konstante dane u tablici:

a_1	23488	b_1	28521
a_2	-30504	b_2	19446
a_3	-32768	b_3	-32768
a_4	26214	b_4	-19661
a_5	0	b_5	-30802
a_6	28160	b_6	-19661
a_7	0	b_7	30199
a_8	16384	b_8	-22938
a_9	-19065	b_9	-31576
a_{10}	0	b_{10}	-17367
a_{11}	22400	b_{11}	-27034
a_{12}	30427	b_{12}	29959
a_{13}	-24576	b_{13}	-29491
a_{14}	23406	b_{14}	-28087

Zaglađivanje odluke

Inicijalnu odluku potrebno je zagladiti da bi se pokazala dugotrajna stacinarna priroda govornog signala. Zaglađivanje odluke događa se u četri stupnja. Zastavica koja indicira da se dogodio takozvani hangover prije pokretanja VAD-a postavlja se na nulu. Hangover je fiksni vremenski interval kod kodera govora kojeg VAD postavlja kad detektira pad govorne amlitude prije nego što prestane slati pakete odnosno kodirati govor. Prvi nivo zaglađivanja odluke radi se prema slijedećem izrazu:

ako je (IVD = 0) i (SVD1 = 1) i (E > E f + T3) tad SVD0 = 1 and v_ flag = 1

gdje su SVD0, SVD1, SVD2 redom zaglađene odluke trenutnog okvira, prethodnog i jednog prije prethodnog.

Druga razina zaglađivanja odvija se prema slijedećem izrazu:

```
ako je (Fvd=1) i (IVD=0) i (SVD1=1) i (SVD2=1) i (abs(Ef-E-1)<T4)tad{
SVD0=1;
V_flag=1;
Ce=Ce+1;
        ako(Ce<N1)tad{
                FVD1=1
        } inače{
                FVD1=0;
                Ce=0;
        }
}
```

Gdje je Ce brojač zaglađivanja, a FVD1 parametar zaglađivanja 2. stupnja.

Treći stupanj sadrži brojač kontinuiteta buke Cs, koji se na početku inicijalizira na 0, a inkrementira se ukoliko je SVD0=0. Također vrijedi pravilo:

```
ako je (SVD0=1) i (Cs>N2) i (Ef-E-1<=T5) tada {
 SVD0=0;
Cs=0;
}
```

Za četvrti stupanj, donosi se konačna odlukao detekciji govorne aktivnosti ako je zadovoljen slijedeći uvijet:

ako ($E_f < \overline{E_f} + 40265318$) i (brojac okvira>N0) i (v_flag=0) tad SVD=0;

Osvježavanje parametara pozadinske buke

Nakon konačnog zaglađivanja odluke za trenutni okvir po potrebi se osvježavaju svi parametri pozadinske buke. U ovom dijelu VAD algoritma testira se slijedeće pravilo :

ako *($E_f < \overline{E_f} + 40265318$) tada osvježi parametre*

Klizeći prosjeci pozadinske buke osvježavaju se pomoću autoregresivnog postupka prvog reda. Različiti autoregresivni koeficijenti koriste se za različite parametre pogotovo pri detekciji velike promjene karakteristike pozadinske buke. Neka je β_{E_f} autoregresivni koeficijent za osvježavanje $\overline{E_f}$, kao i β_{E_l} za $\overline{E_l}$, β_{ZC} za $\overline{ZC}$ te β_{LSF} za $\{\overline{LSF_i}\}_{i=1}^{P}$. Ukupan broj okvira u kojem je uvijet za osvježavanje okvira zadovoljen broji se sa Cn te se stoga koeficijenti β_{E_l}, β_{E_f},β_{ZC} i β_{LSF} određuju prema tom brojaču. Klizeći prosijeci osvježavaju se prema izrazima:

$$\overline{E_f} = \beta_{E_f} * \overline{E_f} + \left(1 - \beta_{E_f}\right) * E_f \quad (66)$$

$$\overline{E_l} = \beta_{E_l} * \overline{E_l} + \left(1 - \beta_{E_l}\right) * E_l \quad (67)$$

$$\overline{ZC} = \beta_{ZC} * \overline{ZC} + (1 - \beta_{ZC}) * ZC \quad (68)$$

$$\overline{LSF_i} = \beta_{LSF} * \overline{LSF_i} + (1 - \beta_{LSF}) * LSF_i \quad (69)$$

Implementacija u matlabu

Matlab kod koji implementira opisan algoritam izvorno je napravio P. Kabal te je za potrebe ovog rada nadograđen. Navedeni matlab kod samo aproksimira rad odabranog algoritma radi smanjenog broja kalkulacija koje procesor treba provesti. Radi bolje detekcije dodano je filtriranje signala, te je je sve skupa prilagođeno za brzu pretvorbu matlab koda u embedded c/c++ pomoću matlabovih emlc i emlmex funkcija.

Slijedi detaljan opis koda kojim je implementiran dani algoritam.

```
global VAD_memory;
clear all;
clc;
[x,fs] = wavread('C:zavrsni.wav');
Ns = length(x);
NsFrame = 80;
NsLA = 40;
NsWin = 240;
NFrame = floor(Ns/NsFrame);
x = [x; zeros(NsFrame, 1)];
HPFilt.b = [1 -1];
HPFilt.a = [1 -127/128];
x_hp = filter(HPFilt.b, HPFilt.a, x);
VADPar = InitVADPar;
x_hp_mem = zeros(NsLA, 1);
```

Prvi dio koda, učitava se zvučna datoteka postavlja se memorija, veličina prozora i parametri za visokopropusni filter te parametri za detekciju govora.

```
[bh,ah]=cheby2(ORD,R,fd/(fs/2),'high');
[bl,al]=cheby2(ORD,R,fg/(fs/2));
b=conv(bh,bl);
a=conv(ah,al);
hilo=filter(b,a,x_hp);
```

Visoko i nisko propusna filtracija za uklanjanje izmjeničnih smetnji napajanja.

```
for (k = 1:NFrame-1)
  ist = k*NsFrame + 1;
  ifn = ist + NsFrame - 1;
  x_new = x(ist:ifn);
  [Ivd, VADPar] = VAD(x_new, VADPar);
  x_hp_buffer = [x_hp_mem; hilo(ist:ifn)];
  x_hp_curr = x_hp_buffer(1:NsFrame);
  x_hp_mem = x_hp_buffer(end-NsLA+1:end);
  VAD_memory(k*NsFrame)=Ivd;
  Ef_memory(k*NsFrame)=VADPar.MeanSE;
  El_memory(k*NsFrame)=VADPar.MeanSLE;
  ZC_memory(k*NsFrame)=VADPar.MeanSZC;
  %LSF_memory(a,k*NsFrame)=VADPar.MeanLSF;
```

Glavna for petlja u kojoj se vrti VAD algoritam za svaki pojedini dio govora.

```
function [Ivd, VADPar, v_flag] = VAD (x_new, VADPar)
VAD_APPENDIX_II = 0;
N = VADPar.N;
N0 = VADPar.N0;
Ni = VADPar.Ni;
INIT_COUNT = VADPar.INIT_COUNT;
NOISE = 0;
VOICE = 1;
v_flag = 0;
VADPar.FrmCount = VADPar.FrmCount + 1;
frm_count = VADPar.FrmCount;
[x_new_hp, VADPar.HPFilt.Mem] = filter(VADPar.HPFilt.b, VADPar.HPFilt.a, ...
                                       32768 * x_new, VADPar.HPFilt.Mem);
xwin = [VADPar.Wmem; x_new_hp];
[r, LSF, rc2] = VADLPAnalysis(xwin, VADPar);
Ef = 10*log10(r(1) / N);
Elow = r(1) * VADPar.LBF_CORR(1) ...
       + 2 * sum(r(2:end) .* VADPar.LBF_CORR(2:end));
El = 10*log10(Elow / N);
SD = sum((LSF-VADPar.MeanLSF).^2);
ist = VADPar.N - VADPar.LA - VADPar.NF + 1;     % Current frame start
ifn = ist + VADPar.NF - 1;                      % Current frame end
ZC = zcr(xwin(ist:ifn+1));
VADPar.Next_MinE = min(Ef, VADPar.Next_MinE);
MinE = min(VADPar.Prev_MinE, VADPar.Next_MinE);
```

VAD funkcija, postavljaju se lokalne varijable, povećava se brojač okvira i radi se još jedna filtracija te se filtrirani signal sprema u memoriju. U funkciji VADLPAnalysis provodi se analiza linearne predikcije te se kao izlaz dobivaju autokorelacijski koeficijenti. Na temelju dobivenih koeficijenata računaju se četiri osnovna parametra, s time da je ZCR normalizirana.

```
if (frm_count <= Ni)
  if (Ef < 21)
    VADPar.less_count = VADPar.less_count + 1;
    marker = NOISE;
  else
    marker = VOICE;
    NEp = (frm_count - 1) - VADPar.less_count;
    NE = NEp + 1;
    VADPar.MeanE = (VADPar.MeanE * NEp + Ef) / NE;
    VADPar.MeanSZC = (VADPar.MeanSZC * NEp + ZC) / NE;
    VADPar.MeanLSF = (VADPar.MeanLSF * NEp + LSF) / NE;
  end

end
```

Inicijalizacija klizećih prosjeka, za srednju energiju, srednji LSF te srednji ZCR.

```
if (frm_count >= Ni)
  if (frm_count == Ni)
    if (VAD_APPENDIX_II)
      if (VADPar.less_count >= Ni)     %
        VADPar.FrmCount = 0;
        frm_count = VADPar.FrmCount;
        VADPar.less_count = 0;
      end
    end
    VADPar.MeanSE = VADPar.MeanE - 10;
    VADPar.MeanSLE = VADPar.MeanE - 12;
  end
  dSE = VADPar.MeanSE - Ef;
  dSLE = VADPar.MeanSLE - El;
  dSZC = VADPar.MeanSZC - ZC;
```

Inicijalizacija energija i generiranje diferencijskih parametrara.

```
  if (Ef < 21)
    marker = NOISE;
  else
    marker = MakeDec(dSLE, dSE, SD, dSZC);
  end

  if (VAD_APPENDIX_II)
    if (marker == VOICE)              % modified for Appendix II
      VADPar.count_inert = 0;
    end

    if (marker == NOISE && VADPar.count_inert < 6)
      VADPar.count_inert = VADPar.count_inert + 1;
      marker = VOICE;
    end
  else
    v_flag = 0;
  end
```

Inicijalno donošenje odluke, Energetski prag postavljen je na 21, umjesto 15 jer kako se pokazuje algoritam u slučajevima nižeg SNR djeluje bolje ako mu se prag blago povisi.

```
if (VADPar.PrevMarkers(1) == VOICE && marker == NOISE ...
    && Ef > VADPar.MeanSE + 2 && Ef > 21)
  marker = VOICE;
  if (~VAD_APPENDIX_II)
    v_flag = 1;
  end
end
if (VADPar.flag == 1)
  if (VADPar.PrevMarkers(2) == VOICE ...
      && VADPar.PrevMarkers(1) == VOICE ...
      && marker == NOISE ...
      && abs(Ef - VADPar.PrevEnergy) <= 3)
    VADPar.count_ext = VADPar.count_ext + 1;
    marker = VOICE;
    if(~ VAD_APPENDIX_II)
      v_flag = 1;
    end

    if (VADPar.count_ext <= 4)
      VADPar.flag = 1;
    else
      VADPar.flag = 0;
      VADPar.count_ext = 0;
    end
  end
else
  VADPar.flag = 1;
```

Koraci I i II zaglađivanja odluke o detekciji govorne aktivnosti.

```
if (marker == VOICE && VADPar.count_sil > 10 ...
     && Ef - VADPar.PrevEnergy <= 3)
   marker = NOISE;
   VADPar.count_sil = 0;
   if (VAD_APPENDIX_II)
      VADPar.count_inert = 6;  % modified for AppendixII
   end
 end

 if (marker == VOICE)
   VADPar.count_sil = 0;
 end
 if (~VAD_APPENDIX_II)
   if (Ef < VADPar.MeanSE + 3 && VADPar.FrmCount > N0 ...
       && v_flag == 0 && rc2 < 0.6)
     marker = NOISE;
   end
 end
```

Koraci III i IV zaglađivanja odluke.

```
% Update mean LSF, SE, SLE, SZC
    VADPar.MeanLSF = COEFSD * VADPar.MeanLSF + (1-COEFSD) * LSF;
    VADPar.MeanSE = COEF * VADPar.MeanSE + (1-COEF) * Ef;
    VADPar.MeanSLE = COEF * VADPar.MeanSLE + (1-COEF) * El;
    VADPar.MeanSZC = COEFZC * VADPar.MeanSZC + (1-COEFZC) * ZC;
  end

  if (frm_count > N0 && ...
        (VADPar.MeanSE < MinE && SD < 0.002532959) ...
          || VADPar.MeanSE > MinE + 10 )
    VADPar.MeanSE = MinE;
    VADPar.count_update = 0;
  end
end

VADPar.PrevEnergy = Ef;
VADPar.PrevMarkers = [marker, VADPar.PrevMarkers(1)];

ist = VADPar.NF + 1;
VADPar.Wmem = xwin(ist:end);

Ivd = marker;

return
```

Osvježavanje parametara i vraćanje konačne odluke u glavni program i spremanje odluke za potrebe odlučivanja sljedećeg okvira.

```
function dec = MakeDec(dSLE, dSE, SD, dSZC)
a = [0.00175, -0.004545455, -25, 20, 0, ...
     8800, 0, 25, -29.09091, 0, ...
     14000, 0.928571, -1.5, 0.714285];
b = [0.00085, 0.001159091, -5, -6, -4.7, ...
     -12.2, 0.0009, -7.0, -4.8182, -5.3, ...
     -15.5, 1.14285, -9, -2.1428571];
dec = 0;
if SD > a(1)*dSZC+b(1)
    dec = 1;
    return;
end
if SD > a(2)*dSZC+b(2)
    dec = 1;
    return;
end
if dSE < a(3)*dSZC+b(3)
    dec = 1;
    return;
```

Funkcija koja donosi osnovnu odluku na temelju četrnaest pravila za pragove svakog od četiri parametra. Kompletan kod funkcije nije prikazan pošto su pravila poznata iz opisa algoritma.

```
function [r, LSF, rc2] = VADLPAnalysis (x, VADPar)

M = VADPar.M;     % LP order
NP = VADPar.NP;   % autocorrelation order

% Apply window to input frame
xw = VADPar.Window .* x;

% Compute autocorrelation
r = acorr(xw, NP+1) .* VADPar.LagWindow;

% Compute normalized LSF
A = ac2poly(r(1:M+1));
LSF = poly2lsf(A) / (2 * pi);     % normalized to 0 to 0.5

% Reflection coefficients
rc = ac2rc(r(1:3));
rc2 = rc(2);

return
```

Funkcija koja obavlja LP analizu, odnosno izračunava LSF iz autokorelacijskih koeficijenata koji se normalizira na vrijednosti od 0 do 0.5 umjesto od 0 do 2π.

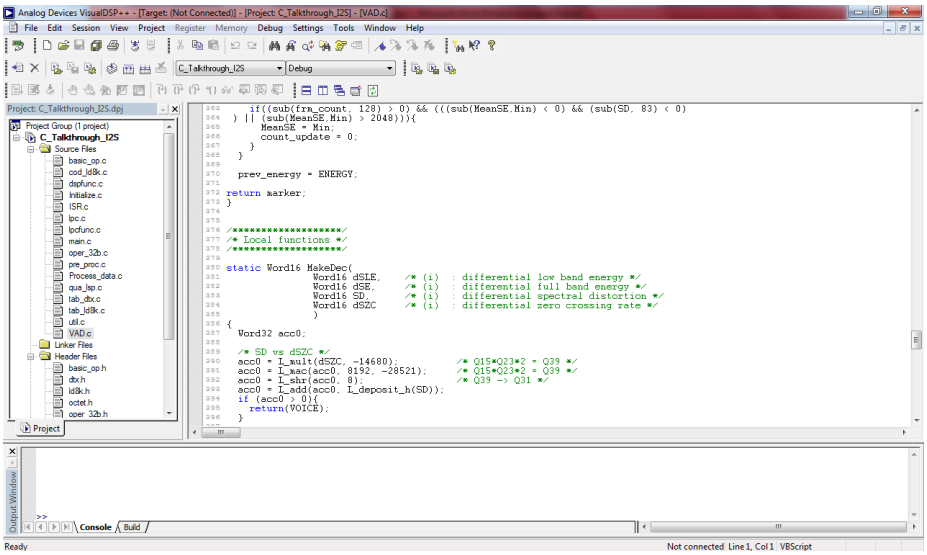
```
function rxx = acorr (x, Nt)

Nx = length (x);
N = Nt;
if (Nt > Nx)
  N = Nx;
end

rxx = zeros(Nt, 1);
for (i = 0:N-1)
  Nv = Nx - i;
  rxx(i+1) = x(1:Nv)' * x(i+1:i+Nv);
end

return
```

Funkcija koja Levinson Durbinovim postupkom računa LPC koeficijente.

Algoritam implementiran na Analog Devices bf537 ez kit lite

Platforma na kojoj je implementiran odabrani algoritam detekcije govora je Blackfin razvojna ploča BF 537 EZ-Lite Kit tvrtke Analog devices. Ploča sadrži ADSP-537 Blackfin procesor čiji je maksimalni takt 600 MHz, brzina vanjske sabirnice 133 Mhz, 64 Mb SDRAM-a, 4 Mb Flash programabilne memorije, analogno audio sučelje koje se sastoji od AD1871 96 kHz analog-to-digital kodeka, AD1854 96 kHz digital-to-analog kodeka (DAC), ulaznog i izlaznog stereo priključka, Ethernet sučelja 10 Mbits/s, JTAG sučelja, LE dioda i 5 tipki od kojih je jedna reset, kao i nekoliko priključnica za daughterboard ploče.

Razvojno okruženje koje se koristi, VISUAL DSP++ je također od tvrtke Analog Devices, te se isporučuje sa razvojnom pločicom, a omogućuje razvoj programske podrške u programskim jezicima ANSI C i C++.



Za implementaciju algoritma korištene su neke gotove biblioteke i c datoteke koje simuliraju dvostruku preciznost pomoću cjelobrojnih vrijednosti (eng. fixed point notation) kao na primjer "lpc.c" koja omogućuje izračunavanja autokorelacijskih koeficijenata, "basic_op32.c" koja omogućuje osnovne operacije poput množenja, zbrajanja, logaritmiranja i djeljenja. Implementacija, čiji opis slijedi, malo je izmijenjena od originalne što se osobito odnosi na zaglađivanje odluke.

C izvorni kod

Izvorni kod algoritma raspodijeljen je u nekoliko datoteka čija je funkcionalnost dana sljedećom tablicom:

basic_op.c	Matematičke operacije i operacije nad bitovima
cod_ld8k.c	Definicija osnovnih parametara i invokacija VAD funkcije
dspfunc.c	Operacije korjenovanja, kvadriranja i logaritmiranja po bazi 2
Initialize.c	Inicijalizacija prekidnih rutina potrebnih za snimanje
ISR.c	Invokacija funkcije za procesiranje podataka
lpc.c	Izračun LPC i LSF koeficijenata
lpcfunc.c	Opreacije nad LPC i LSF koeficijentima
main.c	Glavni dio programa-while(1) petlja
oper_32b.c	Matematičke opreacije dvostruke preciznosti
pre_proc.c	filtriranje signala
Process_data.c	koristan posao sa primljenim okvirom podataka
qua_lsp.c	Kopiranje LSF vektora
tab_dtx.c	Postavljanje VAD cjelobrojnih konstanti
tab_ld8k.c	Postavljanje cjelobrojnih konstanti za LSF i LPC
util.c	Funkcije kopiranja vektora i postavljanja vektora na nulu
VAD.c	VAD funkcija
basic_op.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
dtx.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
ld8k.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
octet.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
oper_32b.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
sid.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
tab_dtx.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
tab_ld8k.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
Talktrough.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke

typedef.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke
vad.h	Globalne konstante , prototipovi operatora i funkcija za istoimene .c datoteke

Neke od važnijih funkcija opisane su u sljedećim poglavljima.

VAD.c

Ovo je glavni algoritam koji u potpunosti prati onaj napisan u matlabu.Funkcija nije napisana u cjelosti nego su izdvojeni neki njeni dijelovi. Ostatak funkcije nalazi se na CD-u priloženom uz završni rad. Prototip funkcije je sljedeći:

```
bool vad(
        Word16 rc,//refleksivni koeficijent
        Word16 *lsf, //pokazivač na vektor LSF-a
        Word16 *r_h, //viši bitovi koeficijenata
        Word16 *r_l, //niži bitovi koeficijenata
        Word16 exp_R0,//eksponent
        Word16 *sigpp,//preprocesirani signal
        Word16 frm_count,//brojač okvira
        Word16 prev_marker,//prethodna detekcija
        Word16 pprev_marker,//detekcija prije prethodne
        Word16 *marker) //povratni parametar, VAD odluka
```

Sljedeći dio koda prikazuje izračun osnovnih parametara. Logaritam po bazi dva koristi se isključivo zbog fixed point izračuna koji se radi po bazi dva.

```
  acc0 = L_Comp(r_h[0], r_l[0]);
  Log2(acc0, &exp, &frac);
  acc0 = Mpy_32_16(exp, frac, 9864);
  i = sub(exp_R0, 1);
  i = sub(i, 1);
  acc0 = L_mac(acc0, 9864, i);
  acc0 = L_shl(acc0, 11);
```

```
ENERGY = extract_h(acc0);
ENERGY = sub(ENERGY, 4875);

/* Izračun niskopojasne energije*/
acc0 = 0;
for (i=1; i<=NP; i++)
  acc0 = L_mac(acc0, r_h[i], lbf_corr[i]);
acc0 = L_shl(acc0, 1);
acc0 = L_mac(acc0, r_h[0], lbf_corr[0]);
Log2(acc0, &exp, &frac);
acc0 = Mpy_32_16(exp, frac, 9864);
i = sub(exp_R0, 1);
i = sub(i, 1);
acc0 = L_mac(acc0, 9864, i);
acc0 = L_shl(acc0, 11);
ENERGY_low = extract_h(acc0);
ENERGY_low = sub(ENERGY_low, 4875);

/* izračun SD */
acc0 = 0;
for (i=0; i<M; i++){
  j = sub(lsf[i], MeanLSF[i]);
  acc0 = L_mac(acc0, j, j);
}
SD = extract_h(acc0);      /* Q15 */

/* Broj prijelaza nule */
ZC = 0;
for (i=ZC_START+1; i<=ZC_END; i++)
  if (mult(sigpp[i-1], sigpp[i]) < 0){
```

```
      ZC = add(ZC, 410);      /* Q15 */
    }
/* Inicijalizacija i osvjezavanje minimuma */

if((frm_count & 0x0007) == 0){
      i = sub(shr(frm_count,3),1);
      Min_buffer[i] = Min;
      Min = MAX_16;
      Prev_Min = Min_buffer[0];
          for (i=1; i<16; i++){
               if (sub(Min_buffer[i], Prev_Min) < 0){
                    Prev_Min = Min_buffer[i];
               }
          }
          Min = Prev_Min;
          Next_Min = MAX_16;
          if (sub(ENERGY, Min) < 0){
          Min = ENERGY;
          }
          if (sub(ENERGY, Next_Min) < 0){
          Next_Min = ENERGY;
          }
          if((frm_count & 0x0007) == 0){
               for (i=0; i<15; i++)
                    Min_buffer[i] = Min_buffer[i+1];
               Min_buffer[15] = Next_Min;
               Prev_Min = Min_buffer[0];
               for (i=1; i<16; i++)
                    if (sub(Min_buffer[i], Prev_Min) < 0){
                         Prev_Min = Min_buffer[i];
```

```
                    }
                }
            }
        /*klizni prosjeci*/
         if (sub(frm_count, INIT_FRAME) <= 0){
           if(sub(ENERGY, 3072) < 0){
             *marker = NOISE;
             less_count++;
           }
           else{
             *marker = VOICE;
             acc0 = L_deposit_h(MeanE);
             acc0 = L_mac(acc0, ENERGY, 1024);
             MeanE = extract_h(acc0);
             acc0 = L_deposit_h(MeanSZC);
             acc0 = L_mac(acc0, ZC, 1024);
             MeanSZC = extract_h(acc0);
             for (i=0; i<M; i++){
               acc0 = L_deposit_h(MeanLSF[i]);
               acc0 = L_mac(acc0, lsf[i], 1024);
               MeanLSF[i] = extract_h(acc0);
             }
           }
         }
```

Prototip funkcije za osnovnu četverodimenzionalnu odluku:

```
static Word16 MakeDec(

    Word16 dSLE,      // (i)   : differential low band energy
    Word16 dSE,       // (i)   : differential full band ener
    Word16 SD,    // (i)   : differential spectral distortion
    Word16 dSZC       /* (i)   : differential zero crossing
    rate */ )

{

  Word32 acc0;


  /* SD vs dSZC */

  acc0 = L_mult(dSZC, -14680);          /* Q15*Q23*2 = Q39 */

  acc0 = L_mac(acc0, 8192, -28521);     /* Q15*Q23*2 = Q39 */

  acc0 = L_shr(acc0, 8);                /* Q39 -> Q31 */

  acc0 = L_add(acc0, L_deposit_h(SD));

  if (acc0 > 0){

    return(VOICE);

  }
```

Funkcija "MakeDec" nije napisana u cijelosti jer je njeno ponašanje definirano preko već spomenutih četrnaest pravila.

Main.c

Glavna funkcija koja poziva algoritam detekcije, a napravljena je tako da učitava .wav datoteku koju smo snimili na računalu te potom ide okvir po okvir i donosi odluku. Prvotni pokušaj ostvarivanja funkcije bazirao se na real time obradi no zbog kompleksnosti algoritma to nije bilo ostvarivo te je stoga implementiran na sljedeći način:

```
#include "Talkthrough.h"
#include <sysreg.h>
#include <ccblkfn.h>

#include <stdio.h>
#include <adi_ssl_init.h>
#include "math.h"
#include "typedef.h"
#include <stdlib.h>
#include "basic_op.h"
#include "octet.h"
short int *zapis;
char *polje_odluka;
#include <time.h>
int i, duljina;

struct TIPzaglavlje {

  long          RIFF;
  char          NI1 [18];
  unsigned int Kanal;
  long          Frekvencija;
  char          NI2 [6];
  char          BitRes;
  char          NI3 [12];

} Zaglavlje;
```

U prvom dijelu definiramo zaglavlja potrebna za ostvarivanje funkcionalnosti programa, kao i strukture podataka potrebne za čitanje Wave datoteka.

```
void main(){
     FILE *stream;
     adi_ssl_Init();
     int  frekvencija,i=0;
     stream = fopen("nino8.wav", "r");
     if (stream == NULL){
        printf("nemogu ucitat");
        fflush (stdout);
     }else {
```

```
            printf("ucitano");
            fflush(stdout);
        }

        fseek(stream, 0L, SEEK_END);
        duljina=ftell(stream)-48;
        fseek(stream, 0L, SEEK_SET);
        printf("velicina datoteke jest: %d", duljina);
        fflush(stdout);
        zapis=(short int *)malloc(sizeof(short int)*duljina);
        polje_odluka = (char *)malloc(sizeof(char)*duljina)
       if (duljina>32000) {
          if (duljina > 64000) {
             duljina = 64000;
             free(zapis);
           zapis = (short int *)malloc(sizeof(short
int)*duljina);
                  if (!zapis) {
                  return;
                  }
            }
        }

        fread(&Header, 46, 1, stream);
        if (Zaglavlje.RIFF != 0x46464952) {
        printf ("Nije wav datoteka");
       }
        if (Header.Channels != 1) {
       //printf ("Nije mono ");
       }
       if (Header.BitRes != 16){
       printf ("Nije 16 bitna datoteka")
      frekvencija= Header.Frequency;
```

U ovom dijelu učitavamo Wav datoteku u instancu tipa podataka zaglavlje te provjeravamo da li je wav datoteka mono ili stereo i kolika je bitna rezolucija.

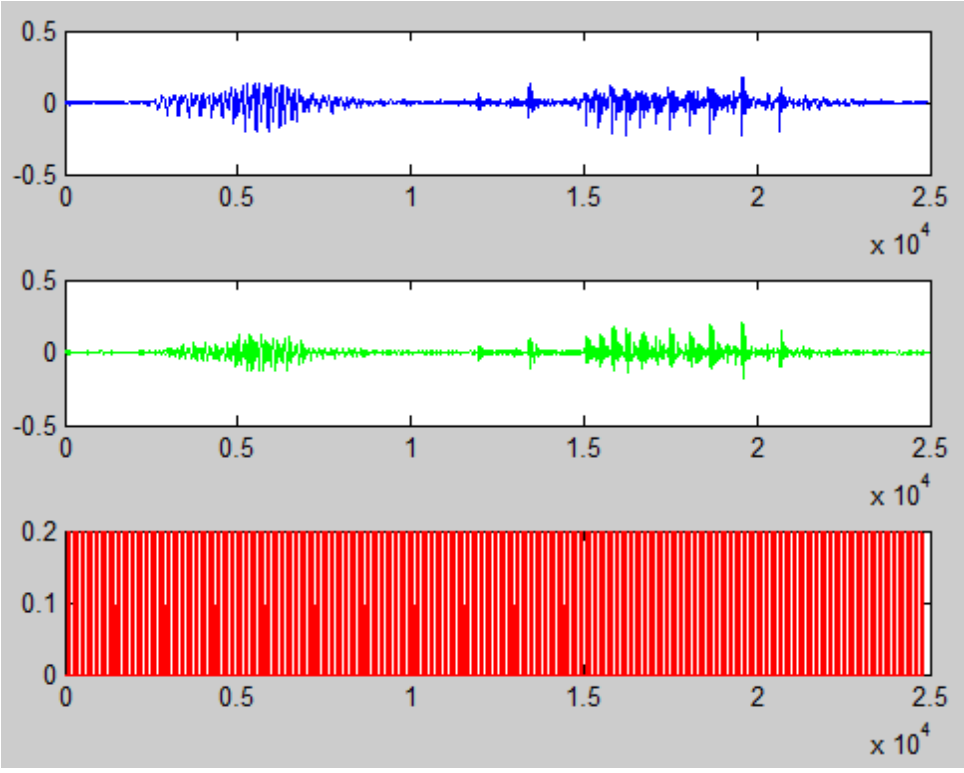
```
        Init_Pre_Process();
        Word16 govorcek;
        bool odluka;
        int velicina;
        Init_Coder_ld8k();
        while(1){
            govorcek=*(zapis+i);
            Pre_Process(govorcek, 4);
            odluka=Coder_ld8k(1,1,i,1);
            *(polje_odluka+i)=odluka;
```

```
fflush(stdout);
//iChannel0LeftOut = *(zapis+i)*100
i+=80;

if(i>duljina){
  printf("dosao do kraja%d\n", i),
        exit(-1);
}}}
```

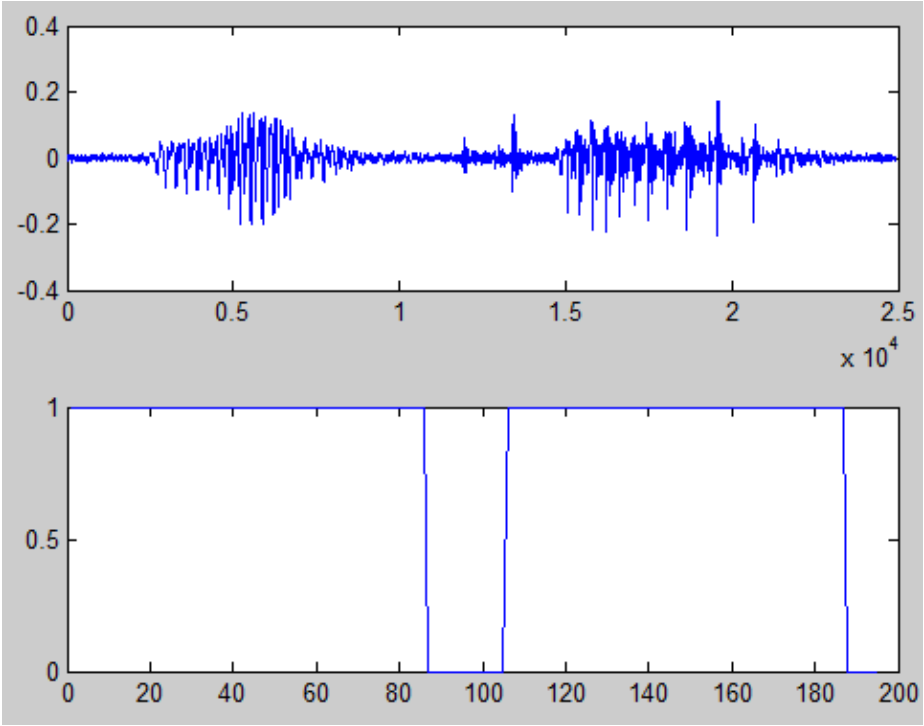
U ovom dijelu pozivamo algoritam detekcije govora i zapisujemo rezultat u polje olduke. Po potrebi možemo pustiiti zapis datoteke kroz lijevi kanal da čujemo zapis na izlazu makete.

Rezultati algoritma

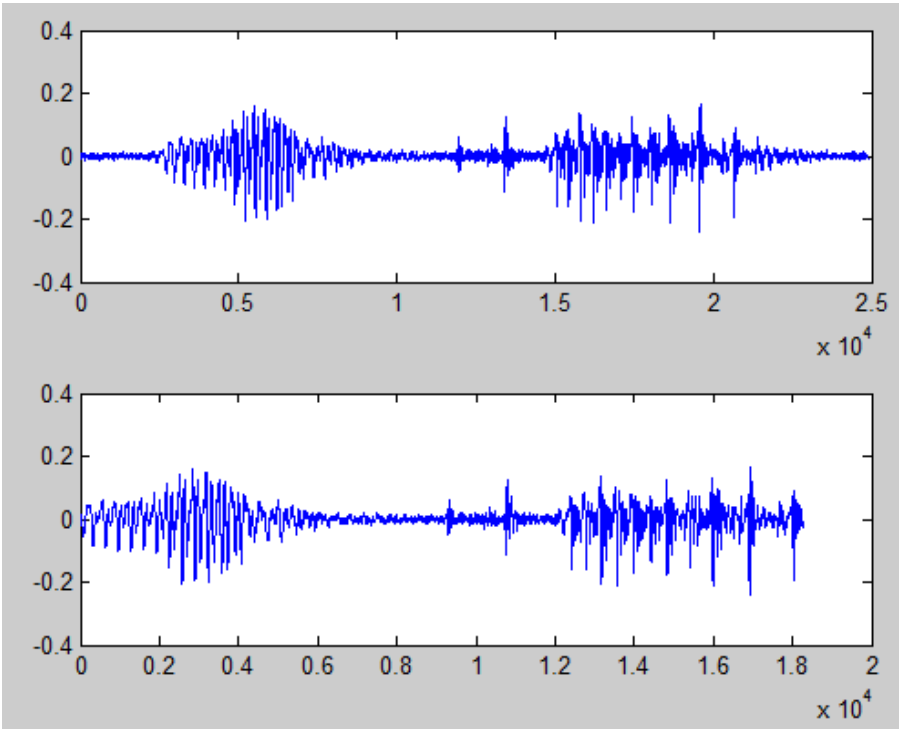
U ovom poglavlju dani su rezultati implementiranog algoritma u matlabu i na dsp procesoru kao i neke usporedbe modernih algoritama sa G.729 dodatak b. Algoritam implementiran u matlabu ima jako loše rezultate pri niskim do srednjim SNR-ovima te se u većini slučajeva događa lažno okidanje odnosno prepoznavanje buke kao govora.



Na slici vidimo da je govor detektiran u gotovo svim okvirima govornog signala koji je prethodno filtriran kako bi se uklonio šum gradske mreže.

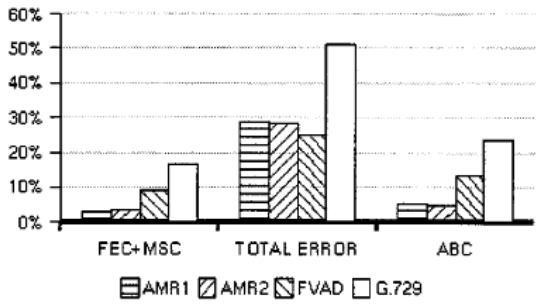


Isti zvučni zapis obrađen je statističkim algoritmom kojeg predlažu Sohn i Sung daje malo preciznije rezultate, no također ima lažno okidanje jer u srednjem segmentu ne prepoznaje bezvučni glas k.



Najblolje rezultate daje drugi algoritam koji je opisan u ovom radu, baziran na prijelazima nule i kratkotrajne energije. Kao što možemo vidjeti na slici algoritam izrezuje sve dijelove koje je detektirao kao tišinu što je napravio iznimno točno.

U radu kojeg su objavili F. Beritelli, S. Casale, G. Ruggeri, and S. Serrano napravljena je usporedba Adaptive MultiRate (AMR) detektora govora, detektora govora G.729.b i detektora baziranog na Fuzzy logici. Procjenjivani parametri (FEC-front end clipping, MSC-mid speech clipping, NDS-noise detectet as speech) dobivani su tako da su prvo granice govora označene ručno, a zatim uspoređene s onima koji su davali algoritmi. Rezultati su slijedeći. Algoritam implementiran u ovom radu (G.729b) je najgori u pogledu totalne pogrešne detekcije (suma svih pogrešnih detekcija FEC, MSC,NDC) dok je najbolji detektor govora baziran na Fuzzy logici. Po pitanju dva parametra FEC+MSC najlošijim se pokazao AMR detektor.



6. Zaključak:

U radu je opisano nekoliko algoritama za detekciju govora, te je jedan od njih odabran i implementiran u matlab programskom paketu te potom u C programskom jeziku te pokrenut na pločici ADSP-BF537 EZ-KIT Lite tvrtke Analog Devices.

Detekcija govora kao jedna od problematika usko vezanih uz digitalnu obradu i prijenos govora, kao što možemo vidjeti iz ovog rada ima još mjesta za poboljšanje i daljnje razvijanje. Neki od algoritama daju iznimno dobre rezultate, kao što su algoritami bazirani na pretraživanju oblika formanata i mjerama periodičnosti, no srednje su ili visoke složenosti te zahtjevaju velike računalne resurse. Drugi pak algoritmi postižu lošije rezultate kao što su detaljno opisani algoritam korišten G.729 dodatak b koderu govora, no izvršavaju se relativno brzo i troše puno manje resursa. S toga ovisno o našim potrebama možemo koristiti ili jednu ili drugu skupinu algoritama, brze ili točne, dok bi idealan slučaj bio naći sredinu između te dvije krajnosti. To bi mogli na dva načina ili poboljšavanjem platforme na kojoj se sam algoritam vrti, primjerice povećavanjem instrukcijskog paralelizma i smanjivanja vremena pristupa memorij ili povećavanjem efikasniosti složenih algoritama smanjenjem broja petlji ili uvođenjem potpuno novih koncepata izvedbe istih algortama. To smo mogli vidjeti kod algoritma baziranog na raspoznavanju uzoraka.

Rad na Blackfin razvojnoj pločici, iako pojednostavljen najviše moguće zbog svojih „plug and play“ mogućnsti, definitivno nije najbolje rješenje za brzi razvoj algoritama.Naime kako je jezik C sam po sebi „low level“ pa je s toga potrebno paziti na rukovanje memorijom, prenošenje parametara funkcije putem pokazivača i na još brojne slične stvari što u biti usporava razvijanje algoritama. Jedna stvar koja bi mogla eventualno ubrzati razvoj je Matlabov target support. Radi se o integriranom kompajleru koji prilagođava maltab enc kod za ciljanu platformu, u nekoliko jednostavnih koraka. Međutim pokazalo se da u slučaju kompleksnijih algoritama cijela stvar zakazuje te se neki dijelovi uopće ne prevode. Sve u svemu jako dobar koncept ali treba još puno dorade.

Na kraju samo valja reći da sam unatoč nedostatku predznanja u obradi govora radeći ovaj rad naučio jako puno bilo da se to odnosi na razvijanje aplikacija za Blackfin platformu, rad sa wav datotekama ili na obradu govora.

7. Literatura

1. L. R. Rabiner M. R. Sainbur, Voiced-Unvoiced-Silence Detection Using the Itakura LPC Distance Measure, Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '77, Svibanj 1977. 323-326

2. J.D. Hoyt, H. Wechsler, Detection of human speech in structured noise, Neural Networks, 1994. IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on, 4493

3. L. R. Rabiner, M.R. Sambur, Algorithm for determining endpoints of isolated utterances J. Acoust. Soc. Am. Volume 56, Issue S1, pp. S31-S31 1974

4. J.A. Haigh, J.S. Mason, Robutst voice activity detection using cepstral features, TENCON '93. Proceedings. Computer, Communication, Control and Power Engineering.1993 IEEE Region 10 Conference on, 321-324, 1993.

5. N.B. Yoma, F. McIannes, M. Jack, Robust speech pulse detection using adaptive noise modelling, Electronics Letters, Volume: 32 Issue: 15 1350 – 1352 , 1996

6. R Tucker, Voice activity detection using a periodicity measure, Communications, Speech and Vision, IEE Proceedings. 39 Issue: 4 377 – 380 1992.

7. F. Beritelli, S. Casale, A. Cavallaro, A Robust Voice Activity Detector for Wireless Communications Using Soft Computing, IEEE Journal on Selected Areas in Communications

8. Ramirez, J.; Segura, J.C.; Benitez, C.; Garcia, L.; Rubio, A. Statistical Voice Activity Detection Using a Multiple Observation Likelihood Ratio Test, Signal Processing Letters, IEEE, 689-692, Oct. 2005.

9. Saeed V. Advanced Digital Signal Processing and Noise Reduction, Second Edition, John Wiley & Sons Ltd, 2000.

10. Abut H., Hansen J.H.L. , Takeda K. DSP for in-vehicle and mobile systems, Springer 2005.

11. Annex B to ITU-T Recommendation G.729, was prepared by ITU-T Study Group 15 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 8th of November 1996

12. Grimm M. , Kroschel K. Robust speech recognition and understanding I-Tech ducation and publishing. 2007.

13. F. Beritelli, S. Casale, G. Ruggeri, S. Serrano Performance Evaluation and Comparison of G.729/AMR/Fuzzy Voice Activity Detectors. Signal Processing Letters, IEEE, 88-85, 2002

14. Chang J.H, Kim N.S, Mitra S.K, Voice Activity Detection Based on Multiple Statistical Models Signal Processing, IEEE Transactions on Communications, 1965 – 1976, 2006.

15. Junqua J., Mak B., Reaves B., A Robust Algorithm for Word Boundary Detection in the Presence of Noise Speech and Audio Processing, IEEE Transactions on Communications 406-412 1994.

16. Visual DSP++ manual http://www.analog.com/static/imported-files/software_manuals/50_linker_man.rev3.2.pdf

17. Blackfin ADSP- 537 hardware reference http://www.analog.com/static/imported-files/processor_manuals/bf537_hwr_Rev3.2.pdf

8. Naslov, sažetak i ključne riječi

Naslov

Izvedba sustava za detekciju govora

Ključne riječi

Detekcija govora, obrada govora, G.729b blackfin BF537, matlab, visualDSP++ .

Sažetak

Cilj rada je implementirati sustav za detekciju govora na blackfin porodici procesora točnije na razvojnoj pločici ADSP-BF537 EZ-KIT Lite i opisati postojeće algoritme za detekciju govora. U prvom dijelu rada opisano je 8 algoritama detekcije govora i jedan detaljno koji je u drugom dijelu modeliran u matlabu i u programskom jeziku ANSI C i prilagođen izvođenju na DSP procesoru.

Title, summary, keywords

Title

Implementation of a voice detection system

Keywords

Speech detection, speech processing, G.729b blackfin BF537, matlab, visualDSP++ .

Summary

Main goal of this work is to implement a voice detection system on blackfin processor family, to be exact ADSP-BF537 EZ-KIT Lite board, and also to describe existing voice detection algorithms. In the first part of the work 8 algorithms were described and one more in detail which is afterwards modeled in matlab and in ANSI C and specially adapted to run on a DSP processor.