SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Tihomir Tadić

KVANTIZACIJA FREKVENCIJA SPEKTRALNIH LINIJA U ADAPTIVNOM KODERU GOVORNOG SIGNALA S VIŠE BRZINA PRIJENOSA PRIMJENOM MODELA S GAUSSOVIM MJEŠAVINAMA

MAGISTARSKI RAD

Zagreb, 2010.

Magistarski rad je izrađen na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Mentor: *prof.dr.sc. Davor Petrinović*

Magistarski rad ima 91 stranicu i 9 stranica priloga.

Povjerenstvo za ocjenu magistarskog rada:

1. Prof.dr.sc. Mladen Vučić – predsjednik, Fakultet elektrotehnike i računarstva
2. Prof.dr.sc. Davor Petrinović – mentor, Fakultet elektrotehnike i računarstva
3. Doc.dr.sc. Ivica Kopriva – Institut „Ruđer Bošković“, Zagreb

Povjerenstvo za obranu magistarskog rada:

1. Prof.dr.sc. Mladen Vučić – predsjednik, Fakultet elektrotehnike i računarstva
2. Prof.dr.sc. Davor Petrinović – mentor, Fakultet elektrotehnike i računarstva
3. Doc.dr.sc. Ivica Kopriva – Institut „Ruđer Bošković“, Zagreb

Datum obrane magistarskog rada: 23. rujna 2010.

Sadržaj

Popis slika

Popis tablica

1. Uvod

Učinkovito kodiranje spektralne ovojnice kratkotrajnog govornog signala, predstavljene parametrima linearne predikcije (engl. *Linear Predictive Coding*, LPC), već je desetljećima značajna tema istraživanja u području kodiranja govora niskim brzinama prijenosa. LPC parametri općenito se kvantiziraju u formi frekvencija spektralnih linija (engl. *Line Spectral Frequencies*, LSFs), upotrebom vektorskog kvantizatora (engl. *Vector Quantizer*, VQ) [24]. Za zadanu prijenosnu brzinu, vektorski kvantizatori s potpunom pretragom tablice reprezentanata (engl. *full-search VQ*) općenito ostvaruju najmanje izobličenje, ali zahtijevaju opsežno pretraživanje i pokazuju velike memorijske zahtjeve pri velikim brzinama prijenosa. Nametanjem određenih strukturnih ograničenja moguće je, na račun učinka (engl. *performance*), reducirati računsku složenost ili memorijsku zahtjevnost (ponekad oboje) takvih vektorskih kvantizatora.

Adaptivni koder govornog signala s više brzina prijenosa (engl. *Adaptive Multi-Rate speech codec*, AMR codec) temelji se na linearnom prediktoru pobuđenom algebarskim kodovima (engl. *Algebraic Code Excited Linear Prediction*, ACELP) [5]. Koder podržava osam različitih môdova rada s brzinama prijenosa u rasponu od 4.75 kbit/s do 12.2 kbit/s, s mogućnošću promjene brzine prijenosa svakih 20 ms tj. iz jednog govornog okvira u drugi. Spektralnu ovojnicu kvantizira koristeći LSF reprezentaciju parametara LPC modela, te se ovisno o brzini prijenosa, estimacija i kvantizacija provodi jednom ili dva puta u jednom okviru od 160 uzoraka, što daje učestalost LPC analize od 50 ili 100 puta u sekundi. Postupak kvantizacije spektralne ovojnice, koji se koristi kod standardnog AMR kodera, relativno je jednostavan i, za môd rada s najvećom brzinom prijenosa, temelji se na razlomljenoj matričnoj kvantizaciji (engl. *Split Matrix Quantization*, SMQ) nad pogreškom MA (engl. *Moving Average*) vektorske predikcije. Ostali modovi rada umjesto SMQ koriste razlomljenu vektorsku kvantizaciju (engl. *Split Vector Quantization*, SVQ). Primjenom SMQ, dva susjedna vektora pogreške predikcije kodiraju se kao matrica 10×2, ali u grupama od samo 2 retka, što je uzrok nepotpunom iskorištenju unutar-okvirnih korelacija LSF vektora. Slično je i sa SVQ, koja vektor pogreške MA vektorske predikcije razdjeljuje na tri pod-vektora dužine 3, 3 i 4, te svaki od njih kvantizira i kodira zasebno. Obje metode pripadaju gore spomenutoj skupini vektorskih kvantizatora sa strukturnim ograničenjima, dobro su poznate u literaturi, relativno jednostavne, ali nažalost ne previsokog stupnja sažimanja.

U novije vrijeme predložen je novi postupak kvantizacije, koji se temelji na modelu s Gaussovim mješavinama (engl. *Gaussian Mixture Model*, GMM) [15], [27], [28]. Funkcija gustoće vjerojatnosti (engl. *Probability Density Function*, *pdf*) ulaznog vektorskog procesa modelira se kao linearna kombinacija (mješavina) *pdf* funkcijâ više Gaussovih razdioba. Primjenom Karhunen-Loève transformacije (KLT), parametri

aproksimacije koriste se za projektiranje transformacijskih kodera, optimiziranih za svaku pojedinu komponentu mješavine. Nakon transformacije, komponente vektora kvantiziraju se skalarnim kvantizatorima, koristeći se optimiziranim shemama alokacije bitova za svaku pojedinu komponentu mješavine. Dakle, riječ je o kombinaciji adaptivne dekorelacije (transformacije) vektorskog procesa i obične skalarne kvantizacije dekoreliranih komponenti. Ulazni LSF vektor kvantizira se svim transformacijskim koderima, te se evaluacijom spektralne distorzije (engl. *Spectral Distortion*, SD) odabire transformacijski koder, odnosno komponenta mješavine, koja najbolje kvantizira ulazni vektor u smislu najmanje distorzije.

Primjena GMM-a u vektorskoj kvantizaciji diskutirana je u nekoliko objavljenih radova. Tako se u [23], spajanjem dva ili više LSF vektora iz susjednih okvira, opisanim postupkom iskorištavaju i među-okvirne korelacije. U [34] se razmatra mogućnost smanjenja računske kompleksnosti na račun transparentnosti kvantizacijskog postupka, smanjivanjem broja komponenti mješavine čije transformacijske kodere uspoređujemo. Nadalje, u [36] je predložen praktičan postupak za projektiranje GMM-temeljenog vektorskog kvantizatora varijabilne brzine prijenosa, koji kombinira skalarnu kvantizaciju s mrežastom strukturom (engl. *lattice quantization*) i entropijsko kodiranje aritmetičkim koderom.

Predloženim postupcima ostvaruje se relativno visok stupanj sažimanja uz istovremeno malu složenost postupka kvantizacije. Tema ovog magistarskog rada odnosi se na istraživanje mogućnosti primjene i doprinosa takvog postupka kvantizacije u AMR koderu, kao jednom od standardiziranih kodera govornog signala, isključivo uz izmjenu dijela algoritma koji se odnosi na kvantizaciju LSF vektora. U tu svrhu koristi se, adaptira i kombinira nekoliko gore spomenutih pristupa. Funkcija gustoće vjerojatnosti pogreške predikcije LSF vektora aproksimira se modelom s Gaussovim mješavinama. Nadalje, adaptivna dekorelacija vektorskog procesa, temeljena na KLT-u, kombinira se s entropijski ograničenom skalarnom kvantizacijom (engl. *Entropy Constrained Scalar Quantization*, ECSQ) u tzv. shemu mekanog odabira (engl. *soft decision*) najbolje komponente mješavine. Entropijskim kodiranjem izlaznih indeksa skalarnih kvantizatora ostvaruje se ušteda u brzini prijenosa, te bolja prilagodba lokalnoj statistici izvora u odnosu na kodiranje s ograničenom rezolucijom (engl. *constrained resolution*). Takva struktura zapravo odgovara adaptivnom transformacijskom kodiranju, uz adaptaciju za svaki ulazni vektor. Krajnji cilj predstavlja projektiranje adaptivnog transformacijskog kodera, koji će iskoristiti unutar-okvirne korelacije diferencijalno kodiranih LSF parametara. Općenito je poznato da entropijski koderi produciraju izlazne tokove varijabilne brzine, te su u svrhu prilagodbe na AMR môdove fiksne prijenosne brzine, istražene tehnike varijabilnog kvantizacijskog koraka i odbacivanja vektorskih komponenti. Takva prilagodba ujedno predstavlja i glavnu novinu ovog rada, u usporedbi s dosadašnjim radom u ovom području. Poznato je, međutim, da su entropijski koderi znatno osjetljiviji na propagaciju pogreške, što aplikaciju predloženog algoritma ograničava na sustave koji koriste neki od mehanizama za zaštitu od pogreške u prijenosu ili pohrani podataka.

U sklopu istraživanja provedena je:

- analiza učinkovitosti sažimanja (odnos distorzije i brzine prijenosa),
- analiza i usporedba složenosti,
- objektivna i subjektivna usporedba polaznog i modificiranog sustava.

Na osnovu provedene analize predložena je struktura GMM-temeljenog kvantizatora spektralne ovojnice, čiji su parametri prilagođeni ostvarenju ili najvećeg mogućeg stupnja sažimanja, ili smanjenju broja matematičkih operacija potrebnih za provođenje postupka kvantizacije. Dobiveni rezultati i zaključci opisani su u radovima [30] i [31].

1.1. Organizacija magistarskog rada

U drugom poglavlju ovog rada ukratko su objašnjeni pojmovi i koncepti vezani uz linearnu predikciju i problematiku kvantizacije spektralne ovojnice. Uvedena je i objašnjena svrha alternativnih reprezentacija koeficijenata LPC filtra, s posebnim osvrtom na LSF parametrizaciju i njene prednosti u kvantizaciji spektralne ovojnice. Nadalje, dan je osvrt na osnovne razlike skalarne i vektorske kvantizacije, te uveden koncept transformacijskog kodiranja, koji obuhvaća i kvantizatore temeljene na modelu Gaussovih mješavina.

Princip kodiranja govornog signala referentnim AMR koderom opisan je u trećem poglavlju. Struktura AMR kodera i dekodera, temeljena na ACELP algoritmu, prikazana je pojednostavljenim blok dijagramima, koji jasno opisuju postupke analize i sinteze govornog signala. S obzirom da se tema rada odnosi na mogućnost korištenja alternativnog postupka kvantizacije LSF vektora, detaljno su opisane kvantizacijske metode koje se koriste u referentnom AMR koderu.

U četvrtom poglavlju opisan je model s Gaussovim mješavinama, te je diskutirana mogućnost njegove primjene u vektorskoj kvantizaciji. Objašnjen je postupak estimacije parametara GMM modela nepoznate *pdf* funkcije primjenom iterativnog EM algoritma, koji je temeljen na kriteriju maksimalne vjerodostojnosti (engl. *maximum likelihood*, ML) promatrane realizacije procesa. U nastavku je opisana dekorelacijska uloga KLT transformacije, koja uz pretpostavku Gaussove razdiobe ulaznog procesa omogućava optimalnu skalarnu kvantizaciju njegovih transformiranih komponenti.

S obzirom na strukturu AMR kodera i opisani koncept vektorske kvantizacije primjenom transformacijskog kodiranja, u petom poglavlju je za kvantizaciju spektralne ovojnice u AMR koderu predložen model kvantizatora temeljen na primjeni GMM. Uz blok dijagram, opisan je postupak projektiranja predloženog kvantizatora, te alternativni kriteriji inicijalnog pridruživanja (asocijacije) najbolje komponente mješavine u procesu kodiranja. Također, napravljen je detaljan osvrt na korištene tehnike prilagodbe varijabilne izlazne brzine entropijski ograničenih skalarnih kvantizatora na fiksnu prijenosnu brzinu AMR kodera. U nastavku su opisane metode koje su korištene za objektivno vrednovanje i usporedbu svojstava polaznog i modificiranog sustava. Na kraju, predložena je jednostavnija varijanta GMM-temeljenog kodera, koja uz nešto

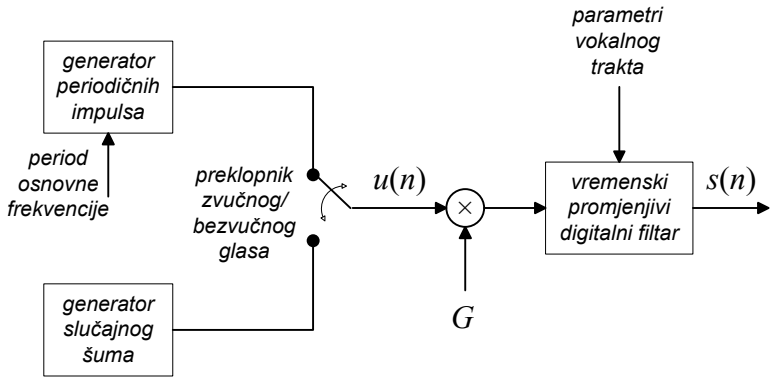
lošiju kvantizaciju spektralne ovojnice smanjuje računsku složenost predloženog algoritma.

Svi eksperimentalni postupci vezani uz ovaj rad provedeni su simulacijama u programskom paketu Matlab. Rezultati simulacija prikazani su, objašnjeni i komentirani u šestom poglavlju. Uz opis izvora podataka (govornih baza) i simulacijskog modela, prezentirani su i diskutirani rezultati varijacije pojedinih parametara kvantizatora i primjene alternativnih procedura u postupku njegova projektiranja. Napravljena je usporedba svojstava polaznog i modificiranog sustava iste prijenosne brzine, te istražena mogućnost njene redukcije uz zadržavanje učinaka referentnog sustava. Na kraju, analizirana je računska složenost i memorijski zahtjevi polaznog i obje varijante modificiranog sustava, te su dobiveni rezultati uspoređeni po odgovarajućim modovima rada AMR kodera.

Sedmo poglavlje donosi kratak pregled zaključaka provedenog istraživanja.

2. Kvantizacija spektralne ovojnice

Kodiranje govornog signala provodi se s osnovnom namjerom ostvarenja najbolje moguće kvalitete rekonstruiranog govora za zadanu prijenosnu brzinu kodiranih parametara, odnosno minimizacije potrebne prijenosne brzine za zadanu razinu distorzije. Pri tome je, pogotovu za aplikacije koje rade u stvarnom vremenu, kodiranje govora i dekodiranje njegovih parametara poželjno provesti uz što manju računsku složenost i memorijske zahtjeve. Ovi ciljevi su najčešće u međusobnoj suprotnosti, jer algoritmi za kodiranje govora visoke kvalitete i male prosječne brzine prijenosa zahtijevaju složene postupke njegove analize i sinteze. U tom je smislu od velikog značaja istraživanje i pronalaženje učinkovitijih algoritama male složenosti, koji postižu dobru kvalitetu pri niskim brzinama prijenosa.



Slika 2.1 *Izvor/filtar model formiranja govornog signala.*

Vrlo velik broj algoritama za kodiranje govora temelji se na modelu izvor/filtar (engl. *source/filter model*), prikazan na slici 2.1. Spomenutim modelom, akustička prijenosna funkcija vokalnog trakta modelira se vremenski promjenjivim digitalnom filtrom, koji pobuđen signalom izvora producira govor. Postupak kodiranja, temeljen na takvom modelu, prvenstveno podrazumijeva određivanje parametara izvora i filtra, te njihovu kvantizaciju. Sintezom se na prijemnoj strani iz kvantiziranih parametara rekonstruira polazni govorni signal. Obzirom da govorni signal nije stacionaran, parametri spomenutog modela određuju se na pretpostavci kvazi-stacionarnosti, odnosno na dovoljno kratkim odsječcima signala (okvirima), na kojima se pretpostavljaju stalna statistička i spektralna svojstva.

Kvaliteta sintetiziranog govora, osim o optimalnosti parametarskog modela, ovisi i o kvantizacijskoj metodi izračunatih parametara modela. Kvantizacija parametara modela omogućava njihov zapis konačne duljine, što smanjuje brzinu potrebnu za njihov prijenos, odnosno veličinu memorije potrebnu za njihovu pohranu. Ova prednost plaćena

je neizbježnom degradacijom (izobličenjem) rekonstruiranog signala u odnosu na polazni govorni signal.

2.1. Linearna predikcija

Rezultati mnogih istraživanja pokazali su da je prijenosnu funkciju filtra u modelu izvor/filtar moguće vrlo dobro aproksimirati prijenosnom funkcijom rekurzivnog digitalnog filtra (IIR) bez nula (engl. *all-pole filter*). U tom slučaju, prijenosna funkcija pojednostavljenog modela formiranja govornog signala (slika 2.1) dana je izrazom:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 + \sum_{k=1}^{p} a_k z^{-k}}, \qquad (2.1)$$

gdje su $U(z)$ i $S(z)$ z-transformacije pobude, odnosno odziva sustava, dok su a_1 do a_p koeficijenti prijenosne funkcije IIR filtra. Konstanta G predstavlja faktor pojačanja. Izražavanjem izlaznog signala u vremenskoj domeni, iz pripadne jednadžbe diferencija dobiva se:

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + Gu(n), \qquad (2.2)$$

gdje $s(n)$ i $u(n)$ predstavljaju odziv, odnosno pobudni signal sustava u vremenskoj domeni. Koeficijenti filtra izračunavaju se LPC metodom, koja je jedan od najčešće korištenih postupaka za analizu i sintezu govornog signala u koderima s niskim brzinama prijenosa. Postupak se temelji na činjenici da je, kod signala koji nemaju ravan (bijeli) spektar, trenutni uzorak moguće aproksimirati linearnom kombinacijom p prethodnih uzoraka, odnosno linearnim prediktorom reda p:

$$\tilde{s}(n) = \sum_{k=1}^{p} \alpha_k s(n-k), \qquad (2.3)$$

gdje su α_1 do α_p koeficijenti linearnog prediktora, a $\tilde{s}(n)$ predikcija izlaznog signala u koraku n. Izraz 2.3 ukazuje da se linearni prediktor može promatrati kao filtar s konačnim impulsnim odzivom (FIR).

Iz sličnosti izraza 2.2 i 2.3 vidimo da bi, oduzimanjem predikcije izlaznog signala od njegove stvarne vrijednosti, pod uvjetom da je

$$a_i = -\alpha_i, \quad \text{za} \quad i = 1 \ \text{ do } \ p, \qquad (2.4)$$

dobili pogrešku predikcije:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k) = Gu(n). \qquad (2.5)$$

Gornji izraz omogućava razlaganje govornog signala na koeficijente linearnog sustava i pobudni signal $Gu(n)$, pod uvjetom da je koeficijente linearnog prediktora α_1 do α_p

moguće direktno odrediti iz govornog signala $s(n)$, tako da oni predstavljaju dobru aproksimaciju stvarnih koeficijenata filtra a_1 do a_p. Ovakvim razlaganjem se govorni signal $s(n)$ transformira u rezidualni signal $e(n)$ značajno manje energije, što olakšava njegovu digitalnu reprezentaciju. Prijenosna funkcija, koja odgovara jednadžbi diferencija u izrazu 2.5, predstavlja FIR filtar reda p, sa govornim signalom $s(n)$ na ulazu i pogreškom predikcije $e(n)$ na izlazu:

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^{p} \alpha_k z^{-k} \,. \qquad (2.6)$$

Filtar opisan gornjom prijenosnom funkcijom se u literaturi često naziva inverznim filtrom, čija se prijenosna funkcija, pod uvjetom 2.4, može pisati u obliku:

$$H(z) = \frac{G}{A(z)} \,. \qquad (2.7)$$

Postupak linearne predikcije provodi se za sve okvire govornog signala i za cilj ima odabir koeficijenata prediktora koji minimiziraju srednju kvadratnu pogrešku predikcije $e(n)$ za sve uzorke analiziranog okvira. Među popularne i učestalo korištene postupke za izračun koeficijenata spadaju postupci autokorelacije i kovarijance, koji daju dobru aproksimaciju stvarnih koeficijenata linearnog sustava. Primjenom autokorelacijske metode dobiva se autokorelacijska matrica sa strukturom Toeplitzove matrice, što omogućava izračun koeficijenata LPC filtra korištenjem računski brzih algoritama poput Levinson-Durbinovog algoritma. Izračunati koeficijenti se onda primjenjuju za određivanje signala pobude, odnosno rezidualnog signala predikcije, filtracijom polaznog govornog signala $s(n)$ inverznim filtrom $A(z)$.

Za uspješnu rekonstrukciju (sintezu) kodiranog govora, potrebno je za svaki okvir na prijemnu stranu prenijeti i estimirane koeficijente linearnog sustava i njegov pobudni signal. Potrebna brzina prijenosa kodiranih parametara ovisit će o učestalosti LPC analize, odnosno duljini vremenskog okvira, redu LPC filtra, te o načinu kodiranja estimiranih LPC parametara i odgovarajuće pobude. S obzirom na kvazi-stacionarnost govornog signala, analiza se tipično provodi na okvirima duljine 20 ms, s učestalošću od 50 Hz. Kako transformacija koeficijenata LPC filtra u frekvencijsku domenu daje ovojnicu spektra analiziranog (kratkotrajnog) okvira govornog signala, red filtra p direktno utječe na preciznost aproksimacije ovojnice. Općenito će veći red predikcijskog filtra rezultirati preciznijom aproksimacijom spektralne ovojnice, ali i potrebom za većom brzinom prijenosa kodiranih koeficijenata. Iz tog se razloga za govorne signale uzorkovane frekvencijom od 8 kHz, tipično primjenjuje prediktor 10-og reda, računajući po dva pola prijenosne funkcije za svaki formant spektralne ovojnice. Na kraju, učinkovitim kodiranjem spektralne ovojnice moguće je značajno reducirati potrebnu brzinu prijenosa, jer kod kodera koji rade u području srednjih brzina prijenosa (2.4 do 8 kbit/s), kodiranje ovojnice troši približno jednu četvrtinu do jedne polovine ukupne brzine prijenosa. Općenito je svim koderima, temeljenim na postupku linearne predikcije, zajedničko kodiranje spektralne ovojnice govornog signala, dok se međusobno prvenstveno razlikuju po načinu kodiranja rezidualnog signala.

2.2. Kvantizacija spektralne ovojnice

Kvantizacijskim se algoritmima parametarski skup koji definira koeficijente LPC filtra nastoji reprezentirati što manjim brojem bitova, što bi omogućilo redukciju brzine potrebne za njihov prijenos na prijemnu stranu, odnosno memorijski prostor potreban za njihovu pohranu. Posljedica kvantizacije koeficijenata LPC filtra je neminovna promjena spektralne ovojnice, koja se očituje kao spektralno izobličenje govornog signala.

Koeficijenti direktne forme LPC filtra, tzv. *a*-koeficijenti, nisu pogodni za kvantizaciju, prvenstveno zbog svoje visoke spektralne osjetljivosti, te širokog raspona vrijednosti koje poprimaju. Naime, male pogreške zaokruživanja prilikom njihove kvantizacije mogu uzrokovati velika izobličenja spektralne ovojnice, te u pitanje dovode stabilnost LPC filtra. U svrhu povećanja robusnosti na kvantizacijski šum razvijene su različite alternativne reprezentacije, koje posjeduju jednoznačnu reverzibilnu transformaciju u *a*-koeficijente. Neke od njih su koeficijenti refleksije (PARCOR) [11], logaritam omjera poprečnih presjeka (engl. *Log Area Ratio*, LAR) [29], te arcus-sinus parametri (ARSIN) [10], koji su redom temeljeni na skalarnoj kvantizaciji individualnih parametara odgovarajuće reprezentacije.

2.2.1. LSF reprezentacija LPC parametara

Jednu od alternativnih reprezentacija koeficijenata direktne forme LPC filtra predstavljaju i frekvencije spektralnih linija [18], također poznate kao parovi spektralnih linija (engl. *Line Spectrum Pairs*, LSP). Od 1980-tih LSF reprezentacija postaje dominantnom parametrizacijom, pokazujući brojna svojstva koja ih čine vrlo pogodnim za kvantizaciju i interpolaciju LPC parametara [29]. Među njima su područje vrijednosti ograničeno na interval $[0,\pi]$ (što olakšava projektiranje kvantizatora), rastući redoslijed parametara, te jednostavna provjera stabilnosti LPC filtra (očuvanjem svojstva alterniranja korijena nakon kvantizacije). Nadalje, kako se radi o reprezentaciji u frekvencijskoj domeni, spektralno izobličenje uzrokovano kvantizacijom LSF parametara može se vrlo dobro aproksimirati težinskom euklidskom udaljenošću kvantiziranog i nekvantiziranog LSF vektora. Možda i najvažniji razlog njihove intenzivne primjene su njihova interpolacijska svojstva, koja omogućavaju prilično točnu predikciju njihovih vrijednosti između dva trenutka s poznatim vrijednostima LSF parametara. Saznanja o vektorskoj kvantizaciji početkom 80-tih godina dovela su do brojnih istraživanja učinkovitijeg kodiranja spektralne ovojnice primjenom LSF reprezentacije.

Uz inverzni LPC filtar reda p opisan polinomom:

$$A(z) = 1 + \sum_{k=1}^{p} a_k z^{-k} , \qquad (2.8)$$

LSF parametri definirani su preko korijena pomoćnih polinoma $P(z)$ i $Q(z)$, reda $p+1$, konstruiranih upotrebom polinoma $A(z)$:

$$P(z) = A(z) + z^{-(p+1)} A(z^{-1})$$
$$= 1 + (a_1 + a_p) z^{-1} + (a_2 + a_{p-1}) z^{-2} + ... + (a_p + a_1) z^{-p} + z^{-(p+1)}, \quad (2.9)$$

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1})$$
$$= 1 + (a_1 - a_p) z^{-1} + (a_2 - a_{p-1}) z^{-2} + ... + (a_p - a_1) z^{-p} - z^{-(p+1)}, \quad (2.10)$$

gdje p predstavlja red linearnog prediktora, a a_i njegov i-ti koeficijent u direktnoj formi. Pri tome je $P(z)$ polinom sa simetričnim koeficijentima, dok su koeficijenti polinoma $Q(z)$ asimetrični. Kako je već spomenuto, za govorne signale uzorkovane frekvencijom od 8 kHz, tipično se primjenjuje prediktor reda p = 10. Za parni je red prediktora po jedan od korijena takvih polinoma unaprijed poznat, pa se njihovim izdvajanjem:

$$P(z) = (1 + z^{-1}) P'(z), \quad (2.11)$$

$$Q(z) = (1 - z^{-1}) Q'(z), \quad (2.12)$$

dobivaju simetrični polinomi $P'(z)$ i $Q'(z)$ reda p, čiji su korijeni locirani na jediničnoj kružnici:

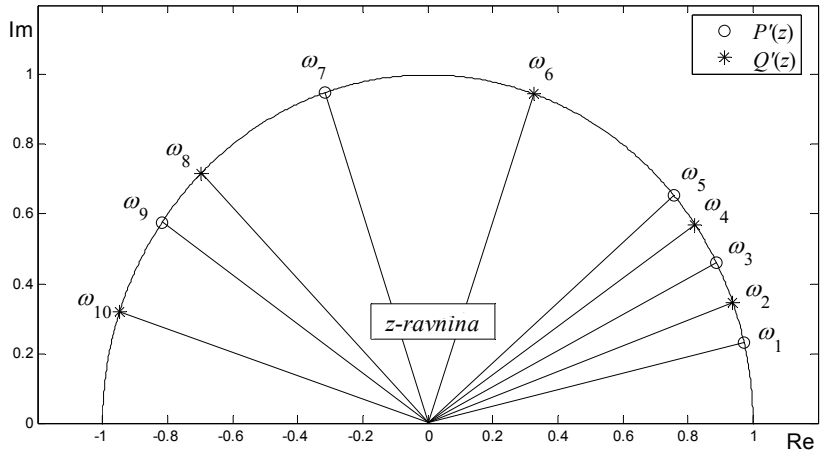
$$P'(z) = \prod_{i=2,4,...,p} \left(1 - e^{j\omega_i} z^{-1}\right)\left(1 - e^{-j\omega_i} z^{-1}\right), \quad (2.13)$$

$$Q'(z) = \prod_{i=1,3,...,p-1} \left(1 - e^{j\omega_i} z^{-1}\right)\left(1 - e^{-j\omega_i} z^{-1}\right). \quad (2.14)$$

Linije spektralnih frekvencija od $A(z)$, odnosno LSF parametri, zapravo odgovaraju kutevima ω_1, ω_2, ... , ω_p odgovarajućih korijena polinoma $P'(z)$ i $Q'(z)$, koji kod stabilnih LPC filtara alterniraju rastućim redoslijedom [25], tj.:

$$0 < \omega_1 < \omega_2 < ... < \omega_p < \pi, \quad (2.15)$$

kako je i prikazano na slici 2.2.



Slika 2.2 *Ilustracija položaja korijena polinoma P'(z) i Q'(z) na jediničnoj kružnici u z-ravnini.*

Činjenica da je stabilnost LPC filtra moguće lako osigurati, na način da svi korijeni polinoma $P(z)$ i $Q(z)$ alterniraju rastućim redoslijedom na jediničnoj kružnici, LSF parametre čini posebno pogodnim za kvantizaciju LPC koeficijenata. Dakle, nakon kvantizacije, za stabilnost LPC filtra dovoljno je osigurati da kvantizirani parametri zadovoljavaju navedeni uvjet. Nadalje, u korist ove teze doprinosi i činjenica o lokaliziranosti spektralne osjetljivosti LSF parametara. To znači da se pogreška određenog LSF parametra reflektira na LPC spektar snage samo u njegovoj okolici, što nije slučaj kod ostalih alternativnih reprezentacija LPC koeficijenata. Skup od dva ili tri LSF parametra karakterizira određenu rezonantnu frekvenciju (formant), čija širina ovisi o njihovoj blizini.

Korijene pomoćnih polinoma, odnosno odgovarajuće LSF parametre, moguće je izračunati korištenjem nekoliko numeričkih metoda. Jedna od njih [25], nakon primjene diskretne kosinusne transformacije (DCT) na polinome $P'(z)$ i $Q'(z)$, svodi se na pronalaženje nula sume kosinusnih funkcija sa cjelobrojnim omjerima frekvencija na intervalu $[0,\pi]$, dok druga metoda koja koristi Chebyshevljeve polinome [20], problem reducira na traženje realnih korijena običnih polinoma na intervalu [-1,1].

Iz izraza 2.9 i 2.10 vidljivo je da je transformacija LPC koeficijenata u LSF parametre reverzibilna, te da je a-koeficijente moguće izračunati preko pomoćnih polinoma $P(z)$ i $Q(z)$:

$$A(z) = \frac{1}{2}\left[P(z) + Q(z)\right]. \qquad (2.16)$$

2.2.2. Skalarna kvantizacija

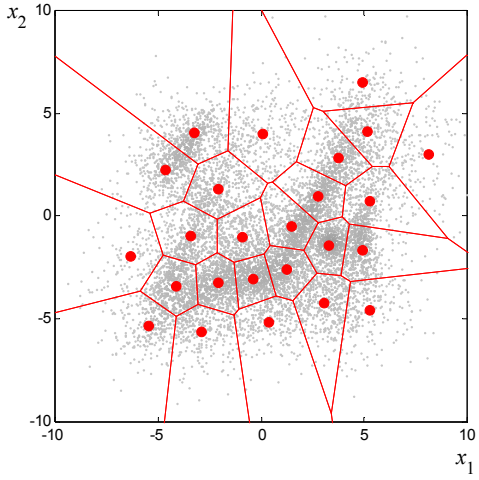
Skalarna kvantizacija je najjednostavniji način kvantizacije, kojim se uzorku koji se kvantizira jednostavno pridjeljuje najbliža razina kvantizatora. Na prijemnu stranu potrebno je prenijeti indeks pridjeljene razine, kojoj se onda dodjeljuje odgovarajuća vrijednost kvantiziranog uzorka. Kao osnovne tipove, razlikujemo uniformne i neuniformne skalarne kvantizatore.

Kvantizacijske razine uniformnih skalarnih kvantizatora jednako su udaljene jedna od druge, što postupak kvantizacije svodi na običnu operaciju zaokruživanja.

Kvantizacijske razine neuniformnih skalarnih kvantizatora nejednoliko su raspoređene u rasponu vrijednosti kvantizatora, tako da područja sa statistički češćim vrijednostima imaju veći broj i gustoću kvantizacijskih razina. Na taj se način omogućava finija kvantizacija područja vrijednosti od većeg interesa, što smanjuje prosječno izobličenje kvantiziranog signala. Međutim, takvo smanjenje izobličenja dobiveno je na račun povećane složenosti kvantizacijskog algoritma, zbog složenije pretrage nejednoliko raspoređenih kvantizacijskih razina. Problem računske složenosti moguće je reducirati primjenom kompandera (kompresor + ekspander), koji nejednoliku distribuciju ulaznog signala nelinearnom transformacijom prevodi u jednoliku (kompresija), što postupak kvantizacije ponovno svodi na funkciju zaokruživanja uniformnim kvantizatorom.

2.2.3. Vektorska kvantizacija

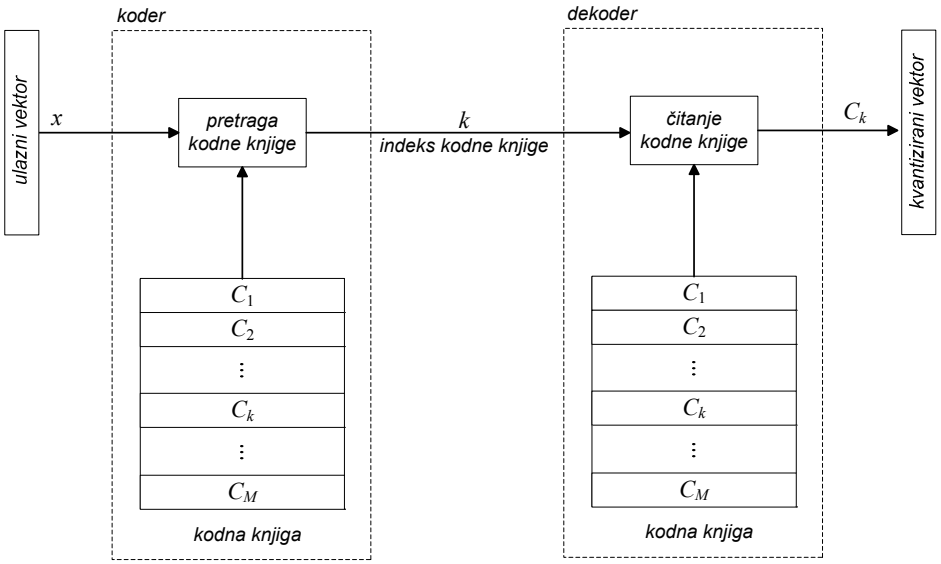
Tehnika vektorske kvantizacije pojavljuje se početkom osamdesetih godina kao vrlo učinkovita alternativa skalarnoj kvantizaciji. Za razliku od neovisnog kodiranja uzoraka skalarnom kvantizacijom, tehnika vektorske kvantizacije skupno kodira cjelokupni vektor koeficijenata LPC filtra, što omogućava maksimalno iskorištenje unutar-okvirnih statističkih zavisnosti među vektorskim komponentama, odnosno iskorištenje činjenice da npr. određena vrijednost prve komponente vektora za sobom povlači vrijednost druge komponente u određenom području. Statistička zavisnost uključuje linearnu statističku zavisnost (korelaciju), koju je moguće ukloniti (dekorelacija) primjenom neke od linearnih transformacija, ali nije ograničena na nju. Ona, naime, uključuje i nelinearnu statističku zavisnost za koju nije moguće pronaći linearnu transformaciju koja bi međusobno zavisne vektorske komponente prevela u statistički nezavisne slučajne varijable [8]. Poznato je da vektorski kvantizatori bez ograničenja i strukture (engl. *unconstrained vector quantizers*), u usporedbi sa svim ostalim kvantizacijskim metodama, ostvaruju najmanje izobličenje kvantiziranog signala za zadanu brzinu prijenosa i dimenziju vektora. Zbog slobode optimalnog razmještanja vektora kodne knjige u višedimenzionalnom prostoru (slika 2.3) na način da vektore kodne knjige najbolje prilagode gustoći vjerojatnosti vektorskog procesa, takvi vektorski kvantizatori u mogućnosti su učinkovito kvantizirati vektore koji, osim korelacije, imaju i nelinearne statističke zavisnosti među vektorskim komponentama. Za razliku od njih, skalarni kvantizatori nisu u mogućnosti ukloniti niti korelaciju među susjednim komponentama, jer svaku od njih kvantiziraju zasebno.



Slika 2.3 *Primjer Voronoi dijagrama vektorskog kvantizatora. Crvenom bojom prikazane su ćelije, te razmještaj reprezentanata kodne knjige. Vektorski kvantizator projektiran je na temelju realizacije 2D vektorskog procesa, prikazane sivom bojom.*

Primjenom nekog od kriterija pogreške, vektorskom kvantizacijom se ulaznom vektoru x pridjeljuje jedan od vektora (centroida) C_k iz tablice reprezentanata (kodne knjige). Tablicu reprezentanata moguće je projektirati na način da kvantizacija baze trening vektora daje najmanje izobličenje, što rezultira optimalnom razdiobom reprezentanata u

d-dimenzionalnom prostoru, u skladu s funkcijom gustoće razdiobe vektora u trening bazi. Na prijemnu stranu šalje se samo indeks *k* odabranog reprezentanta (slika 2.4), što rezultira velikim uštedama u prijenosu.



Slika 2.4 *Vektorska kvantizacija.*

Nadalje, prednost VQ nad SQ ogleda se i u obliku kvantizacijske ćelije. Naime, primjena SQ na *d*-dimenzija rezultira kvantizacijskim ćelijama u obliku hiper-kocki, dok kvantizacijske ćelije VQ poprimaju oblik *d*-dimenzionalnih heksagonalnih poligona, što reducira prosječnu pogrešku kvantizacije u korist VQ.

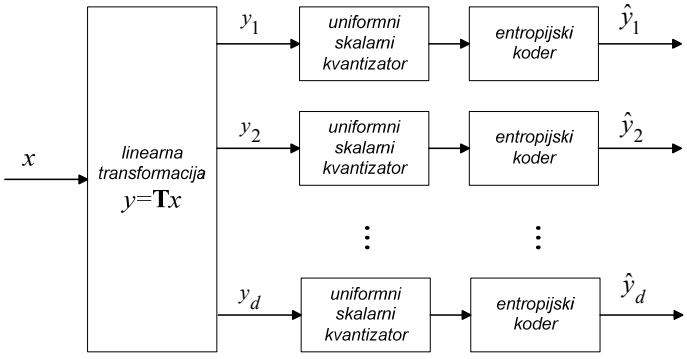
Uz navedene prednosti, vektorska kvantizacija za sobom povlači i određene nedostatke. To se prvenstveno odnosi na povećanu računsku složenost i memorijske zahtjeve vektorskih kvantizatora, koji su posljedica pretrage i pohrane tablice reprezentanata, čija veličina eksponencijalno raste s rezolucijom, odnosno prijenosnom brzinom kvantizatora. To je slučaj kod nestrukturiranog vektorskog kvantizatora, čiji su reprezentanti razmješteni bez reda i strukture. Takav kvantizator u postupku kvantizacije treba izračunati udaljenost ulaznog vektora do svakog reprezentanta, te odabrati vektor s najmanjom udaljenošću (distorzijom). Učinak kvantizacije u tom je smislu bolji što je dimenzija vektora koji se kvantizira veća [13]. Međutim, činjenica da im računska složenost i memorijski zahtjevi eksponencijalno rastu s brzinom prijenosa i dimenzijom vektora čini ih često nepraktičnim za aplikacije s višim brzinama prijenosa.

Računsku složenost i memorijske zahtjeve VQ moguće je reducirati nametanjem određene strukture u tablici reprezentanata, koja za kvantizaciju pojedinog vektora ne zahtijeva pretragu cjelokupne tablice. Takvi VQ spadaju u skupinu tzv. strukturiranih VQ, čiji je učinak, zbog nametnute strukture, ipak lošiji od optimalnog. Jedan od načina za strukturiranje tablice reprezentanata je dekompozicija tablice u Cartesijev produkt manjih tablica, koju koristi skupina kvantizacijskih algoritama s produktnim kodom (engl. *product code quantization algorithms*). Tipičan predstavnik ove skupine je SVQ

algoritam, kod kojeg se više pod-vektora ulaznog vektora kvantizira pomoću više vektorskih kvantizatora s manjim tablicama reprezentanata, gdje svaki od njih kvantizira jedan pod-vektor. Manji kvantizatori omogućavaju jednostavniju pretragu tablice reprezentanata, a često i pohranu u manjem memorijskom prostoru. Razlog nešto lošijim svojstvima ovakvog kvantizatora leži u sekvencijalnom i neovisnom načinu pretrage i projektiranja njegovih tablica, što onemogućava potpuno iskorištenje unutar-okvirnih statističkih zavisnosti u ovim postupcima.

2.2.4. Transformacijsko kodiranje

Općenito se skalarna kvantizacija smatra učinkovitijom, što su komponente vektora statistički nezavisnije [8]. Poznato je da, pod pretpostavkom velikih brzina prijenosa (engl. *high-rate*), skalarna kvantizacija statistički nezavisnih varijabli dostiže svojstva vektorskih kvantizatora, koji koriste kriterij srednje kvadratne pogreške (engl. *Mean Squared Error*, MSE). Pretpostavka velikih brzina prijenosa odgovara brzinama dovoljno velikim da povećanje rezolucije kvantizatora za 1 bit smanjuje prosječnu distorziju za 6 dB. Statistička nezavisnost vektorskih komponenti (koja implicira njihovu nekoreliranost) zapravo neutralizira prednost vektorskih kvantizatora u iskorištavanju unutar-okvirnih statističkih zavisnosti. U tom bi smislu linearna transformacija vektorskog procesa s ciljem dekorelacije njegovih komponenti omogućila primjenu i iskorištenje manje računske složenosti skalarnih kvantizatora. Na opisanom principu temelje se transformacijski koderi, koji predstavljaju jednostavniju alternativu vektorskoj kvantizaciji bez strukturnih ograničenja [2]. Kao poseban slučaj u skupini vektorskih kvantizatora produktnog koda, transformacijski koderi najbolji učinak pokazuju pri kvantizaciji vektorskih izvora s izrazitom korelacijom među svojim komponentama. Potrebno je napomenuti da, za razliku od vektorske kvantizacije bez ograničenja, transformacijsko kodiranje ne iskorištava nelinearne statističke zavisnosti među komponentama vektora. Takav postupak, prikazan blok shemom na slici 2.5, obuhvaća grupiranje n koreliranih uzoraka u vektor x, dekorelaciju vektorskih komponenti primjenom linearne transformacije, te neovisnu skalarnu kvantizaciju dekoreliranih vektorskih komponenti. Općenito se radi o uniformnim skalarnim kvantizatorima, čiji se izlazni indeksi kodiraju entropijskim koderima varijabilne brzine prijenosa.

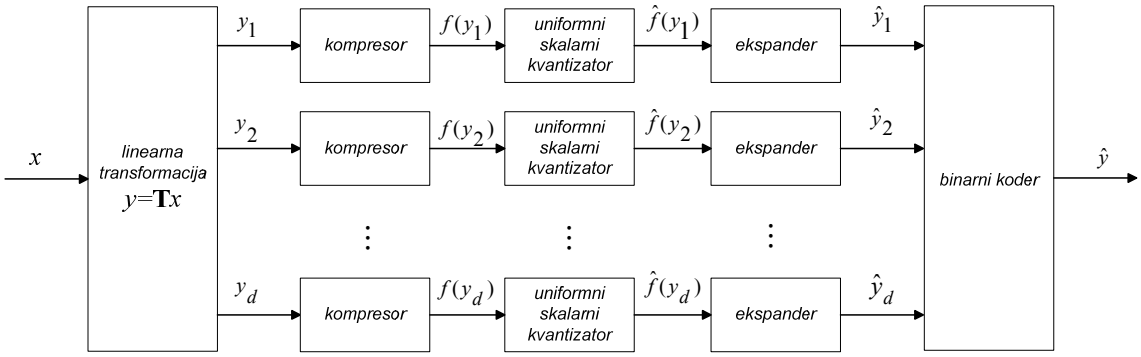


Slika 2.5 *Blok dijagram transformacijskog kodera.*

Pretpostavljajući uniformnu koreliranost vektorskih komponenti kroz cijeli vektorski prostor, njihovu dekorelaciju moguće je napraviti primjenom KLT transformacije, koja osim dekorelacijske uloge, također koncentrira energiju u nekoliko prvih vektorskih komponenti u transformiranoj domeni. Njena optimalnost ovisna je o sličnosti vektorske distribucije s Gaussovom, što često nije slučaj u distribucijama realnih vektorskih procesa. KLT transformacija spada u grupu podatkovno zavisnih transformacija, jer ovisi o matrici kovarijance vektorskog procesa. Njena primjena u praksi uvjetovana je izračunom transformacijskih matrica na velikom broju vektora (trening bazi) koji dobro reprezentiraju vektorski proces koji se kvantizira.

2.2.5. Blok kvantizacija

Posebnim slučajem transformacijskih kodera, smatraju se blok kvantizatori (engl. *block quantizers*), koji za kvantizaciju u transformiranoj domeni koriste neuniformne skalarne kvantizatore fiksne prijenosne brzine. Zbog svoje složenosti, neuniformni skalarni kvantizatori često se zamjenjuju kombinacijom kompandera i uniformnog skalarnog kvantizatora za svaku komponentu transformiranog vektorskog procesa (slika 2.6), na taj način izbjegavajući upotrebu klasične tablice reprezentanata.



Slika 2.6 *Blok dijagram blok kvantizatora.*

Kako svaka komponenta transformiranog vektorskog procesa ima drugačiju varijancu, što je prvenstveno posljedica KLT transformacije, obično se za svaku od njih projektira poseban SQ. Zbog fiksne brzine prijenosa, kod ovih kvantizatora javlja se problem optimalne raspodjele raspoloživih bitova, jer je distorzija svakog kvantizatora direktno određena njegovom rezolucijom. U tom smislu raspoložive bitove potrebno je raspodijeliti na način da se minimizira ukupna pogreška svih skalarnih kvantizatora. Jedan od popularnih algoritama za raspodjelu raspoloživih bitova je tzv. 'pohlepni' (engl. *greedy*) algoritam, koji raspoložive bitove dodjeljuje vektorskim komponentama s najvećom varijancom, jedan po jedan, redom do nestanka. Njegova raspodjela, zbog svoje cjelobrojne prirode, često je suboptimalno rješenje, jer obično neke od komponenti imaju nekoliko razina previše, dok ih druge imaju nekoliko premalo.

3. Struktura AMR kodera

AMR koder predstavlja algoritam optimiziran za kodiranje govornog signala. U listopadu 1998. prihvaćen je za standardni koder govornog signala od strane 3GPP (engl. *3rd Generation Partnership Project*), te je trenutno u širokoj upotrebi u GSM (engl. *Global System for Mobile Communications*) i 3G mobilnim mrežama. Za razliku od prethodnih GSM kodera fiksne prijenosne brzine, koji koriste fiksnu razinu zaštite od pogrešaka u prijenosu, AMR sustav prilagođava se uvjetima prijenosnog kanala. Pri lošijim kanalnim uvjetima koder smanjuje prijenosnu brzinu kodiranog govora, pri tome povećavajući udio kanalnog kodiranja (bitova za zaštitu od pogreške) u prijenosnoj brzini kanala. Na taj način smanjuje se vjerojatnost pojavljivanja pogrešaka u prijenosu na račun smanjene kvalitete kodiranog govora, što se pokazalo robusnijim rješenjem od kodiranja govora najvećom prijenosnom brzinom uz učestale pogreške u prijenosu. U tom smislu AMR koder podržava 8 modova rada, s različitim prijenosnim brzinama kodiranog govora u rasponu od 12.2 kbit/s do 4.75 kbit/s.

AMR koder temelji se na ACELP algoritmu, koji pripada općenitijoj klasi kodno pobuđenih linearnih prediktora (engl. *Code Excited Linear Prediction*, CELP). Način rada sličan je za sve modove, osim za môd s najvećom prijenosnom brzinom (12.2 kbit/s), koji je ekvivalentan GSM EFR (engl. *Enhanced-Full Rate*) koderu govornog signala. Ostali modovi razlikuju se po alokacijama bitova i kvantizacijskim razinama korištenim za kodiranje pojedinih parametara govornog signala.

Detaljan opis funkcionalnosti pojedinih dijelova AMR kodera nadilazi okvire ovog rada, pa je u nastavku dan pojednostavljeni prikaz strukture AMR kodera i dekodera, s nešto detaljnijim osvrtom na način kvantizacije spektralne ovojnice, koji je neposredno vezan uz temu ovog rada.

3.1. Princip kodiranja govornog signala AMR koderom

AMR koder spada u skupinu kodera valnog oblika, jer govorni signal kodira u vremenskoj domeni. Kodiranje govora provodi se nad okvirima duljine trajanja 20 ms, uz frekvenciju uzorkovanja 8 kHz. Posljedično, svaki kodirani okvir sadrži kodirane parametre ACELP modela, koji predstavljaju 160 uzoraka originalnog govora. U dekoderu se kodirani parametri dekodiraju i upotrebljavaju za sintezu govornog signala.

CELP algoritmi općenito su temeljeni na postupku linearne predikcije, koji govorni signal razlaže na ovojnicu kratkotrajnog spektra, oblikovanu vokalnim traktom, i

rezidualni signal predikcije, koji prvenstveno određuje visinu glasa i njegovu zvučnost. Ovojnica kratkotrajnog spektra modelirana je koeficijentima rezultirajućeg LPC, odnosno STP (engl. *Short-Term Prediction*) filtra, $H(z)$, koji opisuje korelacije među susjednim uzorcima govornog signala. Prijenosna funkcija STP filtra opisana je izrazom:

$$H(z) = \frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{m} \hat{a}_i z^{-i}}, \qquad (3.1)$$

gdje $\hat{a}_i$, $i = 1, \ldots, m$, predstavlja kvantizirane koeficijente LPC filtra reda $m = 10$. Rezidualni signal predikcije predstavlja kvazi-periodičnu ili slučajnu pobudu linearnog sustava, koja se u ACELP algoritmu modelira zbrojem doprinosa vektora iz adaptivne (engl. *adaptive codebook*) i algebarske kodne knjige (engl. *algebraic codebook*), skalirani odgovarajućim faktorima pojačanja (engl. *pitch gain* i *codebook gain*).

Vektori adaptivne kodne knjige modeliraju tzv. dugotrajne korelacije (engl. *long-term correlations*) među uzorcima govornog signala udaljenih za period osnovne frekvencije (engl. *pitch period*), odnosno periodičnu strukturu rezidualnog (pobudnog) signala, koja je posebno izražena za vrijeme zvučnih glasova. Drugim riječima, ova kodna knjiga modelira finu strukturu kratkotrajnog spektra i sastoji se samo od zakašnjelih verzija prethodne pobude. Na ovaj način zapravo je implementiran LTP (engl. *long-term prediction*) filtar, odnosno filtar za sintezu osnovne frekvencije (engl. *pitch synthesis filter*), opisan izrazom

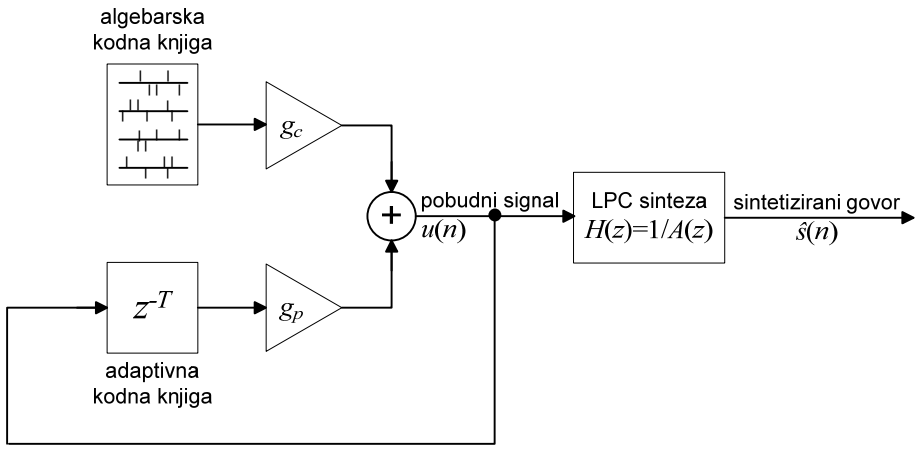
$$\frac{1}{B(z)} = \frac{1}{1 - g_p z^{-T}}, \qquad (3.2)$$

gdje je T osnovna frekvencija (engl. *pitch delay*), a g_p predstavlja faktor pojačanja osnovne frekvencije.

Na vektorima algebarske kodne knjige ostaje zadatak da modeliraju sve preostale strukture u rezidualnom signalu koji nisu modelirani spektralnim modelom vremenski promjenjivih STP i LTP filtara. Uloga algebarske kodne knjige posebno je izražena za vrijeme bezvučnih govornih sekvenci, najviše doprinoseći tijekom frikativnih (tjesnačnih) glasova i tranzijenata.

U skladu s opisanim principom, govorni signal se primjenom ACELP algoritma modelira izračunom sljedećih parametara:

- koeficijente LPC modela (a_i)
- indeks adaptivne kodne knjige (period osnovne frekvencije, T)
- faktor pojačanja osnovne frekvencije (g_p)
- indeks algebarske kodne knjige
- faktor pojačanja algebarske kodne knjige (g_c).



Slika 3.1 *Pojednostavljeni blok dijagram modela za sintezu signala u ACELP algoritmu.*

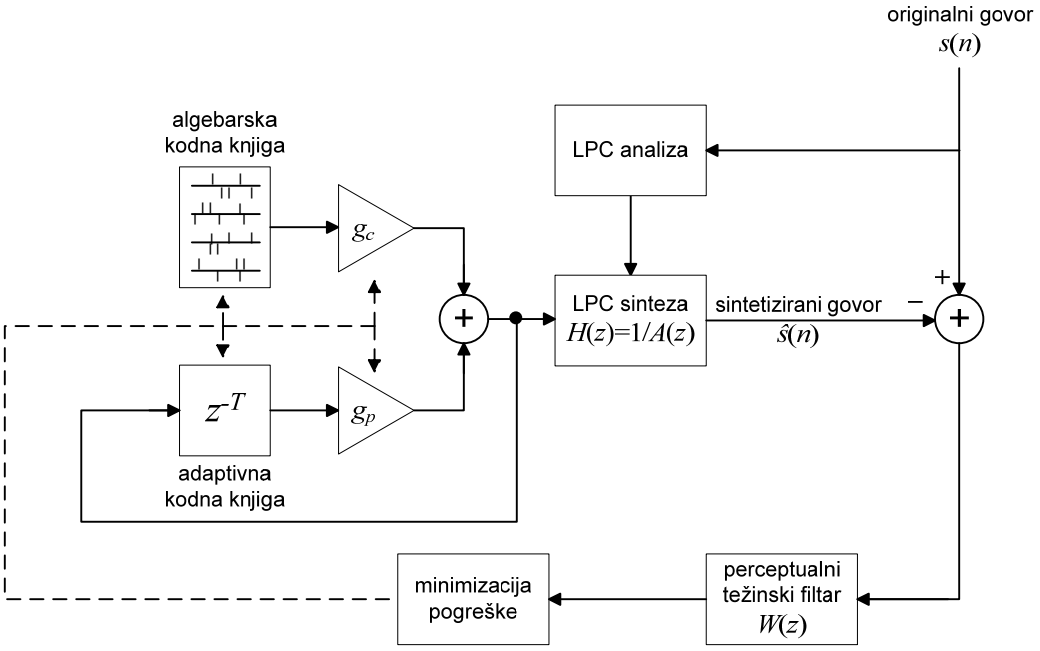
Iz navedenih parametara moguće je, prema pojednostavljenom blok dijagramu na slici 3.1, napraviti sintezu (rekonstrukciju) govornog signala. Pobuda LPC filtra konstruira se zbrajanjem odgovarajućih vektora pobude iz adaptivne i algebarske kodne knjige. Odabrani vektori se prije zbrajanja skaliraju odgovarajućim faktorima pojačanja. Filtracija konstruirane pobude LPC filtrom rezultira sintetiziranim govornim signalom.

Za određivanje navedenih parametara govornog signala, ACELP algoritam primjenjuje princip analize sintezom (engl. *Analysis-by-Synthesis*, AbS), što podrazumijeva proces analize signala temeljen na optimizaciji njegovih parametara u zatvorenoj petlji, kako bi dobili dekodirani (sintetizirani) signal što sličniji valnom obliku originalnog govornog signala. Testiranje svih mogućih kombinacija navedenih parametara, u svrhu odabira najbolje od njih, koja predloženim modelom daje rekonstruirani signal najsličniji originalnom, bilo bi izrazito zahtjevno u smislu računske složenosti algoritma. Iz tog je razloga postupak analize podijeljen u više sekvencijalnih pretraga, koje pojedine parametre modela, poput koeficijenata LPC filtra, određuju u otvorenoj petlji.

Postupak analize govornog signala CELP algoritmom, koji za modeliranje periodičnih struktura rezidualnog signala koristi adaptivnu kodnu knjigu, prikazan je na slici 3.2. Nakon LPC analize, odabir optimalnih parametara pobude radi se minimizacijom signala pogreške $e(n)$, između originalnog $s(n)$ i sintetiziranog govornog signala $\hat{s}(n)$. Prije minimizacije, signal pogreške filtrira se težinskim filtrom $W(z)$ koji oblikovanjem šuma u frekvencijskoj domeni pokušava smanjiti percepciju šuma, maskirajući ga u područje formanata kratkotrajnog spektra. Težinski filtar opisan je izrazom

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}, \qquad (3.3)$$

pri čemu su $0 < \gamma_2 < \gamma_1 \leq 1$ perceptualni težinski faktori, gdje je kod AMR kodera $\gamma_1 = 0.9$ za modove 12k2 i 10k2, te $\gamma_1 = 0.94$ za sve ostale modove, dok je $\gamma_2 = 0.6$ za sve modove rada. Potrebno je napomenuti da se težinski filtar izračunava za svaki pod-okvir iz nekvantiziranih i interpoliranih parametara LPC modela.



Slika 3.2 *Blok dijagram generičkog modela CELP kodera koji koristi adaptivnu kodnu knjigu za modeliranje periodičnih struktura rezidualnog signala.*

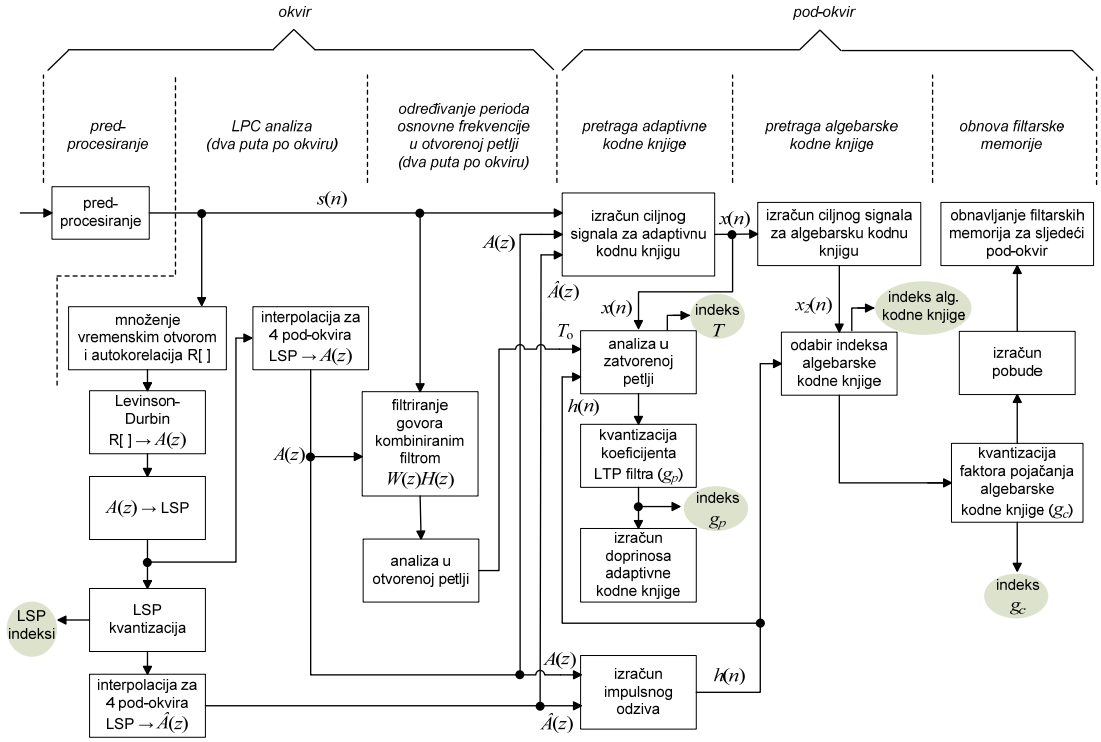
3.1.1. Struktura AMR kodera

Struktura AMR kodera prikazana je na slici 3.3, koja ujedno prikazuje slijed postupaka prilikom kodiranja jednog vremenskog okvira govornog signala. Kako je svaki okvir podijeljen na 4 pod-okvira duljine 40 uzoraka (5 ms), slika odvojeno prikazuje postupke koje se provode na razini okvira, te postupke koje se provode na razini svakog pod-okvira.

Upotrebom autokorelacijske metode, LPC analiza provodi se dva puta u jednom okviru za môd 12k2, te jedanput po okviru za sve ostale modove rada. LPC parametri izračunavaju se primjenom Levinson-Durbin algoritma, te se prevode u LSP domenu radi kvantizacije i interpolacije. U môdu 12k2, dva skupa koeficijenata LPC filtra skupno se kvantiziraju primjenom SMQ, dok se u svim ostalim modovima izračunati skup LPC parametara kvantizira primjenom SVQ. Kvantizirani i nekvantizirani LSP vektori koriste se za drugi i četvrti pod-okvir u 12k2 môdu, odnosno za četvrti pod-okvir u svim ostalim modovima. U ostalim pod-okvirima koriste se linearne interpolacije LSP parametara odgovarajućih pod-okvira trenutnog i prošlog okvira. Nakon interpolacije, kvantizirani i nekvantizirani LSP parametri svih pod-okvira prevode se natrag u LPC koeficijente radi konstrukcije LPC i težinskog percepcijskog filtra za svaki pod-okvir.

Parametri adaptivne i algebarske kodne knjige određuju se i kodiraju za svaki pod-okvir govornog signala. Računska složenost postupka za određivanje perioda osnovne frekvencije pojednostavljen je izvođenjem u dva koraka. U prvom koraku se iz govornog signala, filtriranog težinskim percepcijskim filtrom, analizom u otvornoj petlji (engl.

open-loop pitch analysis) ograničava područje pretrage perioda osnovne frekvencije na mali broj kandidata, koji se onda u drugom koraku pretražuju analizom u zatvorenoj petlji (engl. *closed-loop pitch analysis*). Analiza u otvorenoj petlji radi se za svaki drugi pod-okvir u svim modovima, osim za 5k15 i 4k75 modove u kojima se provodi jednom za svaki okvir. Analiza u zatvorenoj petlji, ovisno o môdu rada, u pretrazi koristi rezoluciju od 1/6 ili 1/3 perioda uzorkovanja. Postupak se provodi za svaki pod-okvir na temelju ciljnog signala $x(n)$ i impulsnog odziva $h(n)$ kombinacije LPC i težinskog percepcijskog filtra $W(z)H(z)$, koji se također izračunavaju za svaki pod-okvir govornog signala. Ciljni signal $x(n)$ izračunava se filtriranjem rezidualnog signala linearne predikcije kombinacijom LPC i težinskog percepcijskog filtra $W(z)H(z)$, čija su inicijalna stanja obnovljena filtracijom razlike (pogreške) između rezidualnog signala predikcije i nulte pobude. Ovakav izračun ciljnog signala ekvivalentan je uobičajenom pristupu, koji ciljni signal izračunava oduzimanjem odziva kombiniranog filtra na nultu pobudu od govornog signala, filtriranog tim istim težinskim percepcijskim filtrom. Rezultat analize u zatvorenoj petlji je odabir faktora pojačanja perioda osnovne frekvencije, odnosno koeficijenta LTP filtra (g_p), i frakcionalnog perioda osnovne frekvencije (T) za koji se interpolacijom prošle pobude izračunava vektor adaptivne kodne tablice. Slijedi kodiranje perioda osnovne frekvencije, koji se npr. za môd 12k2 kodira sa 9 bitova za prvi i treći pod-okvir, dok se za ostale pod-okvire sa 6 bitova kodira njegova relativna promjena. Faktor pojačanja adaptivne kodne knjige se, ovisno o môdu rada, kodira sa 4 bita samostalno (u 12k2 i 7k95 modovima) ili sa 6-7 bitova skupno (VQ) sa faktorom pojačanja algebarske kodne knjige.



Slika 3.3 *Pojednostavljeni blok dijagram AMR kodera.*

Oduzimanjem doprinosa adaptivne kodne knjige od ciljnog signala $x(n)$, za svaki pod-okvir dobiva se novi ciljni signal $x_2(n)$, koji se koristi za pretragu algebarske kodne knjige. Variranjem broja pulseva u svakom pod-okviru od 10 do 2, postiže se najveći udio u redukciji prijenosne brzine za sporije modove rada. Pretraga algebarske kodne knjige obavlja se minimizacijom srednje kvadratne pogreške između polaznog govornog signala, filtriranog težinskim percepcijskim filtrom, i sintetiziranog signala, dobivenog filtriranjem pobude težinskim LPC filtrom. Slijedi kvantizacija faktora pojačanja algebarske kodne knjige primjenom MA predikcije (u svim modovima rada), koji se u 12k2 i 7k95 modovima kodira samostalno s 5 bitova, dok se u ostalim modovima rada kodira skupno s faktorom pojačanja adaptivne kodne knjige.

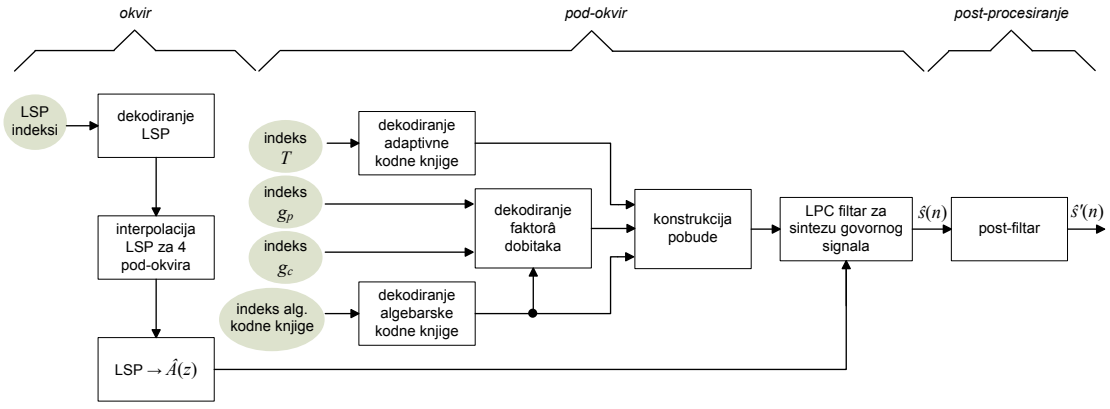
Na kraju, stanja LPC i težinskog percepcijskog filtra obnavljaju se filtriranjem upravo određene pobude trenutnog pod-okvira za izračun ciljnog signala u sljedećem pod-okviru.

3.1.2. Struktura AMR dekodera

Uloga dekodera sastoji se od dekodiranja primljenih parametara, te rekonstrukciji govornog signala njihovom sintezom. Struktura AMR dekodera, koja ujedno prikazuje slijed postupaka prilikom dekodiranja jednog vremenskog okvira govornog signala, prikazana je na slici 3.4. Parametri ACELP modela izdvajaju se iz primljenog bitovnog niza.

Preko dekodiranih primljenih indeksa pojedinih pod-matrica/pod-vektora SMQ/SVQ, iz odgovarajućih tablica konstruiraju se kvantizirani LSF vektor(i) za svaki okvir. Nakon interpolacije, LSF vektori prevode se u koeficijente LPC filtara pojedinih pod-okvira.

Za svaki pod-okvir, iz primljenog indeksa perioda osnovne frekvencije izračunava se njegov cijeli i frakcionalni dio. Dobiveni period upotrebljava se za izračun vektora adaptivne kodne knjige interpolacijom prošle pobude $u(n)$. Nakon dekodiranja, za svaki pod-okvir se iz primljenog indeksa konstruira vektor algebarske kodne tablice, izdvajanjem pozicija i amplituda pobudnih pulseva. Ovisno o môdu rada, slijedi dekodiranje faktora pojačanja adaptivne i algebarske kodne tablice, za svaki pod-okvir govornog signala.

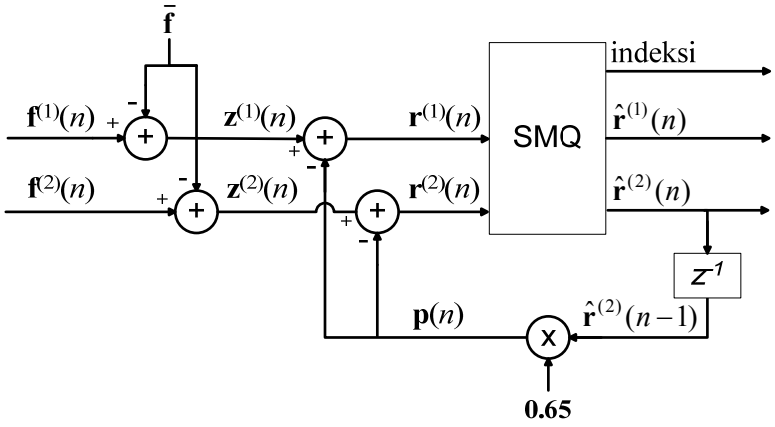


Slika 3.4 *Pojednostavljeni blok dijagram AMR dekodera.*

Za svaki pod-okvir, zbrajanjem skaliranih vektora adaptivne i algebarske kodne tablice formira se vektor pobude duljine 40 uzoraka, koji nakon filtriranja LPC filtrom daje rekonstruirani govorni signal $\hat{s}(n)$.

3.2. Kodiranje spektralne ovojnice u referentnom AMR koderu

Ovisno o brzini prijenosa, estimacija i kvantizacija LP koeficijenata provodi se jednom ili dva puta u jednom okviru, što daje učestalost LPC analize od 50 ili 100 puta u sekundi. U svrhu kvantizacije spektralne ovojnice, AMR koder koristi LSF vektore koji se izračunavaju iz LPC modela [1]. Oduzimanjem srednje vrijednosti LSF vektorima, te primjenom MA vektorske predikcije prvog reda na dobivenu razliku, uklanja se dio među-okvirne redundancije među susjednim okvirima. Pogreška MA predikcije se dalje kvantizira SMQ ili SVQ metodom, ovisno o môdu rada AMR kodera. Dakle, u svrhu kvantizacije spektralne ovojnice, AMR koder zapravo kvantizira pogrešku MA vektorske predikcije, koja se u nastavku ovog rada uglavnom naziva rezidualom, odnosno rezidualnim vektorom.



Slika 3.5 *Blok shema postupka kvantizacije LSF vektora u AMR koderu (môd 12k2).*

Slika 3.5 prikazuje blok shemu postupka kvantizacije LSF vektora u môdu rada s najvećom brzinom prijenosa (12.2 kbit/s), u kojem AMR koder izračunava dva skupa LSF parametara za svaki okvir od 160 uzoraka govornog signala. Dva susjedna vektora pogreške MA predikcije kvantiziraju se SMQ metodom kao matrica dimenzija 10x2 (svaki stupac odgovara jednom vektoru), ali u grupama od samo 2 retka. Pogreška MA vektorske predikcije $\mathbf{r}^{(1)}(n)$ i $\mathbf{r}^{(2)}(n)$ dana je izrazima:

$$\begin{aligned} \mathbf{r}^{(1)}(n) &= \mathbf{z}^{(1)}(n) - \mathbf{p}(n), \\ \mathbf{r}^{(2)}(n) &= \mathbf{z}^{(2)}(n) - \mathbf{p}(n), \end{aligned} \qquad (3.4)$$

gdje $\mathbf{z}^{(1)}(n)$ i $\mathbf{z}^{(2)}(n)$ predstavljaju odgovarajuće razlike LSF vektora $\mathbf{f}^{(1)}(n)$ i $\mathbf{f}^{(2)}(n)$, i njihove srednje vrijednosti $\bar{\mathbf{f}}$ u okviru n. $\mathbf{p}(n)$ predstavlja predikciju LSF vektora u okviru n pri čemu se koristi MA predikcija prvog reda:

$$\mathbf{p}(n) = 0.65\hat{\mathbf{r}}^{(2)}(n-1)\,, \qquad (3.5)$$

gdje je $\hat{\mathbf{r}}^{(2)}(n-1)$ drugi kvantizirani rezidual iz prethodnog okvira.

Podjelom matrice [$\mathbf{r}^{(1)}$ $\mathbf{r}^{(2)}$] dobiva se pet pod-matrica dimenzija 2×2 (po dva elementa iz svakog vektora pogreške), koje se sukladno tablici 3.1 kvantiziraju redom sa 7, 8, 8+1, 8 i 6 bita, tj. sa ukupno 38 bitova/okviru. Drugim riječima, prva pod-matrica sastoji se od prva dva elementa iz prvog i prva dva elementa iz drugog vektora pogreške MA predikcije. Ta četiri elementa kvantiziraju se kao matrica 2×2 koristeći 7 bita. Druga pod-matrica kvantizira se s 8 bita, itd. Treća pod-matrica, za razliku od ostalih, koristi kodnu knjigu od 256 riječi s predznakom, te jedan bit koristi za predznak vrijednosti s odabranog indeksa.

Tablica 3.1 *Alokacija bitova po pojedinim pod-matricama u AMR koderu (môd 12k2).*

môd (kbit/s)	*pod-matrica 1 (bitovi)*	*pod- matrica 2 (bitovi)*	*pod- matrica 3 (bitovi)*	*pod- matrica 4 (bitovi)*	*pod- matrica 5 (bitovi)*	*ukupno (bitova/okviru)*
12.2	7	8	9	8	6	38

Proces matrične kvantizacije koristi težinsku LSP mjeru distorzije, koja se za ulazni LSF vektor **f** svodi na određivanje indeksa kodne knjige *k* sa pripadajućim kvantiziranim vektorom $\hat{\mathbf{f}}^k$, koji minimizira izraz:

$$E_{LSP}(\hat{\mathbf{f}}^k) = \sum_{i=1}^{10}\left[f_i w_i - \hat{f}_i^k w_i\right]^2 . \qquad (3.6)$$

Težinski faktori w_i, i=1,..., 10, dani su izrazom

$$w_i = 3.347 - \frac{1.547}{450} d_i \text{ za } d_i < 450,$$

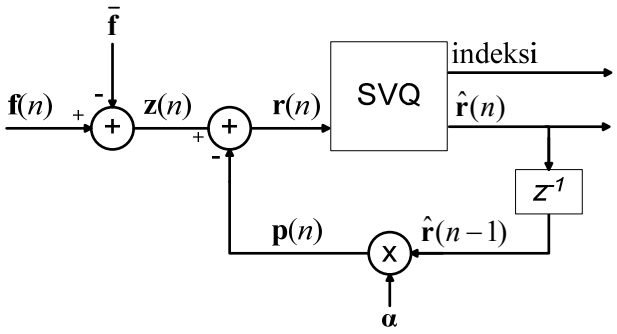
$$= 1.8 - \frac{0.8}{1050}\left(d_i - 450\right) \text{ inače,} \qquad (3.7)$$

gdje je $d_i = f_{i+1} - f_{i-1}$, $f_0 = 0$ i $f_{11} = 4000$.

U SMQ se gore definirana težinska LSP mjera distorzije primjenjuje za kvantizaciju svih pet pod-matrica koje sadrže odgovarajuće komponente oba vektora pogreške MA predikcije. Npr., kvantizacija prve pod-matrice provodi se određivanjem indeksa *k* odgovarajuće kodne knjige, sa pripadajućim odgovarajućim kvantiziranim komponentama vektorâ pogreške predikcije $\hat{\mathbf{r}}^{(1)k}$ i $\hat{\mathbf{r}}^{(2)k}$, koji minimizira izraz

$$E_{LSP}(\hat{\mathbf{r}}^{(1)k}, \hat{\mathbf{r}}^{(2)k}) = \sum_{i=1}^{2}\left[\left(r_i^{(1)} w_i^{(1)} - \hat{r}_i^{(1)k} w_i^{(1)}\right) + \left(r_i^{(2)} w_i^{(2)} - \hat{r}_i^{(2)k} w_i^{(2)}\right)\right]^2 , \qquad (3.8)$$

gdje $\mathbf{w}^{(1)}$ i $\mathbf{w}^{(2)}$ predstavljaju vektore težinskih faktora, koji se izračunavaju posebno za svaki ulazni LSF vektor prema izrazu 3.7 i koriste s odgovarajućim vektorima pogreške MA predikcije prilikom kvantizacije svih pod-matrica u jednom okviru.



Slika 3.6 *Blok shema postupka kvantizacije LSF vektora u AMR koderu (svi môdovi osim 12k2).*

U svim ostalim môdovima rada, AMR koder računa samo jedan skup LSF parametara u jednom govornom okviru. Slika 3.6 prikazuje odgovarajuću blok shemu kvantizacijskog postupka. Slično môdu s najvećom brzinom prijenosa, vektor pogreške MA predikcije kvantizira se SVQ metodom, dijeljenjem na tri pod-vektora. Pogreška MA vektorske predikcije $\mathbf{r}(n)$ dana je izrazom:

$$\mathbf{r}(n) = \mathbf{z}(n) - \mathbf{p}(n)\,, \qquad (3.9)$$

gdje $\mathbf{z}(n)$ predstavlja razliku LSF vektor $\mathbf{f}(n)$ u okviru n umanjen za srednju vrijednost LSF vektora $\bar{\mathbf{f}}$. $\mathbf{p}(n)$ predstavlja predikciju LSF vektora u okviru n pri čemu se također koristi MA predikcija prvog reda:

$$p_j(n) = \alpha_j \hat{r}_j(n-1) \quad j = 1,...,10\,, \qquad (3.10)$$

gdje je $\hat{\mathbf{r}}(n-1)$ kvantizirani rezidual iz prethodnog okvira i α_j predikcijski faktor j-te komponente LSF vektora ($\boldsymbol{\alpha}$=[0.291626; 0.328644; 0.383636; 0.405640; 0.438873; 0.355560; 0.323120; 0.298065; 0.262238; 0.197876]).

Tablica 3.2 *Alokacija bitova po pod-vektorima u môdovima AMR kodera koji koriste SVQ.*

môd	*brzina (kbit/s)*	*pod-vektor 1 (bitovi)*	*pod-vektor 2 (bitovi)*	*pod-vektor 3 (bitovi)*	*ukupno (bitova/okviru)*
10k2	10.2	8	9	9	26
7k95	7.95	9	9	9	27
7k4	7.40	8	9	9	26
6k7	6.70	8	9	9	26
5k9	5.90	8	9	9	26
5k15	5.15	8	8	7	23
4k75	4.75	8	8	7	23

Tri pod-vektora, dimenzija 3, 3 i 4, kvantiziraju se redom, rezolucijom od 7 do 9 bita sukladno tablici 3.2. Npr., u môdu s brzinom prijenosa 10.2 kbit/s, prvi pod-vektor sadrži prve tri komponente vektora pogreške MA predikcije koje se vektorski kvantiziraju 8-bitnom rezolucijom, drugi pod-vektor sadrži sljedeće tri komponente i vektorski kvantizira 9-bitnom rezolucijom, dok zadnje četiri komponente pripadaju

trećem pod-vektoru koji se kvantizira s 9 bita. U zbroju se, u navedenom môdu, za kvantizaciju vektora pogreške MA predikcije SVQ metodom koristi ukupno 26 bita.

Rani radovi iz područja kodiranja spektralne ovojnice pokazuju da postoji izrazita korelacija među LSF vektorima susjednih okvira (među-okvirna korelacija), te među susjednim komponentama LSF vektora pojedinog okvira (unutar-okvirna korelacija) [6]. Kao posljedica dijeljenja LSF vektora na pod-matrice, odnosno pod-vektore, SMQ i SVQ metode nisu u mogućnosti iskoristi unutar-okvirne korelacije među susjednim komponentama LSF vektora u cijelosti.

4. Modeli Gaussovih mješavina i njihova primjena u vektorskoj kvantizaciji

Kako je već spomenuto u poglavlju 2.2.4, pretpostavka o Gaussovoj razdiobi vjerojatnosti vektorskog procesa koji se kvantizira, čini Karhunen-Loève transformaciju optimalnom u smislu najmanje distorzije [9]. Drugim riječima, pretpostavka o izrazito uniformnoj korelaciji među komponentama vektorâ kroz cijeli vektorski prostor omogućila bi dekorelaciju svih vektora jednom globalnom matricom transformacije. Činjenica da funkcija gustoće vjerojatnosti realnih signala rijetko odgovara funkciji gustoće vjerojatnosti Gaussove razdiobe neminovno utječe na degradaciju svojstava skalarnih kvantizatora transformacijskog kodera, projektiranih pod tom pretpostavkom.

Kao alternativa pretpostavci Gaussove razdiobe vjerojatnosti vektorskog procesa, funkcija gustoće vjerojatnosti može se proizvoljno točno modelirati upotrebom modela s Gaussovom mješavinom. Na taj način, distribuciju realnog vektorskog procesa moguće je definirati kao težinsku sumu, odnosno mješavinu više preklapajućih procesa s Gaussovom razdiobom. Dizajn vektorskog kvantizatora, temeljen na dobivenoj aproksimaciji, rezultira boljim svojstvima u usporedbi s kvantizatorom čiji je postupak projektiranja temeljen na aproksimaciji razdiobe vektorskog procesa standardnom funkcijom poput Gaussove.

Općenito se GMM može koristiti za aproksimaciju proizvoljnih funkcija gustoće vjerojatnosti i, kao takav, primjenu nalazi u mnogim aplikacijama, primjerice u aplikacijama za prepoznavanje govora [35] i identifikaciju govornika [25].

4.1. Aproksimacija funkcije gustoće vjerojatnosti primjenom modela s Gaussovim mješavinama

Proizvoljna funkcija gustoće vjerojatnosti d-dimenzionalnih vektora x može se aproksimirati linearnom kombinacijom (mješavinom) m višedimenzionalnih funkcija gustoće Gaussove razdiobe vjerojatnosti (komponenti mješavine):

$$G(x \mid \boldsymbol{\Theta}) = \sum_{i=1}^{m} \rho_i \mathcal{N}(x; \mu_i; \mathbf{C}_i), \tag{4.1}$$

$$\boldsymbol{\Theta} = \left[m, \rho_1, ..., \rho_m, \mu_1, ..., \mu_m, \mathbf{C}_1, ..., \mathbf{C}_m\right], \tag{4.2}$$

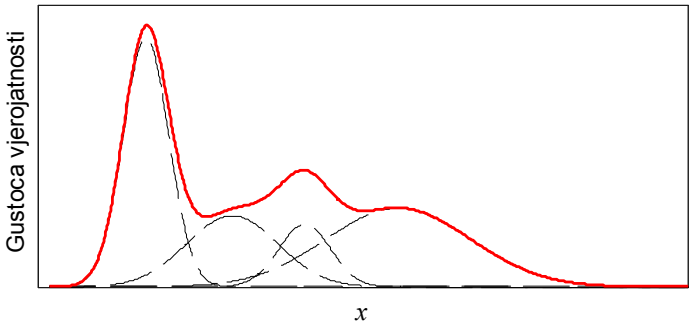
gdje je x ulazni vektor, $\boldsymbol{\Theta}$ predstavlja parametarski model aproksimirane funkcije gustoće vjerojatnosti, $\mathcal{N}(x;\mu;\mathbf{C})$ je višedimenzionalna funkcija gustoće Gaussove razdiobe vjerojatnosti:

$$\mathcal{N}(x;\mu;\mathbf{C}) = \frac{1}{\sqrt{(2\pi)^d \, det(\mathbf{C})}} \mathrm{e}^{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\mathbf{C}^{-1}(x-\mu)}, \qquad (4.3)$$

sa parametrima μ i $\mathbf{C}$ koji predstavljaju vektor srednje vrijednosti i matricu kovarijance, dok ρ_i predstavlja težinu i-te komponente mješavine uz uvjet

$$\sum_{i=1}^{m} \rho_i = 1. \qquad (4.4)$$

Izraz 4.1 može se promatrati kao funkcijska dekompozicija nepoznate *pdf* funkcije na parametarski opisane funkcije gustoće koje predstavljaju bazne funkcije ove dekompozicije (slika 4.1). Točnost i primjenjivost aproksimacije prvenstveno ovisi o kompromisu (engl. *trade-off*) oko broja komponenata mješavine kojim se aproksimira funkcija gustoće vjerojatnosti izvora. Veći m značio bi mogućnost točnije aproksimacije konkretne *pdf* funkcije izvora, ali i veću složenost algoritma kodiranja, koja direktno ovisi o broju komponenti mješavine. Međutim, preveliki m bi aproksimaciju mogao učiniti pretjerano prilagođenom ograničenom broju vektora u trening bazi (engl. *overfitting*), čime bi aproksimacija zapravo odražavala slučajna svojstva tih vektora i time izgubila na općenitosti i primjenjivosti za reprezentaciju statističkih svojstava danog izvora.



Slika 4.1 *Aproksimacija funkcije gustoće vjerojatnosti jednodimenzionalnog procesa primjenom modela s Gaussovim mješavinama.*

Za danu trening bazu, parametri modela, $\boldsymbol{\Theta}$, najčešće se određuju primjenom tzv. EM (engl. *expectation-maximization*) algoritma [4]. Spomenuti algoritam, nakon inicijalizacije, iterativno izračunava parametre modela primjenom kriterija maksimalne vjerodostojnosti. Pri tome se težinska suma individualnih Gaussovih *pdf* funkcija promatra kao jedna *pdf* funkcija određene strukture, čiji se parametri primjenom EM algoritma pokušavaju poklopiti sa statistikom realizacije vektorskog procesa.

4.2. EM algoritam

U praksi postoje situacije u kojima promatranjem nije moguće spoznati sve relevantne atribute podataka. U tom slučaju govorimo o problemu podataka koji nedostaju (engl. *missing data*), odnosno o problemu skrivenih podataka (engl. *hidden data*). Na sličan način možemo promatrati i modele Gaussovih mješavina, kod kojih je vjerojatnost da realizirani vektor x pripada komponenti mješavine i jednaka težini komponente ρ_i. Podatak koji u ovom slučaju nedostaje pri modeliranju nepoznate *pdf* funkcije GMM modelom je zapravo informacija o komponenti mješavine kojoj pripada pojedini vektor iz realizacije slučajnog vektorskog procesa. EM algoritam predstavlja jednostavan, stabilan i učinkovit iterativni postupak koji parametre modela u takvim okolnostima estimira na temelju kriterija maksimalne vjerodostojnosti promatrane realizacije procesa. Drugim riječima, svaka iteracija poboljšava estimirane parametre modela, povećavajući vjerojatnost da su vektori konkretne realizacije procesa generirani upravo modelom čiji se parametri estimiraju ovim algoritmom. EM algoritam svoju primjenu nalazi prvenstveno u situacijama kad je parametre modela teško ili nemoguće izraziti eksplicitnim (engl. *closed-form*) analitičkim izrazima.

Svaka iteracija algoritma izvodi se u dva koraka: u E-koraku (engl. *expectation step*, E-step) se pomoću uvjetnog očekivanja (engl. *conditional expectation*) na temelju poznate realizacije procesa i trenutne estimacije parametara modela estimira vrijednost podataka koji nedostaju, dok se u M-koraku (engl. *maximization step*, M-step) maksimizira funkcija vjerojatnosti pod pretpostavkom da su podaci koji nedostaju poznati (iz procjene u E-koraku). Izvođenjem nad istom realizacijom procesa (npr. baza trening vektora), algoritam garantira monotono povećanje vjerojatnosti u svakoj iteraciji, što osigurava njegovu konvergenciju ka lokalnom maksimumu.

Ako sa $\mathbf{X}$ označimo slučajnu vektorsku varijablu opisanu proizvoljnom *pdf* funkcijom koju nastojimo parametrizirati, onda EM algoritam nastoji estimirati vrijednosti parametarskog skupa, $\boldsymbol{\Theta}$, tako da gustoća vjerojatnosti $P(\mathbf{X}|\boldsymbol{\Theta})$ bude maksimalna na mjestu realiziranih vrijednosti. Kod određivanja parametarskog skupa, vjerojatnost obično poprima veoma male vrijednosti, te se uvodi funkcija logaritamske vjerodostojnosti (engl. *log-likelihood function*), definirana kao:

$$LL(\boldsymbol{\Theta}) = \log P(\mathbf{X} \mid \boldsymbol{\Theta}). \qquad (4.5)$$

Maksimum funkcije u izrazu 4.5, pod pretpostavkom analitičke funkcije logaritamske vjerodostojnosti, teoretski je moguće izračunati iz uvjeta da je njena prva derivacija jednaka nuli

$$\frac{\partial}{\partial \boldsymbol{\Theta}} LL(\boldsymbol{\Theta}) = 0. \qquad (4.6)$$

Pod pretpostavkom međusobne nezavisnosti ishodâ (vektorâ) slučajne varijable, logaritamska vjerodostojnost cijele trening baze, koja sadrži s vektora, jednaka je umnošku gustoće vjerojatnosti svih vektora u bazi, pa u slučaju aproksimacije GMM modelom imamo:

$$
\begin{aligned}
LL(\boldsymbol{\Theta}) &= \log P(x_1,...,x_s \mid \boldsymbol{\Theta}) \\
&= \log\left(\prod_{n=1}^{s} G(x_n \mid \boldsymbol{\Theta})\right) = \sum_{n=1}^{s} \log\left(G(x_n \mid \boldsymbol{\Theta})\right) \\
&= \sum_{n=1}^{s} \log\left(\sum_{i=1}^{m} \rho_i \mathcal{N}(x_n; \mu_i; \mathbf{C}_i)\right).
\end{aligned}
\tag{4.7}
$$

Ovako definirana funkcija logaritamske vjerodostojnosti se onda koristi kao mjera točnosti aproksimacije *pdf* funkcije GMM modelom. Naravno, što je $LL(\boldsymbol{\Theta})$ veći, to se aproksimacija parametarskim modelom $\boldsymbol{\Theta}$ smatra točnijom. Na žalost, kako izraz 4.7 sadrži logaritam sume, eksplicitne analitičke izraze za optimalne parametre modela teško je, odnosno nemoguće izvesti. U ovakvom slučaju pribjegava se primjeni EM algoritma koji iterativnim izvođenjem nad istom realizacijom garantira monotono povećanje logaritamske vjerodostojnosti u svakoj iteraciji, tj.

$$
LL(\boldsymbol{\Theta}^{(k+1)}) \geq LL(\boldsymbol{\Theta}^{(k)}),
\tag{4.8}
$$

gdje $\boldsymbol{\Theta}^{(k)}$ predstavlja vrijednost parametara modela u iteraciji k. Prema [4], nakon inicijalizacije, iterativna estimacija parametara modela odvija se izmjenom E-koraka i M-koraka u svakoj iteraciji prema sljedećim izrazima:

- ***E-korak*** – u koraku k+1 računa se *a posteriori* vjerojatnost pripadanja i-tom klasteru za svih s vektora i m klastera, pod uvjetom da su poznati parametri modela i realizacija procesa

$$
p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right) = \frac{\rho_i^{(k)} \mathcal{N}(x_n; \mu_i^{(k)}, \mathbf{C}_i^{(k)})}{\sum_{j=1}^{m} \rho_j^{(k)} \mathcal{N}(x_n; \mu_j^{(k)}, \mathbf{C}_j^{(k)})},
\tag{4.9}
$$

gdje je $i = 1,2,...,m$ i $n = 1,2,...,s$.

- ***M-korak*** – u koraku k+1, na osnovu *a posteriori* vjerojatnosti iz E-koraka ponovno se estimiraju srednje vrijednosti, matrice kovarijance i težinski koeficijenti komponenata mješavine prema izrazima

$$
\rho_i^{(k+1)} = \frac{1}{s} \sum_{n=1}^{s} p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right),
\tag{4.10}
$$

$$
\mu_i^{(k+1)} = \frac{\sum_{n=1}^{s} p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right) x_n}{\sum_{n=1}^{s} p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right)},
\tag{4.11}
$$

$$\mathbf{C}_i^{(k+1)} = \frac{\sum_{n=1}^{s} p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right)\left(x_n - \mu_i^{(k+1)}\right)\left(x_n - \mu_i^{(k+1)}\right)^T}{\sum_{n=1}^{s} p\left(i \mid x_n, \boldsymbol{\Theta}^{(k)}\right)}. \qquad (4.12)$$

Inicijalizaciju parametara (za $k = 0$) moguće je provesti primjenom Linde-Buzo-Gray (LBG) algoritma [22] nad vektorima trening baze, koja predstavlja distribuciju izvora. Rezultat inicijalizacije je m inicijalnih klastera opisanih srednjim vrijednostima, matricama kovarijance i težinama pojedinih klastera. Inicijalizirani parametri modela se onda prosljeđuju u EM algoritam, koji konvergencijom daje konačan skup od m srednjih vrijednosti, matrica kovarijance i težina komponenti mješavine.

Kako EM algoritam, izvođenjem nad istom realizacijom procesa, garantira konvergenciju logaritma vjerojatnosti ka lokalnom maksimumu, iz praktičnih je razloga potrebno ograničiti broj iteracija do točke, kad daljnja reestimacija parametara modela postaje suvišna u smislu povećanja njihove logaritamske vjerodostojnosti. U tu svrhu, kao kriterij završetka optimizacijskog postupka moguće je upotrijebiti relativnu promjenu logaritamske vjerodostojnosti:

$$\Delta LL^{(k)} = \frac{LL^{(k)} - LL^{(k-1)}}{LL^{(k-1)}}. \qquad (4.13)$$

Pad ΔLL ispod unaprijed određene vrijednosti ΔLL_{min} označava završetak EM algoritma. Broj iteracija potrebnih za konvergenciju EM algoritma ovisi o veličini korištene realizacije vektorskog procesa čija se funkcija gustoće modelira, dimenzionalnosti vektorskog procesa, broju komponenti mješavine, te inicijalizaciji GMM parametara. Promatranjem procesa konvergencije, za završetak EM algoritma određena je empirijska vrijednost od $\Delta LL_{min} = 10^{-5}$. Na primjer, realizacija od 56030 10-dimenzionalnih vektora, uz GMM model sa $m = 8$ komponenti mješavine i inicijalizaciju LBG algoritmom, treba svega 30-tak iteracija za zadovoljenje spomenutog kriterija za završetak optimizacijskog postupka.

4.3. Karhunen-Loève transformacija

Poznato je da, pod pretpostavkom kodiranja visokim prijenosnim brzinama [7], KLT predstavlja najbolju transformaciju za kvantizaciju koreliranih izvora s Gaussovom razdiobom vjerojatnosti [16], optimalnu u smislu minimalne distorzije tj. minimalne prosječne kvadratne pogreške.

Općenito su komponente slučajnog vektorskog procesa **X** međusobno korelirane. Za danu funkciju gustoće vjerojatnosti koja opisuje **X** moguće je pronaći ortogonalnu transformacijsku matricu **T** koja ortogonalnom linearnom transformacijom

$$\mathbf{y} = \mathbf{T}\mathbf{x} \qquad (4.14)$$

korelirane komponente vektorskog procesa **X** transformira u vektorski proces **Y** s međusobno nekoreliranim komponentama. Retke takve transformacijske matrice čine ortogonalni vektori koji razapinju transformacijsku (KLT) domenu, a linearna

transformacija s takvim svojstvom naziva se Karhunen-Loève transformacija. KLT zapravo projicira ulazne vektore na novu bazu, koju čine svojstveni vektori matrice kovarijance polaznog procesa. Kako se matrica kovarijance ulaznog procesa, a time i njeni svojstveni vektori, koji se još nazivaju i glavne komponente procesa (engl. *principal components*), izračunava iz ulaznih vektora, to KLT čini ovisnom o realizaciji ulaznog procesa, što zahtijeva njeno ponovno izračunavanje za svaku realizaciju procesa. U praktičnoj primjeni ovisnost o realizaciji ulaznog procesa nije poželjna i predstavlja problem na strani dekodera, koji za inverznu transformaciju treba poznavati matricu kovarijance ili transformacijsku matricu konkretnih podataka koje dekodira. Prijenos takve dodatne informacije zahtijevao bi i dodatne resurse (engl. *overhead*) u transmisijskom sustavu, što je svakako nepoželjno. Alternativa takvom pristupu bila bi korištenje statičke, globalne KLT matrice koja bi se derivirala tijekom procesa učenja, te koristila prilikom kodiranja i dekodiranja. Takvo rješenje bi onda bilo suboptimalno i učinak bi mu ovisio o sličnosti statističkih svojstava podataka koji se kodiraju sa statističkim svojstvima podataka korištenim u procesu učenja (engl. *training data*).

Inverzna transformacija, koja transformirani vektor prevodi natrag u originalne koordinate, dana je izrazom:

$$\mathbf{x} = \mathbf{T}^{-1}\mathbf{y}\,, \tag{4.15}$$

koji se zbog ortogonalnosti redaka transformacijske matrice može pisati kao:

$$\mathbf{x} = \mathbf{T}^{T}\mathbf{y}\,. \tag{4.16}$$

Matrica kovarijance, odnosno korelacijska matrica transformiranog procesa, $\mathbf{C}_\mathbf{y}$, dana je izrazom:

$$\begin{aligned}\mathbf{C}_\mathbf{y} &= E\left[\mathbf{y}\mathbf{y}^T\right] \\ &= E\left[\mathbf{T}\mathbf{x}\left(\mathbf{T}\mathbf{x}\right)^T\right] \\ &= \mathbf{T}E\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{T}^T \\ &= \mathbf{T}\mathbf{C}_\mathbf{x}\mathbf{T}^T,\end{aligned} \tag{4.17}$$

gdje je $\mathbf{C}_\mathbf{x}$ matrica kovarijance polaznog procesa, a $E[\bullet]$ označava operator očekivanja. Dekorelacija komponenti ulaznog procesa podrazumijeva dijagonalnu matricu kovarijance transformiranog procesa. Supstitucijom $\mathbf{Q} = \mathbf{T}^T$, izraz 4.17 možemo prevesti u faktorizaciju matrice kovarijance polaznog procesa u izrazu 4.19:

$$\mathbf{C}_\mathbf{y} = \mathbf{Q}^T\mathbf{C}_\mathbf{x}\mathbf{Q}\,, \tag{4.18}$$

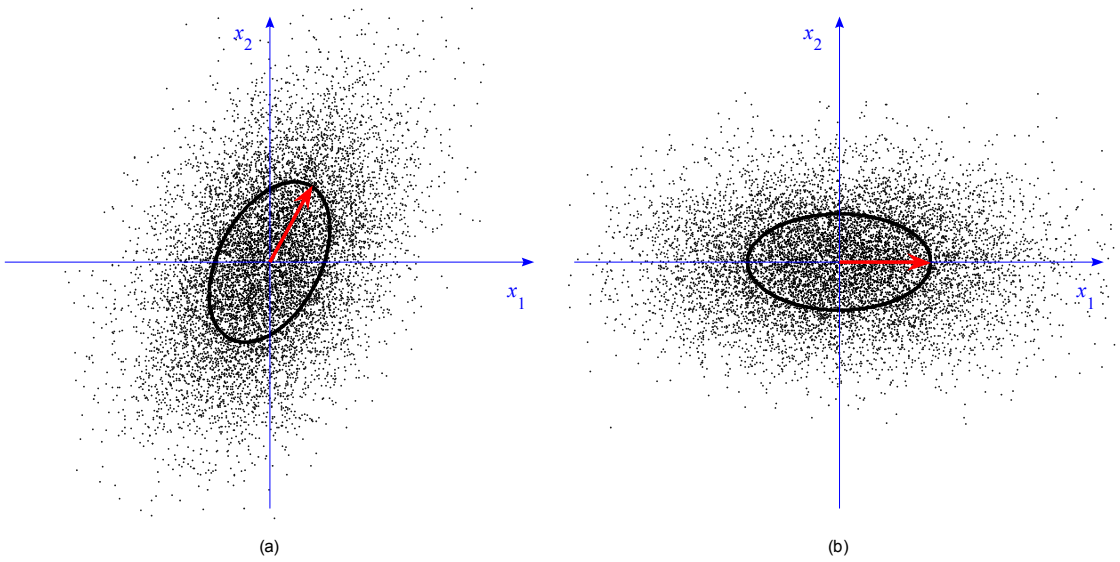
$$\mathbf{C}_\mathbf{x} = \mathbf{Q}\mathbf{C}_\mathbf{y}\mathbf{Q}^T\,, \tag{4.19}$$

gdje je $\mathbf{C}_\mathbf{y}$ dijagonalna matrica, koja na svojoj dijagonali ima silazno poredane svojstvene vrijednosti matrice kovarijance polaznog procesa (od najveće ka najmanjoj),

$\lambda_1 > \lambda_2 > ... > \lambda_d$, koje ujedno odgovaraju varijancama komponenti transformiranog procesa:

$$\mathbf{C}_\mathbf{y} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_d \end{bmatrix}. \qquad (4.20)$$

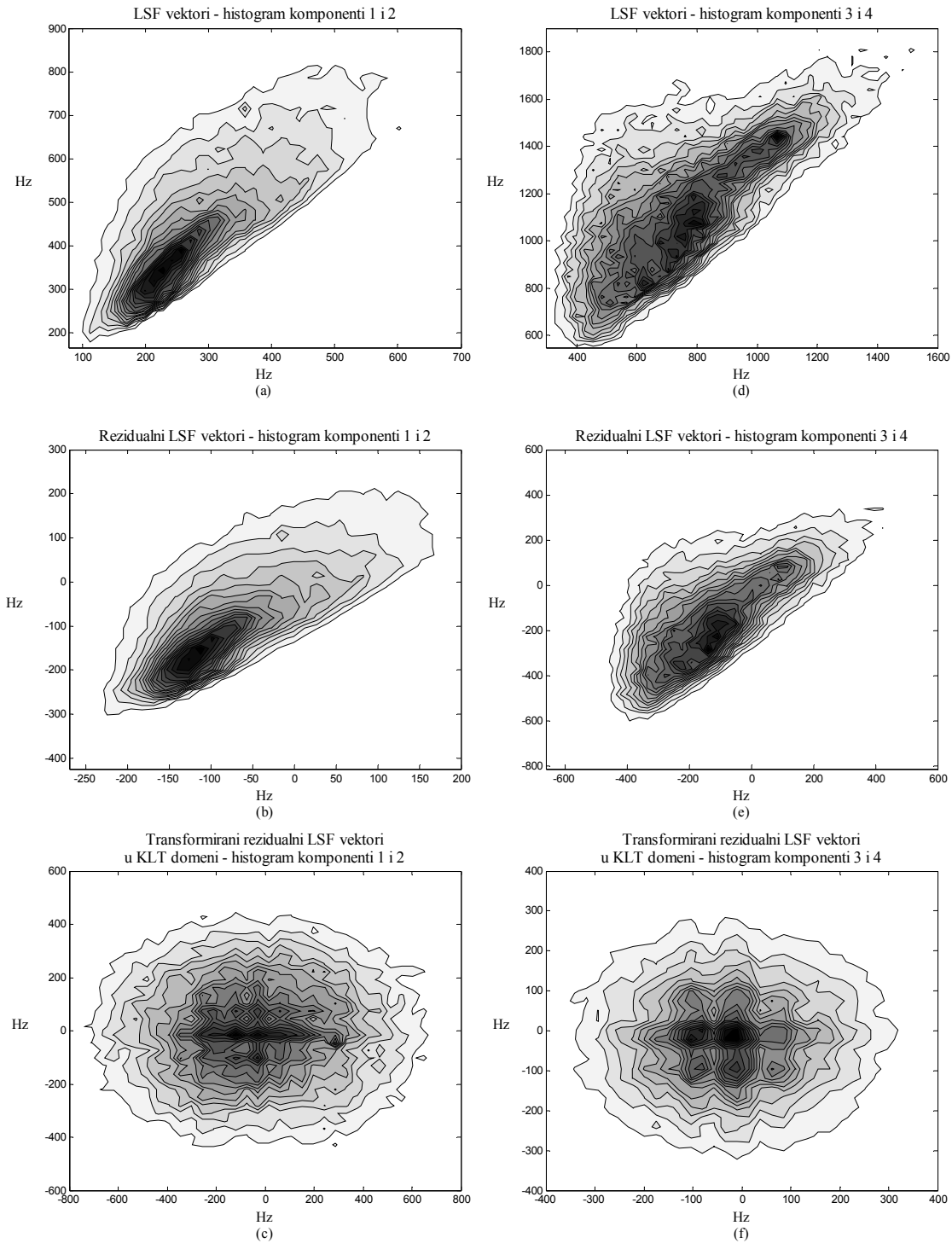
Opisana faktorizacija matrice kovarijance polaznog procesa se u literaturi često naziva metodom dekompozicije svojstvenim vrijednostima (engl. *eigenvalue decomposition*). U istom redoslijedu poredani su i odgovarajući svojstveni vektori u transformacijskoj matrici **T**, gdje se svojstveni vektor s najvećom svojstvenom vrijednošću pojavljuje na vrhu transformacijske matrice.



Slika 4.2 *Ilustracija dekorelacijskog svojstva KLT u 2D prostoru: distribucija vektora a) prije i b) poslije transformacije.*

KLT zapravo nastoji zarotirati vektorski prostor (slika 4.2), tako da smjer svojstvenog vektora s najvećom svojstvenom vrijednošću uskladi s prvom dimenzijom [9]. To će osigurati najveću varijancu prvoj komponenti transformiranog procesa, te opadanje varijance komponenti s povećanjem njihovog rednog broja. Kako se radi o unitarnoj transformaciji koja čuva Euklidsku udaljenost, ukupna varijanca svih komponenti procesa ostaje nepromijenjena nakon transformacije [8], tj. jednaka je zbroju svojstvenih vrijednosti matrice kovarijance polaznog procesa, koje se nalaze na glavnoj dijagonali matrice kovarijance transformiranog procesa:

$$\sum_{i=1}^{d} \sigma_i^2 = \sum_{i=1}^{d} \lambda_i . \qquad (4.21)$$



Slika 4.3 *2D histogrami: (a) i (d) pojedine komponente LSF vektora korištenih u postupku projektiranja GMM VQ (trening baza), (b) i (e) odgovarajuće komponente rezidualnih LSF vektora, (c) i (f) odgovarajuće komponente rezidualnih LSF vektora nakon KLT transformacije. Tamnije boje označavaju područja veće gustoće populacije.*

Posljedično, KLT zapravo koncentrira energiju procesa u nekoliko prvih komponenti transformiranog vektorskog procesa, u smjeru svojstvenih vektora s najvećim svojstvenim vrijednostima.

Na slici 4.3(a) je, za primjer, prikazan 2D histogram dviju najnižih frekvencija spektralnih linija, temeljen na 56030 LSF vektora korištenih u postupku projektiranja GMM vektorskih kvantizatora. Ti vektori zapravo sačinjavaju trening bazu (engl. *training database*) za simulacije čiji su rezultati opisani u poglavlju 6. Uočljiva je izrazita korelacija dvije spomenute komponente LSF vektora. Na slici 4.3(b) prikazan je 2D histogram odgovarajućih komponenti rezidualnih LSF vektora. Vidljivo je da, nakon MA predikcije, komponente rezidualnih vektora zadržavaju međusobnu koreliranost. Slika 4.3(c) prikazuje 2D histogram istih komponenti rezidualnih LSF vektora, nakon transformacije u KLT domenu. Vidljivo je da su komponente transformiranih reziduala dekorelirane, te da za razliku od raspodjele prije transformacije, raspodjela transformiranih reziduala nalikuje Gaussovoj. Slike 4.3(d)-(f) ekvivalentne su opisanim slikama 4.3(a)-(c), s tim da prikazuju odgovarajuće distribucije treće i četvrte komponente odgovarajućih vektora. Usporedbom spomenutih histograma, vidljivo je smanjenje raspona vrijednosti s rednim brojem transformirane komponente, iako komponente 3 i 4 prije transformacije imaju veće raspone vrijednosti od komponenti 1 i 2. To naravno upućuje na KLT svojstvo koncentracije energije u niže komponente transformiranog procesa. Potrebno je napomenuti da je pri transformaciji korišten GMM model s 8 komponenti, te da je svaki rezidualni LSF vektor transformiran korištenjem najpogodnije od 8 KLT transformacijskih matrica, kako je to opisano u poglavlju 4.5.

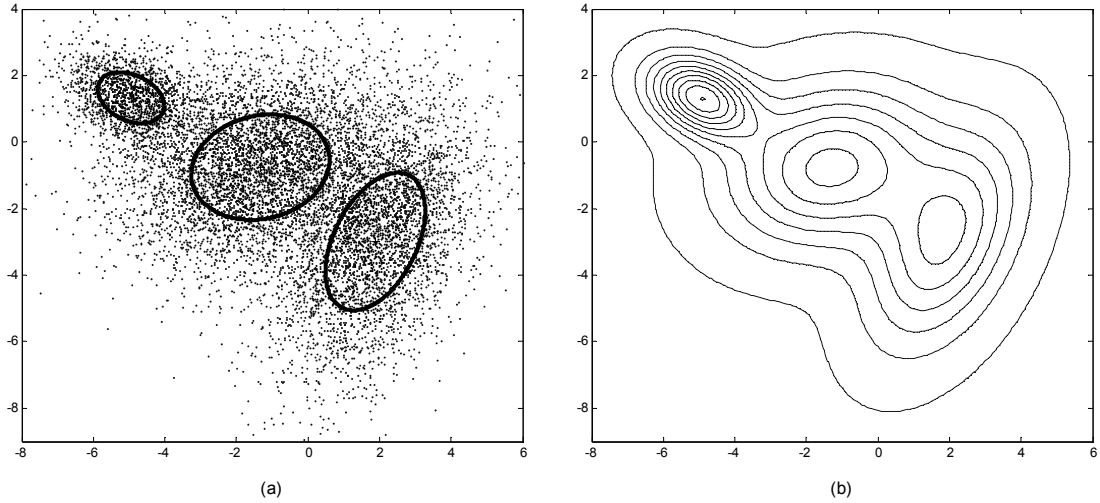
4.4. Skalarna kvantizacija transformiranih komponenti

KLT transformacijom *d*-dimenzionalnih vektora dobiva se *d* transformiranih vektorskih komponenti (skalara) koji se onda neovisno skalarno kvantiziraju. Pod pretpostavkom Gaussove razdiobe ulaznog vektorskog procesa, s uniformnim korelacijama među komponentama kroz cijeli vektorski prostor, nakon primjene KLT svaka komponenta transformiranog procesa predstavlja nezavisnu realizaciju Gaussovog procesa. Ovisno o kvantizacijskoj metodi, kodiranje ovojnice rezultirat će podatkovnim tokom fiksne ili varijabilne brzine. Kako je već spomenuto, transformacijski koderi općenito obuhvaćaju skupinu kodera koji kombiniraju uniformnu skalarnu kvantizaciju komponenti transformiranog procesa sa entropijski ograničenom skalarnom kvantizacijom izlaznih indeksa skalarnih kvantizatora. Takva kombinacija rezultira koderom s izlaznim podatkovnim tokom varijabilne brzine. S druge strane, kao podskupina, odnosno specijalan slučaj transformacijskih kodera, blok kvantizatori [16] koriste neuniformne skalarne kvantizatore i imaju fiksnu izlaznu prijenosnu brzinu. Ako funkcija gustoće vjerojatnosti izvornog procesa nije uniformna, entropijsko kodiranje izlaznih indeksa skalarnih kvantizatora može rezultirati značajnom redukcijom prijenosne brzine. Štoviše, pokazano je da je, bez obzira na funkciju gustoće vjerojatnosti izvornog procesa, pod pretpostavkom kvantizacije visokim brzinama prijenosa (engl. *high-rate quantization*), izlazna entropija uniformnog skalarnog kvantizatora manja od entropije optimalnog nelinearnog skalarnog kvantizatora fiksne prijenosne brzine [29]. Iz toga proizlazi općenito veća učinkovitost entropijskog kodiranja indeksa dobivenih uniformnom

skalarnom kvantizacijom od nelinearne skalarne kvantizacije kakva se primjenjuje u blok-kvantizatorima [13], [29]. Međutim, varijabilna priroda izlaznog podatkovnog toka takvih kvantizatora povećava složenost kvantizacijskog postupka, te uzrokuje probleme s propagacijom pogreške i punjenjem izlaznog spremnika (engl. *buffer underflow / overflow*) [8], [9].

4.5. Vektorska kvantizacija temeljena na GMM-u

Kako je već spomenuto u uvodu ovog poglavlja, razdioba realnih vektorskih procesa rijetko odgovara Gaussovoj. S druge strane, skalarni kvantizatori transformacijskih kodera temeljenih na KLT-u projektirani su pod pretpostavkom nezavisnosti komponenti transformiranog procesa, što za preduvjet ima Gaussovu razdiobu ulaznog vektorskog procesa. Ova razlika u razdiobi realnih vektorskih procesa od pretpostavljene Gaussove razdiobe za sobom povlači degradaciju učinka skalarnih kvantizatora u transformacijskoj domeni.



Slika 4.4 *Primjer realizacije dvodimenzionalnog vektorskog procesa (a) i odgovarajućeg GM modela s tri komponente mješavine (b).*

Umjesto na pretpostavci o Gaussovoj razdiobi realnog vektorskog procesa, vektorski kvantizator moguće je projektirati na temelju modela njegove stvarne razdiobe. Stvarnu razdiobu realnog vektorskog procesa moguće je proizvoljno točno aproksimirati primjenom parametarskog modela s Gaussovom mješavinom, odnosno modeliranjem izvora mješavinom više individualnih i preklapajućih Gaussovih izvora. Vektorski prostor se na taj način dijeli na više relativno stacionarnih i preklapajućih regija, koje nazivamo klasterima (engl. *clusters*). Takvim se pristupom svaki vektor realiziranog vektorskog procesa može promatrati na način da je generiran jednim od preklapajućih Gaussovih izvora, ovisno o težini i *pdf* funkciji odgovarajućeg klastera. Na slici 4.4(a) prikazan je primjer realizacije dvodimenzionalnog vektorskog procesa, čija je funkcija gustoće vjerojatnosti aproksimirana modelom s Gaussovom mješavinom s tri komponente, prikazan na slici 4.4(b). Komponente mješavine i odgovarajući GM model

prikazane su linijama jednake vjerojatnosti (engl. *iso-probability contours*). Dekompozicijom *pdf* funkcije vektorskog procesa pomoću GMM-a, za svaki od preklapajućih Gaussovih izvora može se projektirati transformacijski koder koji optimalno kvantizira vektore odgovarajućeg klastera. Na taj bi način lokalna statistika svakog od klastera producirala jedinstvenu vlastitu matricu transformacije, što predstavlja osnovnu prednost GMM-temeljenog kodiranja u odnosu na konvencionalno transformacijsko kodiranje temeljeno na jedinstvenoj matrici transformacije.

Ovakav pristup odgovara adaptivnom transformacijskom kodiranju, koji dediciranjem transformacijskog kodera za svaki od klastera, optimizira učinkovitost skalarnih kvantizatora u transformacijskoj domeni. Sličnost distribucije pojedinog klastera s Gaussovom općenito se povećava s povećanjem ukupnog broja komponenti mješavine. Pod pretpostavkom da je svaki vektor generiran samo jednim od Gaussovih izvora dobivenih dekompozicijom, vrijedi zaključak da će transformacijski koder klastera s najvećom vjerojatnošću generiranja konkretnog vektora spomenuti vektor kodirati uz minimalno izobličenje. Međutim, preklapanje susjednih klastera omogućava korištenje tzv. mekog odabira odgovarajućeg (najboljeg) klastera (engl. *soft-clustering*) korištenjem kriterija najmanjeg izobličenja umjesto kriterija maksimalne vjerojatnosti. Iz opisanog je vidljivo da, primjenom GMM-a, računska složenost vektorskog kvantizatora ovisi o broju komponenti mješavine, umjesto o prijenosnoj brzini, kako je to slučaj kod klasičnih vektorskih kvantizatora.

S obzirom da GMM parametri aproksimirane *pdf* funkcije predstavljaju svojstvo vektorskog procesa, odnosno izvora, njihova estimacija neovisna je o prijenosnoj brzini kvantizatora. Isto vrijedi i za KLT transformacijske matrice, koje se izračunavaju iz estimiranih GMM parametara. Njihovim spremanjem na obje transmisijske strane (koder i dekoder) izbjegava se potreba za prijenosom dodatnih informacija (engl. *transmission overhead*), odnosno povećanjem brzine prijenosa.

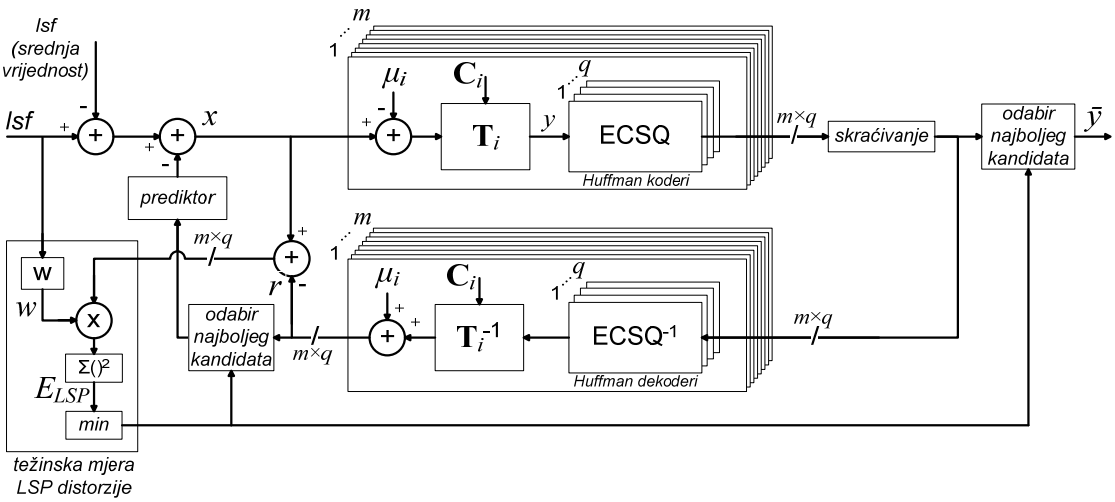
Ovisno o načinu skalarne kvantizacije, primjena GMM-a u transformacijskim koderima može rezultirati skalabilnošću vektorskog kvantizatora, odnosno mogućnošću promjene prijenosne brzine kvantizatora u bilo koju drugu brzinu tijekom procesa kvantizacije. To je slučaj kod blok kvantizatora, koji za promjenu brzine prijenosa trebaju iz eksplicitnih izraza izračunati novu raspodjelu raspoloživih bitova. U slučaju promjene brzine, izračun je moguće jednostavno i brzo napraviti na obje transmisijske strane, te nastaviti s kodiranjem koristeći novu brzinu prijenosa. To, međutim, nije slučaj kod transformacijskih kodera koji entropijskim kodom varijabilne duljine kodiraju izlazne indekse EC kvantizatora. Kod njih je za svaku od podržanih prijenosnih brzina potrebno unaprijed, tijekom dizajna kvantizatora, projektirati tablice entropijskih kodera.

5. Dizajn GMM-VQ za primjenu AMR koderu

Princip transformacijskog kodiranja, temeljen na modelu s Gaussovim mješavinama, u ovom je poglavlju primijenjen u svrhu dizajna konkretnog vektorskog kvantizatora spektralne ovojnice za primjenu u AMR koderu. Predloženi kvantizator parametarski modelira funkciju gustoće razdiobe rezidualnog signala MA predikcije, koju koristi i referentni AMR koder. Parametarska reprezentacija GMM modelom omogućava dekorelaciju reziduala predikcije primjenom KLT transformacije. Dekorelirani reziduali se onda kvantiziraju entropijski ograničenim skalarnim kvantizatorima, čiji se izlazni indeksi kodiraju primjenom entropijskih kodera. Varijabilnu izlaznu brzinu entropijskih kodera potrebno je prilagoditi fiksnoj prijenosnoj brzini AMR kodera, pa se u tu svrhu razmatra primjena dviju različitih tehnika prilagodbe.

5.1. Blok shema predloženog kvantizatora

Blok shema predloženog vektorskog kvantizatora LSF parametara temeljenog na GMM modelu prikazana je na slici 5.1. Kao i u standardnom AMR koderu, LSF vektorima oduzima se srednja vrijednost, te se nad razlikom radi MA predikcija u svrhu uklanjanja među-okvirnih korelacija. Kvantizacija LSF vektora se zapravo provodi kvantizacijom pogreške MA predikcije, kao što to radi i standardni AMR koder. Njihova *pdf* funkcija opisana je GMM modelom, čiji su parametri korišteni za projektiranje m transformacijskih kodera za dekorelaciju i kodiranje reziduala. S obzirom na međusobno preklapanje komponenti mješavine, nije unaprijed poznato koja će komponenta mješavine, odnosno transformacijski koder, rezidual kodirati uz najmanju pogrešku kvantizacije. Iz tog razloga, za odabir najbolje od m komponenti mješavine koristi se shema mekanog odabira. Svaki rezidual, x, prosljeđuje se u m transformacijskih kodera, koji odgovaraju pojedinim komponentama Gaussove mješavine. Svaki transformacijski koder zasebno dekorelira, kvantizira i entropijski kodira komponente reziduala. Nakon translacije u smjeru odgovarajućeg centroida (oduzimanjem odgovarajućeg vektora srednje vrijednosti, μ_i), slijedi dekorelacija reziduala množenjem s odgovarajućom transformacijskom matricom, $\mathbf{T}_i$. Komponente dekoreliranog reziduala, y, se onda kvantiziraju i kodiraju primjenom entropijski ograničenog skalarnog kvantizatora. Zbog prilagodbe varijabilne duljine entropijski kodiranih reziduala na fiksnu duljinu okvira za prijenos kodirane ovojnice spektra u AMR koderu (poglavlje 5.4), dekorelirani rezidual se zapravo kvantizira i kodira s q različitih ECSQ kvantizatora. Takav postupak rezultira sa $m \times q$ različitih kodiranih reziduala koji predstavljaju moguće kandidate za reprezentaciju ulaznog reziduala.



Slika 5.1 *Blok shema vektorskog kvantizatora LSF parametara temeljenog na modelu s Gaussovim mješavinama (m×q verzija).*

Kvantizirani i kodirani reziduali se onda inverznim postupkom prevode u originalnu domenu, te se uspoređuju s nekvantiziranim ulaznim rezidualom *x*. Usporedba se radi izračunom iste težinske mjere LSP distorzije (izraz 3.6), koja se u AMR koderu koristi za vektorsku kvantizaciju pojedine pod-matrice, odnosno pod-vektora. Konačno, odabire se komponenta mješavine čiji je transformacijski koder rezidual kodirao s najmanjom distorzijom. Dakle, uz poznat kvantizirani rezidual iz prethodnog okvira, svaka komponenta mješavine natječe se u produkciji kodiranog reziduala s najmanjom distorzijom u trenutnom okviru. Kako predloženi kvantizator, u svrhu odabira najboljeg kvantiziranog reziduala, u zatvorenoj petlji istovremeno procesira $m \times q$ kandidata, opisana verzija kvantizatora nazvana je $m \times q$ verzija. U poglavlju 5.7 opisana je računski jednostavnija alternativna verzija ovog kvantizatora.

5.2. Postupak projektiranja predloženog kvantizatora

Parametri predloženog vektorskog kvantizatora spektralne ovojnice izračunavaju se na temelju statističkih svojstava signala govornih sekvenci koje se upotrebljavaju u fazi učenja, odnosno treniranja kvantizatora (engl. *training phase*). Iz tog se razloga spomenute govorne sekvence nazivaju trening bazom za učenje (treniranje) kvantizatora. Poželjno je u trening bazu odabrati govorne sekvence širokog spektra govornika, kako bi svojstva projektiranog kodera bila što neovisnija o karakteristikama govornih sekvenci iz evaluacijske baze (engl. *evaluation database*), koje se u evaluacijskoj fazi (engl. *evaluation phase*) koriste za objektivno vrednovanje (testiranje) njegovih svojstava. Govorne sekvence iz evaluacijske baze nisu dio trening baze, te se ne upotrebljavaju u fazi učenja, odnosno dizajna vektorskog kvantizatora. Na taj način, svojstva projektiranog kvantizatora evaluiraju se na govornim sekvencama koje se prvi put procesiraju predloženim koderom.

Trening faza može se podijeliti u nekoliko sekvencijalnih koraka:

1. Primjenom GMM modela parametarski se modelira *pdf* funkcija vektorâ pogreške MA predikcije. Vektori pogreške MA predikcije izdvajaju se, skupa s pripadajućim LSF vektorima, iz standardnog AMR kodera tijekom procesiranja govornih sekvenci iz trening baze. Estimirani parametri *m* komponenti Gaussove mješavine, se onda u sljedećim koracima koriste za projektiranje *m* transformacijskih kodera.
2. U skladu s postupkom opisanim u poglavlju 4.3, iz estimiranih matrica kovarijanci se za svaku od komponenti mješavine izračunava KLT transformacijska matrica.
3. Svaki ulazni rezidual transformira se parametrima svake komponente mješavine. Nakon translacije u obrnutom smjeru odgovarajućeg centroida (oduzimanjem vektora srednje vrijednosti), slijedi rotacija primjenom odgovarajuće transformacijske matrice. Na taj način, svaki od ulaznih vektora imat će po *m* reprezentanata u transformacijskoj domeni.
4. Primjenom jednog od principa diskutiranih u sljedećem poglavlju, svakom od ulaznih vektora pridružuje se jedna komponenta mješavine, odnosno jedan od *m* reprezentanata u transformacijskoj domeni. Ovakav način inicijalnog pridruživanja (asocijacije) odgovara odabiru u otvorenoj petlji (engl. *open loop selection*). Rezultat inicijalnog pridruživanja je baza transformiranih reziduala na kojoj se temelji dizajn entropijski ograničenih skalarnih kvantizatora za kodiranje komponenti transformiranih reziduala.
5. U skladu s razmatranjima u poglavlju 5.4, slijedi određivanje *q* koraka kvantizacije, koji će producirati kvantizirane komponente transformiranih reziduala, čija je entropija jednaka unaprijed određenim *q* različitih entropijskih vrijednosti.
6. Izračunati koraci kvantizacije se onda dalje koriste za projektiranje *q* Huffmanovih kodera za svaku komponentu transformiranog procesa.

Provedbom opisanih koraka bi se dizajn LSF VQ-a temeljenog na GMM-u mogao smatrati gotovim. Ulazni parametri postupka projektiranja, dakle, podrazumijevaju:

- trening bazu izdvojenih rezidualnih vektora,
- informaciju u broju komponenti modela Gaussove mješavine (*m*) kojom će se modelirati funkcija gustoće razdiobe izdvojenih rezidualnih vektora,
- informaciju o broju ECSQ kvantizatora (*q*) za prilagodbu na fiksnu strukturu kodiranog okvira AMR kodera (poglavlje 5.4), te vrijednosti njihovih ukupnih entropija.

Kao rezultat gore opisanog postupka proizlazi *m* transformacijskih kodera koje u potpunosti opisuju sljedeći parametri:

- vektori srednjih vrijednosti svih komponenti mješavine, μ_i, i=1,...,m,

- transformacijske matrice, $\mathbf{T}_i$, svih komponenti mješavine, producirane iz odgovarajućih matrica kovarijanci, $\mathbf{C}_i$,
- koraci kvantizacije, s_k, k=1,...,q, za kvantizaciju komponenti transformiranih reziduala,
- $d \times q$ Huffmanovih kodera za entropijsko kodiranje kvantiziranih komponenti transformiranih reziduala.

Opisanim postupkom su parametri projektiranog kvantizatora zapravo optimizirani za kodiranje reziduala koje producira odgovarajući kvantizator spektralne ovojnice referentnog AMR kodera, jer postupak modelira funkciju gustoće vjerojatnosti izdvojenih nekvantiziranih reziduala. Međutim, kako zbog prediktivne prirode MA strukture izračun reziduala podrazumijeva poznavanje kvantiziranih reziduala LSF vektora prethodnih okvira, reziduali istih LSF vektora, izračunati projektiranim kvantizatorom, razlikovat će se od izdvojenih reziduala. Dakle, razlika u kvantizacijskoj metodi uzrokuje razliku u statističkoj razdiobi reziduala, što model Gaussove mješavine, a samim tim i projektirani kvantizator čini suboptimalnim.

Nadalje, za uspješnu rekonstrukciju kodiranog govornog signala, predloženi kvantizator mora dekoderu prenijeti informaciju o odabranoj kombinaciji komponente mješavine i ECSQ kvantizatora koja najbolje kodira ulazni rezidual. Za kodiranje informacije o odabranoj kombinaciji moguće je koristiti običan binarni kôd duljine $log_2 m + log_2 q$ bitova. Naravno, kodiranje odabrane kombinacije smanjit će broj bitova dostupnih za kodiranje komponenti transformiranog reziduala za duljinu binarnog kôda odabrane kombinacije. Međutim, u skladu s argumentacijom u poglavlju 5.4.4, odabranu kombinaciju bilo bi poželjno također kodirati entropijskim kôdom. U tom slučaju, postupak projektiranja mora uključivati i dizajn Huffmanovog kodera za kodiranje odabrane kombinacije komponente mješavine i ECSQ kvantizatora. Dizajn takvog kodera kao ulaznu informaciju zahtijeva *pdf* funkciju odabranih kombinacija, za čiju je dostupnost potrebno provesti postupak kodiranja izdvojenih LSF vektora.

Primjenom projektiranog vektorskog kvantizatora za kodiranje LSF vektora izdvojenih iz trening baze, zapravo će se kodirati nova baza reziduala, koja se dobiva MA predikcijom na temelju kvantiziranih reziduala iz prethodnih okvira. Kako se reziduali kvantizirani projektiranim kvantizatorom razlikuju od reziduala kvantiziranih referentnim AMR koderom, statistička distribucija novih reziduala razlikuje se od distribucije reziduala izdvojenih iz AMR kodera. Proces kvantizacije i kodiranja novih reziduala će, evaluacijom distorzije u zatvorenoj petlji, napraviti novo združivanje izdvojenih LSF vektora s odgovarajućim komponentama mješavine i najboljim ECSQ kvantizatorima. Ponavljanjem postupka projektiranja, uz uvjet da se, umjesto *pdf* funkcije izdvojenih reziduala, Gaussovom mješavinom modelira *pdf* funkcija novih reziduala iz prve iteracije, očekuje se bolji model Gaussove mješavine, a samim tim i bolji parametri kvantizatora koji se iz njega izračunavaju. Proces kodiranja kvantizatorom iz prve iteracije postupka projektiranja će, evaluacijom težinske mjere LSP distorzije u zatvorenoj petlji, ponoviti združivanje LSF vektora s odgovarajućim komponentama mješavine i najboljim ECSQ kvantizatorima. Rezultat ponovnog

združivanja je nova baza transformiranih reziduala za novo projektiranje ECSQ kvantizatora. Nadalje, informacija o odabranim kombinacijama se u drugoj iteraciji postupka projektiranja može upotrijebiti za dizajn Huffmanovog kodera u svrhu entropijskog kodiranja indeksa odabrane kombinacije.

Do sada opisani postupak projektiranja u dvije iteracije svodi se na sljedeće:

- prva iteracija, opisana točkama 1-6, temelji se na izdvojenim rezidualima i inicijalnom pridruživanju komponenti mješavine LSF vektorima u otvorenoj petlji, te kao takva, iz gore opisanih razloga, rezultira suboptimalnim parametrima projektiranog kvantizatora;
- druga iteracija temelji se na rezidualima i reasociranim indeksima najboljih kombinacija komponente mješavine i ECSQ kvantizatora, koje u zatvorenoj petlji izračunava suboptimalni kvantizator projektiran u prvoj iteraciji. Rezultirajući parametri novog kvantizatora se, u svakom slučaju, smatraju boljim od parametara iz prve iteracije, jer je faza učenja provedena na relevantnijim ulaznim parametrima.

Postupak projektiranja mogao bi se nastaviti kroz sljedeće iteracije, gdje bi svaka sljedeća iteracija kao ulazne parametre koristila različite reziduale istih LSF vektora i indekse kombinacija reasocirane kvantizatorom iz prethodne iteracije. Naime, u svakoj iteraciji bi promijenjeni parametri kvantizatora rezultirali drugačijim rezidualima i reasocijacijom s indeksima najboljih kombinacija, koji bi onda u sljedećoj iteraciji opet promijenili parametre projektiranog kvantizatora. Učinkovitost takvog iterativnog pristupa dizajnu LSF VQ temeljenog na modelu Gaussovih mješavina diskutiran je uz simulacijske rezultate u poglavlju 6.6.

Sumirajući sve navedeno, postupak projektiranja opisan koracima 1-6 može se nadopuniti sljedećim koracima:

7. Izdvojeni LSF vektori kodiraju se kvantizatorom iz prethodne iteracije postupka projektiranja. Tijekom kodiranja izračunavaju se novi reziduali. Evaluacijom težinske LSP distorzije u zatvorenoj petlji, indeksi komponente mješavine i ECSQ kvantizatora, koji ulazni vektor kodiraju s najmanjom distorzijom, pridružuju se odgovarajućim LSF vektorima.
8. Počinje druga iteracija postupka projektiranja. GMM modelom modelira se *pdf* funkcija novih reziduala iz koraka 7.
9. Iz novih matrica kovarijanci novog GMM modela izračunavaju se nove KLT transformacijske matrice za svaku od komponenti mješavine.
10. Koristeći pridruživanje iz koraka 7, transformacijom novih reziduala s odgovarajućim transformacijskim matricama iz prethodnog koraka, dobiva se nova baza transformiranih reziduala na kojoj se temelji projektiranje novih ECSQ kvantizatora.

11. Slijedi određivanje q novih koraka kvantizacije, koji će producirati kvantizirane komponente transformiranih reziduala čija je ukupna entropija jednaka unaprijed određenim q različitih entropijskih vrijednosti.
12. Izračunati koraci kvantizacije se, skupa s novom bazom transformiranih vektora iz koraka 10, dalje koriste za projektiranje q Huffmanovih kodera za svaku komponentu transformiranog procesa.
13. Na temelju kombinacija komponenti mješavine i ECSQ kvantizatora, pridruženih izdvojenim LSF vektorima u koraku 7, projektira se Huffmanov koder za entropijsko kodiranje indeksa odabrane kombinacije.
14. Ovisno o unaprijed određenom broju iteracija postupka projektiranja, koraci 7-13 ponavljaju se u svakoj sljedećoj iteraciji.

5.3. Inicijalno pridruživanje komponenti mješavine ulaznim LSF vektorima

Opisani postupak projektiranja, u točkama 5 i 6 prve iteracije, za izračun kvantizacijskih koraka i dizajn odgovarajućih ECSQ kvantizatora podrazumijeva postojanje baze transformiranih reziduala. Transformacija reziduala sastavni je dio procesa kodiranja LSF vektora predloženim kvantizatorom. S obzirom na međusobno preklapanje, svaka komponenta mješavine natječe se u produkciji kodiranog reziduala s najmanjom distorzijom, te se u tom smislu najbolja komponenta pridružuje odgovarajućem LSF vektoru. Kako u prvoj iteraciji proces kodiranja izdvojenih LSF reziduala još uvijek nije proveden, ne postoji asocijacija pojedinih reziduala s komponentom mješavine, odnosno transformacijskim koderom koji će rezidual kodirati uz najmanju pogrešku kvantizacije. Stoga je, za izračun baze transformiranih reziduala, potrebno provesti inicijalno pridruživanje odgovarajuće komponente mješavine svakom izdvojenom rezidualu. Za razliku od pridruživanja koje se tijekom kodiranja provodi evaluacijom težinske LSP distorzije u zatvorenoj petlji, inicijalno pridruživanje potrebno je provesti u otvorenoj petlji, tj. bez usporedbe inverzno transformiranog kodiranog reziduala s ulaznim rezidualom u originalnoj domeni. To je razumljivo, jer su za kodiranje reziduala potrebni ECSQ kvantizatori za čije je projektiranje neophodna baza transformiranih vektora. Inicijalno pridruživanje u otvorenoj petlji može se provesti primjenom jednog od sljedećih principa:

- ***princip najveće vjerojatnosti*** – uz poznate parametre Gaussove mješavine kojom se modelira *pdf* funkcija izdvojenih reziduala, za svaki rezidual evaluira se vjerojatnost pripadanja svakoj od m komponenti mješavine, uzimajući u obzir i težinu i-te komponente mješavine, ρ_i. Transformacijska matrica komponente (m_{sel}) s najvećom vjerojatnošću:

$$m_{sel} = \arg\max_i \left(\rho_i \mathcal{N}(x; \mu_i; \mathbf{C}_i)\right), \qquad (5.1)$$

  se onda primjenjuje za transformaciju danog reziduala u prvoj iteraciji postupka projektiranja predloženog kvantizatora.

- ***princip najmanje energije u transformacijskoj domeni*** – za svaku komponentu mješavine, svaki izdvojeni rezidual translatira se u obrnutom smjeru odgovarajućeg centroida, te rotira primjenom odgovarajuće KLT transformacijske matrice. Drugim riječima, svaki ulazni rezidual transformira se parametrima svake komponente mješavine. Komponenta koja za dani ulazni rezidual producira transformirani vektor najmanje energije (duljine), smatra se najučinkovitijom:

$$m_{sel} = \arg\min_{i}\left(\sum_{j=1}^{d} y_{i,j}^{2}\right), \qquad (5.2)$$

  te se onda primjenjuje za transformaciju danog reziduala u prvoj iteraciji postupka projektiranja predloženog kvantizatora.

- ***princip najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni*** – ovaj princip također podrazumijeva KLT transformaciju svakog izdvojenog reziduala parametrima svake komponente mješavine. Komponenta mješavine koja rezultira s najmanjim umnoškom apsolutnih vrijednosti komponenti transformiranog vektora smatra se najučinkovitijom:

$$m_{sel} = \arg\min_{i}\left(\prod_{j=1}^{d} \left|y_{i,j}\right|\right), \qquad (5.3)$$

  te se onda primjenjuje za transformaciju danog reziduala u prvoj iteraciji postupka projektiranja predloženog kvantizatora. Ovaj princip je zapravo konzistentan s konceptom maksimiziranja dobitka transformacijskog kodiranja (engl. *Transform Coding Gain,* TCG) za pojedini ulazni vektor, kako je i pojašnjeno u sljedećem podpoglavlju.

5.3.1. Princip najvećeg dobitka transformacijskog kodiranja

Dobitak transformacijskog kodiranja (TCG) [18] predstavlja uobičajenu mjeru učinkovitosti u transformacijskom kodiranju. Za svaku od m komponenti Gaussove mješavine, definiran je odnosom aritmetičke i geometrijske sredine varijanci komponenata transformiranih vektora:

$$G_i = \frac{\frac{1}{d}\sum_{j=1}^{d}\lambda_{i,j}}{\left(\prod_{j=1}^{d}\lambda_{i,j}\right)^{1/d}} = \frac{aritmet.\ sredina}{geomet.\ sredina} = \frac{AS}{GS}, \qquad (5.4)$$

gdje d predstavlja dimenzionalnost vektorâ koji se transformiraju, $\lambda_{i,j}$ predstavlja varijancu j-te komponente vektora transformiranih parametrima i-te komponente Gaussove mješavine, i=1,...,m i j=1,...,d. Ovako izražena mjera učinkovitosti ne može se koristiti za inicijalno pridruživanje najbolje komponente Gaussove mješavine pojedinom

rezidualu, jer je definirana u smislu statističkog očekivanja koje podrazumijeva poznavanje varijanci komponenata transformiranih vektora. Međutim, kako varijance predstavljaju očekivanu vrijednost kvadrata apsolutnih vrijednosti pojedinih komponenti transformiranih vektora, jednostavna zamjena varijanci sa kvadratima apsolutnih vrijednosti komponenti pojedinog transformiranog vektora u izrazu 5.4 zadržat će konzistentnost s konceptom maksimiziranja TCG-a.

Dobitak G_i u izrazu 5.4 ima maksimalnu vrijednost pri minimalnoj vrijednosti geometrijske sredine u nazivniku, jer je aritmetička sredina u brojniku nepromjenjiva za bilo koju ortonormalnu transformaciju kao što je KLT. Dakle, kod inicijalnog pridruživanja primjenom ovog principa odabire se komponenta Gaussove mješavine koja KLT transformacijom minimizira geometrijsku sredinu kvadrata apsolutnih vrijednosti komponenti transformiranog vektora, što je ekvivalentno odabiru komponente mješavine koja minimizira umnožak njihovih apsolutnih vrijednosti:

$$m_{sel} = \arg\min_i \left(\left(\prod_{j=1}^{d} y_{i,j}^2 \right)^{1/d} \right) = \arg\min_i \left(\prod_{j=1}^{d} \left| y_{i,j} \right| \right), \qquad (5.5)$$

Ovakav kriterij spada u skupinu kriterija za odabir u otvorenoj petlji, jer u procesu pridruživanja ne evaluira učinak odabira na svojstva kvantizatora.

5.4. Prilagodba na fiksnu prijenosnu brzinu AMR kodera

Iz razloga opisanih u poglavlju 4.4, u svrhu kvantizacije i kodiranja komponenti transformiranog vektorskog procesa odabrano je entropijsko kodiranje izlaznih indeksa uniformnih skalarnih kvantizatora. Drugim riječima, radi se o skalarnoj kvantizaciji s ograničenom entropijom. Naravno, uz prednosti entropijskog kodiranja, takav izbor unosi probleme vezane uz varijabilnu prijenosnu brzinu i propagaciju pogreške, te probleme vezane s punjenjem izlaznog spremnika.

Varijabilna prijenosna brzina zapravo znači da izlazna prijenosna brzina ovisi o trenutnim vrijednostima komponenti dekoreliranih vektora pogreške predikcije. Kako AMR koder, ovisno o môdu rada, za kodiranje spektralne ovojnice koristi unaprijed određen fiksan broj bitova po okviru (tablica 3.2), varijabilnu prijenosnu brzinu entropijskih kodera potrebno je prilagoditi strukturi AMR kodera s fiksnom prijenosnom brzinom. Dakle, entropijskim koderima na raspolaganju stoji isti broj bita koji je u AMR koderu dostupan SMQ i SVQ metodama u odgovarajućim modovima rada. U fazi projektiranja entropijskih kodera ograničava se samo njihova prosječna izlazna prijenosna brzina na način da se statistički vjerojatnijim simbolima dodjeljuju kraći kodovi i obrnuto. Posljedično, reziduali čije dekorelirane komponente imaju relativno visoku pojavnost kodiraju se kraćim kodovima, koji su u zbroju kraći od broja dostupnih bita. Takva situacija ne rezultira optimalnim iskorištenjem dostupne prijenosne brzine. S druge strane, reziduali sa simbolima male vjerojatnosti koristit će duge kodove koji u zbroju prekoračuju dostupnu fiksnu duljinu.

Očito je da su za primjenu entropijskog kodiranja pri kodiranju spektralne ovojnice u AMR koderu nužne određene modifikacije. Rezultat modifikacija treba osigurati sustav fiksne (stalne) prijenosne brzine na razini okvira, zadržavajući varijabilnu prirodu na razini pojedine komponente reziduala. U tu svrhu korištena je kombinacija tehnika varijacije koraka kvantizacije i skraćivanja transformiranog vektora. S ciljem optimiziranja navedenih metoda, istražen je njihov utjecaj na kvantizaciju LSF parametara kao i svojstva modificiranog AMR kodera u cjelini.

5.4.1. Varijacija koraka kvantizacije

Zbog činjenice da KLT koncentrira varijancu, odnosno energiju izvora u prvih nekoliko komponenti transformiranog vektora, svaka komponenta imat će drugačiji raspon vrijednosti i statistička svojstva. Iz tog se razloga za svaku od *d* komponenti projektira poseban Huffmanov entropijski koder. Kako svih *d* Huffmanovih kodera koriste isti korak kvantizacije, u nastavku rada se tretiraju kao jedan ECSQ kvantizator čija je ukupna entropija jednaka zbroju pojedinačnih entropija svakog kodera. Ukupna entropija ECSQ kvantizatora je na taj način povezana sa maksimalnom dužinom koda dostupnog za kodiranje ovojnice spektra jednog vremenskog okvira.

U slučaju da su Huffmanovi koderi projektirani s ciljem da ukupna entropija ECSQ kvantizatora, odnosno prosječna dužina kodiranih vektora bude jednaka najvećoj dostupnoj dužini kôda, veliki dio kodiranih vektora će dužinom prekoračiti dostupni broja bitova za kodiranje ovojnice (engl. *buffer overflow*). Kako veličina kvantizacijskog koraka direktno utječe na entropiju izlaznih indeksa skalarnih kvantizatora, problem prekoračenja moguće je rješavati povećanjem koraka kvantizacije, odnosno smanjivanjem ukupne entropije ECSQ kvantizatora. Takvo bi rješenje, međutim, vodilo ka suboptimalnom iskorištenju dostupne prijenosne brzine, jer bi se proporcionalno smanjila i dužina svih kodiranih vektora koji u prethodnoj varijanti nisu prelazili raspoloživu duljinu (engl. *buffer underflow*).

Očito je da bi sustav sa više ECSQ kvantizatora bio fleksibilniji i učinkovitiji u smislu optimalnog iskorištenja dostupne prijenosne brzine, a samim time i u smislu učinkovitijeg kodiranja. Takav bi sustav koristio *q* ECSQ kvantizatora koji bi koristili različite, predefinirane korake kvantizacije. Kao rezultat, sustav bi adaptivno koristio odgovarajući ECSQ kvantizator, čija je ukupna entropija odabrana u blizini najveće dostupne duljine kôda. Općenito, „jednostavni“ transformirani vektori će nakon kvantizacije sadržavati simbole visoke vjerojatnosti, te će im Huffmanovi koderi dodijeliti kraće kodove. Takve vektore moguće je kodirati korištenjem ECSQ kvantizatora s najvećom ukupnom entropijom, odnosno prijenosnom brzinom, što će osigurati najbolju kvantizaciju s obzirom na korištenje najfinijeg (najmanjeg) kvantizacijskog koraka. Nasuprot tome, „kompliciran“ transformirani vektori će nakon kvantizacije sadržavati simbole manje vjerojatnosti koji za kodiranje Huffmanovim koderima zahtijevaju duže kodove. U tom slučaju, kodirani vektori neće stati u dostupni okvir kodiranjem ECSQ kvantizatorom s najvećom ukupnom entropijom, pa će se provesti kvantizacija ostalim ECSQ kvantizatorima s većim koracima kvantizacije,

odnosno manjim ukupnim entropijama, u nastojanju da duljina kvantiziranog vektora ne prekorači najveću dostupnu duljinu.

Očito je da učinkovitost kvantizacije opisanom adaptivnom metodom značajno ovisi o izboru, odnosno kombinaciji ukupnih entropija ESCQ kvantizatorâ. Nadalje, treba spomenuti da je opisano poboljšanje u smislu optimalnosti plaćeno dodatnim bitovima potrebnim za kodiranje informacije (engl. *side information*) o odabranom ECSQ kvantizatoru, koja je dekoderu nužna za rekonstrukciju. Kodiranje te informacije moguće je pojednostavniti odabirom fiksnog broja unaprijed definiranih koraka kvantizacije. S tom svrhom eksperimentirano je s nekoliko različitih strategija za odabir kombinacije ukupnih entropija. Rezultati eksperimenata opisani su u poglavlju sa simulacijskim rezultatima.

Tijekom kvantizacijskog procesa transformirani vektor poželjno je kodirati upotrebom ECSQ kvantizatora s najmanjim (najfinijim) korakom kvantizacije, pod uvjetom da je entropijski kôd kodiranog vektora ograničen najvećom dostupnom duljinom kôda. Kako nije unaprijed moguće znati koji od ECSQ kvantizatora će proizvesti entropijski kôd ograničene duljine, transformirani vektor se kodira redom, svim projektiranim ECSQ kvantizatorima, počevši od kvantizatora s najvećom ukupnom entropijom. Prvi od njih čiji je entropijski kôd manji ili jednak raspoloživoj duljini kôda može biti odabran za kodiranje transformiranog vektora. Međutim, moguća je i situacija u kojoj čak ni kvantizator s najvećim korakom, odnosno najmanjom ukupnom entropijom ne može proizvesti entropijski kôd ograničene duljine. Takve „kompliciranе“ transformirane vektore moguće je kodirati primjenom druge tehnike za ograničavanje duljine kôda, opisane u sljedećem podpoglavlju.

5.4.2. Skraćivanje transformiranog vektora

Ova tehnika za ograničavanje duljine kôda se zapravo svodi na obično skraćivanje duljine transformiranog vektora, tj. odbacivanje komponenti transformiranog vektora kako bi na razini vektora dobili entropijski kôd ograničen dostupnom duljinom. Naime, kako značaj komponenti vektorâ transformiranih KLT transformacijom opada u skladu s rednim brojem komponente, odnosno njenom varijancom, jednostavnim odbacivanjem najmanje značajnih komponenti na kraju transformiranog vektora moguće je reducirati njegovu dimenzionalnost bez značajnijeg utjecaja na cjelokupna svojstva kvantizatora. Tako će zadnji bitovi entropijski kodiranog vektora, koji je skraćen ovom tehnikom, zapravo sadržavati frakciju entropijskog koda prve od odbačenih komponenti, tj. njegov dio koji je stao u kodirani bitovni niz, kako njegova ukupna duljina ne bi prekoračila raspoloživu duljinu na razini vektora.

Za razliku od tehnike varijacije koraka kvantizacije, ova tehnika ne zahtijeva prijenos dodatnih informacija prema dekoderu u svrhu rekonstrukcije. Tijekom dekodiranja, dekoder jednostavno prati trenutnu poziciju u bitovnom nizu koji predstavlja entropijski kôd transformiranog vektora. Ukoliko je maksimalna duljina niza prekoračena, tj. ako je dekoder došao do kraja poruke prije nego je došao do lista, odnosno dok se još nalazi na nekom od međučvorova Huffmanovog stabla (tablice), za komponentu vektora koja se trenutno dekodira odaslani simbol je nepoznat (nepotpun). Odaslani simbol može biti

bilo koji od listova koji izlaze iz konkretnog međučvora. U tom slučaju, dekodirana vrijednost trenutne i svih sljedećih komponenti trenutnog vektora zamjenjuju se nulom. Zapravo, nakon inverzne KLT transformacije, vrijednost tih komponenti zamijenit će se odgovarajućim komponentama vektora srednje vrijednosti odabrane komponente mješavine.

5.4.3. Kvantizacija transformiranih vektorskih komponenti

U svrhu kvantizacije komponenti transformiranih vektora predloženi VQ spektralne ovojnice temeljen na GMM-u kombinira i koristi obje opisane metode za ograničavanje duljine entropijskog kôda, na taj način kompenzirajući nedostatke jedne ili druge metode. Transformirani vektor kodira se primjenom svih q unaprijed projektiranih ECSQ kvantizatora. Odgovarajući entropijski kodovi se po potrebi skraćuju opisanom tehnikom, te se na taj način dobiva q entropijski kodiranih kandidata koji zadovoljavaju uvjet duljine. Primjenom principa odgođene odluke (engl. *delayed decision*), nakon rekonstrukcije LSF vektorâ i evaluacije distorzije u originalnoj domeni za sve kandidate, odabire se ECSQ kvantizator čiji kandidat unosi najmanju distorziju u postupak kvantizacije.

Očito je da se, primjenom tehnika varijacije koraka kvantizacije i skraćivanja transformiranog vektora, bilo koji transformirani vektor, uključujući i one „jednostavne“ i „komplicirane“, može kodirati primjenom bilo kojeg ECSQ kvantizatora, neovisno o njegovom koraku kvantizacije, odnosno ukupnoj izlaznoj entropiji. Naravno, ovisno o odabiru kvantizatora ovisit će i distorzija kvantiziranog vektora. U svakom slučaju, „komplicirani“ vektori, kodirani kvantizatorima s većom ukupnom entropijom, bit će više skraćeni u usporedbi s kodiranim kandidatima kvantizatora s manjom ukupnom entropijom.

Kako se princip odgođene odluke također koristi i za odabir najbolje komponente mješavine za kodiranje konkretnog ulaznog vektora, za svaki ulazni vektor producira se m kandidata koji se onda entropijski kodiraju korištenjem opisanih tehnika. U konačnici, takav pristup daje $m \times q$ kodiranih kandidata među kojima se onda evaluacijom distorzije u originalnoj domeni bira najbolji.

5.4.4. Kvantizacija indeksa komponente mješavine i odabranog kvantizacijskog koraka

Za rekonstrukciju ovojnice spektra kodiranog govornog signala, dekoderu je, osim kodiranih komponenti transformiranog vektora, potrebno prenijeti i informaciju o odabranoj kombinaciji komponente mješavine i odgovarajućeg ECSQ kvantizatora. U tu svrhu moguće je koristiti jednostavne binarne kodove duljine $\log_2 m$ bitova za kodiranje odabrane komponente mješavine, te $\log_2 q$ bitova za kodiranje odabranog ECSQ kvantizatora. Takav pristup bi za $\log_2 m + \log_2 q$ bitova umanjio maksimalnu duljinu entropijski kodiranog transformiranog vektora. Kako statistika pojedinačnih indeksa nije uniformna, prirodno je kodirati ih, također, entropijskim koderom koji zajednički kodira oba indeksa. Ovakav pristup u prosjeku povećava broj bita dostupan za kodiranje komponenti transformiranog vektora, jer je entropija kodera odabrane kombinacije u

pravilu manja od $m+q$ za neuniformnu razdiobu korištenih kombinacija. Pri odabiru ukupnih entropija ECSQ kvantizatora, dakle, treba uzeti u obzir i broj bitova potrebnih za kodiranje indeksa odabrane kombinacije komponente mješavine i odgovarajućeg ECSQ kvantizatora.

Kao zaključak, predloženi sustav potpuno je opisan sa q ECSQ kvantizatora po svakoj komponenti transformiranog procesa, što u konačnici znači $d \times q$ odgovarajućih Huffmanovih tablica za entropijsko kodiranje transformiranog vektora, te Huffmanovom tablicom za entropijsko kodiranje odabrane kombinacije.

5.5. Odabir najbolje komponente mješavine

Predloženi kvantizator odabir najbolje komponente Gaussove mješavine i ECSQ kvantizatora radi u zatvorenoj petlji. Za odabir kombinacije koja najbolje kodira ulazni LSF rezidual x, kvantizirani kandidati se inverznim procesiranjem prevode u originalnu domenu i uspoređuju s ulaznim rezidualnim vektorom koji se kvantizira. Rekonstrukcija svakog kvantiziranog reziduala jednostavno se provodi inverznom KLT transformacijom transformiranog vektora čije se komponente izračunavaju jednostavnim množenjem kvantizacijskih indeksa s odgovarajućim korakom kvantizacije. Dodavanjem vektora srednje vrijednosti odgovarajuće komponente Gaussove mješavine dobiva se rekonstruirani kvantizirani LSF rezidual r^i koji aproksimira originalni rezidual x. Od ukupno $m \times q$ rekonstruiranih reziduala, odabire se kandidat s indeksom i koji minimizira težinsku mjeru LSP distorzije

$$m_{sel} = \arg\min_i E_{LSP}(r^i), \tag{5.6}$$

gdje je

$$E_{LSP}(r^i) = \sum_{j=1}^{10} \left[x_j w_j - r_j^i w_j \right]^2 . \tag{5.7}$$

Izraz 5.7 predstavlja istu mjeru distorzije koju referentni AMR koder koristi pri odabiru vektora iz kodne knjige odgovarajućeg vektorskog kvantizatora (izraz 3.6). Težinski koeficijenti w_j izračunavaju se na temelju odgovarajućih nekvantiziranih LSF vektora prema izrazu 3.7, jednako kao i u referentnom AMR koderu. Za môd 12k2, iz nekvantiziranih se LSF reziduala u svakom okviru izračunavaju dva skupa težinskih koeficijenata, za evaluaciju distorzije dva LSF reziduala.

Inverzno procesiranje kvantiziranih kandidata nužno je iz razloga što se ortogonalne baze transformiranih prostora različitih komponenti mješavine međusobno razlikuju. U tom se slučaju, usporedba izračunom udaljenosti, kao u izrazu 5.7, mora napraviti u originalnoj domeni.

5.6. Objektivno vrednovanje učinkovitosti

Za objektivno vrednovanje učinkovitosti predloženog algoritma za kodiranje spektralne ovojnice u AMR koderu korištene su dvije metode. Učinkovitost kodiranja spektralne ovojnice samog LSF VQ temeljenog na GMM modelu evaluira se izračunom spektralne

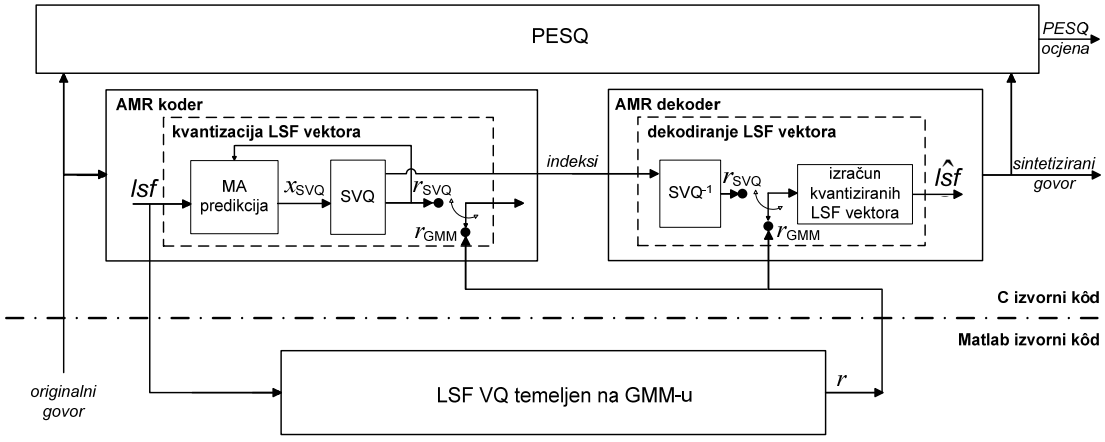
distorzije, dok se cjelokupna svojstva modificiranog AMR kodera evaluiraju primjenom algoritma perceptualne procjene kvalitete govora (engl. *Perceptual Evaluation of Speech Quality,* PESQ). Svojstva modificiranog sustava se onda uspoređuju sa svojstvima polaznog sustava, dobivenim primjenom obje evaluacijske metode na kvantizacijski algoritam referentnog AMR kodera.

5.6.1. Mjera spektralne distorzije

SD mjera predstavlja koristan kriterij za objektivno vrednovanje učinkovitosti kvantizacije spektralne ovojnice i uobičajeno se koristi za usporedbu različitih kvantizacijskih tehnika LPC modela [20]. Ovom metodom izračunava se kvadratna srednja vrijednost (engl. *Root Mean Square,* RMS) udaljenosti logaritamski izraženih amplitudnih spektara originalnog i kvantiziranog LPC modela za svaki vremenski okvir govornog signala. Usrednjavanjem za sve okvire svih govornih sekvenci iz trening ili evaluacijske baze dobiva se prosječna SD vrijednost kvantizatora u odgovarajućem môdu. Uz spektralnu distorziju izračunava se postotak p2 i p4 pogreški, koje odgovaraju postotku okvira sa spektralnom distorzijom većom od 2dB, odnosno 4 dB.

5.6.2. Perceptualna procjena kvalitete govora

SD mjera, zbog orijentiranosti na spektralnu ovojnicu, nije nužno povezana sa učinkovitošću cjelokupnog kodera, pogotovu kod kodera sa zatvorenom petljom kao što je to AMR koder. Iz tog razloga, za objektivno vrednovanje učinkovitosti AMR kodera s modificiranim LSF kvantizatorom koristi se PESQ algoritam, koji predstavlja objektivni mjerni algoritam za evaluaciju kvalitete transmisije. PESQ algoritam definiran je ITU-T preporukom P.862.



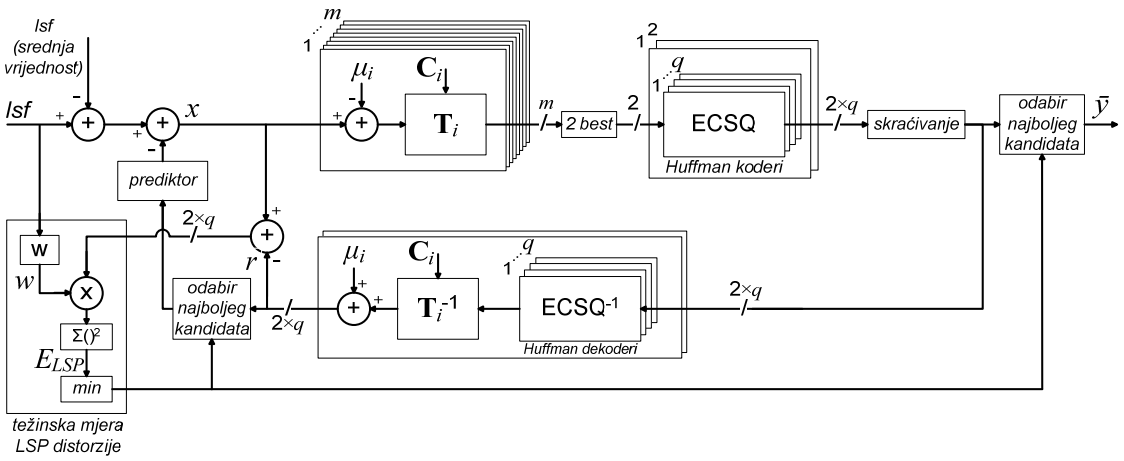
Slika 5.2 *Blok shema simulacijskog modela za objektivno vrednovanje učinkovitosti PESQ algoritmom.*

Slika 5.2 prikazuje blok shemu simulacijskog modela za objektivno vrednovanje učinkovitosti PESQ algoritmom. Kvantizirani se LSF reziduali u referentnom AMR koderu i dekoderu (r_{SVQ}) zamjenjuju rezidualima koji su kvantizirani LSF VQ-om temeljenim na modelu Gaussovih mješavina (r_{GMM}), koji se onda dalje koriste u procesu

kodiranja i dekodiranja govorne sekvence. Dekodirane i degradirane govorne sekvence se primjenom PESQ algoritma uspoređuju s odgovarajućim originalnim (neprocesiranim) govornim sekvencama. Rezultat usporedbe za svaku procesiranu govornu sekvencu je PESQ ocjena kvalitete, koja je analogna subjektivnoj ocjeni dobivenoj MOS (engl. *Mean Opinion Score*) testom prema ITU-T P.800.

5.7. Jednostavnija varijanta predloženog kodera

Kodiranje ulaznog reziduala transformacijskim koderima svih komponenti Gaussove mješavine, te njihovo inverzno procesiranje u svrhu odabira najbolje od njih, značajno doprinosi računskoj kompleksnosti predloženog LSF kvantizatora. S ciljem reduciranja računske kompleksnosti, projektiran je alternativni sustav, koji u zatvorenoj petlji procesira manji broj reziduala u usporedbi sa $m \times q$ verzijom kvantizatora. Blok dijagram alternativne *2-best* verzije LSF kvantizatora prikazan je na slici 5.3.



Slika 5.3 *Blok shema računski jednostavnije verzije LSF VQ temeljenog na modelu Gaussovih mješavina (2-best verzija).*

Nakon transformacije ulaznog reziduala primjenom transformacijskih matrica svih komponenti mješavine, prikazani kvantizator od m dostupnih reziduala u transformacijskoj domeni odabire 2 najbolja (engl. *2 best*). Odabir se radi primjenom TCG kriterija, opisanog u poglavlju 5.3.1, koji bi najboljim dobitkom transformacijskog kodiranja trebao osigurati odabir dviju komponenti mješavine, čiji će transformacijski koderi dani ulazni rezidual kodirati uz najmanju distorziju. Naravno, zbog statističke prirode TCG kriterija, i ovdje se konzistentnost s konceptom maksimiziranja TCG-a zadržava jednostavnom zamjenom varijanci u izrazu 5.4 sa kvadratima apsolutnih vrijednosti komponenti pojedinog transformiranog vektora. Ovakav predodabir dvije od m komponenti mješavine odgovara odabiru u otvorenoj petlji, koji se koristi kod inicijalnog združivanja izdvojenih reziduala i najbolje komponente mješavine. Svaki se odabrani transformirani rezidual entropijski kodira s q ECSQ kvantizatora, što rezultira s ukupno $2 \times q$ kvantiziranih kandidata. Inverznim procesiranjem kvantizirani se kandidati prevode u originalnu domenu i uspoređuju s ulaznim rezidualom u svrhu odabira kombinacije komponente mješavine i ECSQ kvantizatora, koja ulazni rezidual kodira uz

najmanju distorziju. Drugim riječima, nakon predodabira dvije od m komponenti mješavine u otvorenoj petlji, broj kandidata za procesiranje u zatvorenoj petlji smanjen je sa $m \times q$ na $2 \times q$.

Kombinirajući odabir u otvorenoj i zatvorenoj petlji, *2-best* verzija predloženog LSF kvantizatora bi, u usporedbi s $m \times q$ verzijom, trebala osigurati redukciju računske kompleksnosti algoritma kodiranja za faktor $m/2$.

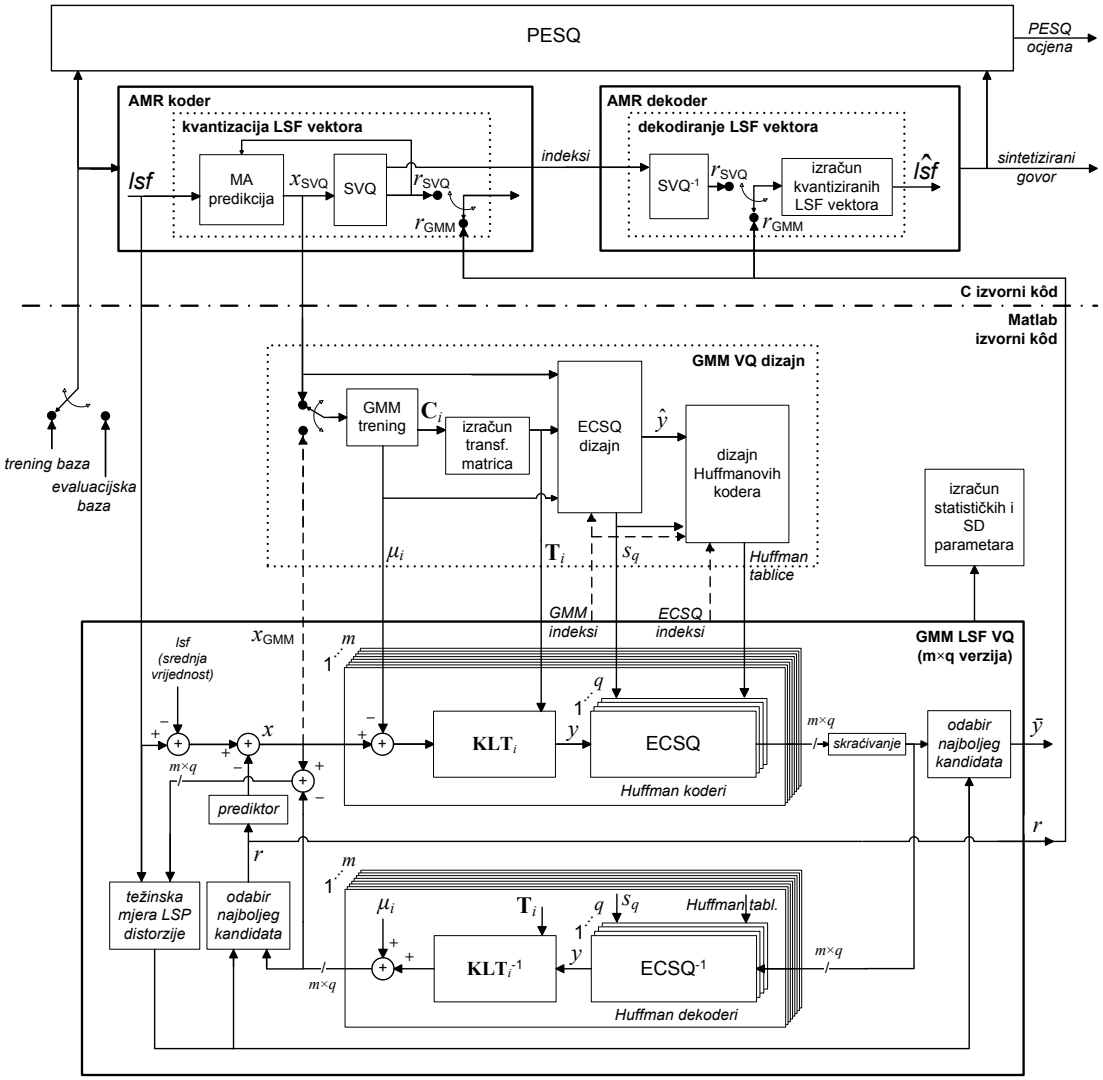
6. Simulacije i rezultati

Provedene simulacije obuhvaćaju odabir optimalnih parametara LSF VQ temeljenog na GMM, te usporedbu svojstava referentnog AMR kodera i nekoliko verzija modificiranog AMR kodera, koji za kvantizaciju spektralne ovojnice koriste algoritam temeljen na modelu s Gaussovim mješavinama. Vektorski kvantizator LSF parametara temeljen na GMM simuliran je u programskom paketu Matlab.

6.1. Simulacijski model

Pojednostavljena blok shema simulacijskog modela prikazana je na slici 6.1. Simulacijski model koristi $q = 4$ ECSQ kvantizatora i GMM sa $m = 8$ komponenti, čiji su elementi odgovarajućih matrica kovarijance izvan glavne dijagonale različiti od nule (engl. *full covariance matrices*). Zbog jednostavnijeg prikaza, slika se odnosi samo na AMR modove rada koji kodiraju jedan LSF vektor po okviru, odnosno modove u kojima referentni koder koristi SVQ metodu kvantizacije. Za môd 12k2, u kojem referentni koder zbog dvostruko veće učestalosti LPC analize koristi SMQ, dva rezultirajuća LSF vektora se GMM-temeljenim VQ kodiraju skupno, kao jedan vektor dvostruke duljine, dobiven nastavljanjem komponenti vektora trećeg pod-okvira na komponente vektora prvog pod-okvira. Ovakav pristup podrazumijeva i dvostruko veće dimenzije transformacijskih matrica, te dvostruki broj Huffmanovih tablica, koje kodiraju izlazne indekse ECSQ kvantizatora. Blok shema simulacijskog modela za môd 12k2 bi u svom prikazu trebala sadržavati blokove za konstrukciju i izdvajanje vektorâ iz vektora dvostruke duljine koji je procesiran u postupku kodiranja i dekodiranja.

GMM-temeljeni LSF VQ projektiran je (treniran) koristeći 56030 10-dimenzionalnih LSF vektora i odgovarajućih reziduala. Ti vektori čine bazu trening vektora koji se koriste u procesu učenja (treniranja) modela sa Gaussovim mješavinama, te projektiranja transformacijskih kodera odgovarajućih komponenti mješavine. Vektori trening baze izdvojeni su iz različitih muških i ženskih govornih sekvenci na nekoliko različitih jezika, koristeći standardiziranu implementaciju AMR kodera s pomičnim zarezom (engl. *floating point*) u C programskom jeziku [2]. Svi govorni signali uzorkovani su frekvencijom uzorkovanja 8 kHz.



Slika 6.1 *Blok shema simulacijskog modela.*

Izdvojeni reziduali (x_{SVQ}) korišteni su za treniranje inicijalnog GMM modela, čije su matrice kovarijance ($\mathbf{C}_i$) dalje korištene za izračun KLT transformacijskih matrica ($\mathbf{T}_i$). Transformacijskim matricama izračunavaju se reziduali u transformacijskoj domeni, koji su onda korišteni kao ulazni podaci za projektiranje ECSQ kvantizatora, odnosno određivanje njihovih kvantizacijskih koraka, s_q. Nakon kvantizacije, kvantizirani reziduali ($\hat{y}$) korišteni su za projektiranje Huffmanovih entropijskih kodera. U sljedećim iteracijama postupka projektiranja, kako je i objašnjeno u poglavlju 5.2, GMM model trenira se za reziduale (x_{GMM}) koji su, procesiranjem iste trening baze LSF vektora, producirani kvantizatorom s parametrima iz prethodne iteracije.

Na način istovjetan izdvajanju vektora za trening bazu, za objektivno vrednovanje svojstava modificiranog AMR kodera izdvojeno je dodatnih 18050 LSF vektora. Ti vektori sačinjavaju evaluacijsku bazu vektora, koji nisu dio trening baze. Procesiranjem

evaluacijske baze vektora, za konkretne parametre predloženog kvantizatora određuje se prosječna SD, koja reflektira učinkovitost samog LSF VQ. Objektivno vrednovanje utjecaja predloženog algoritma na učinkovitost cjelokupnog AMR kodera napravljena je primjenom PESQ algoritma. Usporedbom sintetiziranog i originalnog govornog signala izračunava se prosječna PESQ ocjena, koja odgovara učinkovitosti sustava u cjelini. U tu svrhu, referentni C izvorni kôd AMR kodera i dekodera modificiran je na način da zamijeni originalne kvantizirane LSF reziduale (r_{SVQ}) sa rezidualima koji su eksterno izračunati i kvantizirani u Matlabu (r_{GMM}).

6.2. Referentni sustav: standardni AMR koder

Kao referentni sustav korišten je standardizirani AMR koder. Prije usporedbe sa svojstvima modificiranog AMR kodera, sve korištene govorne sekvence kodirane su i dekodirane referentnim AMR koderom. Za dekodirane sekvence izmjerena je PESQ ocjena koja onda, kao pokazatelj učinkovitosti referentnog AMR kodera, služi za objektivno vrednovanje učinkovitosti cjelokupnog modificiranog sustava.

6.3. Modificirani sustav: *m*×*q* i *2-best* verzija AMR kodera

Dvije su verzije modificiranog sustava korištene za usporedbu s referentnim sustavom:

- *m*×*q* verzija – (slika 5.1), predstavlja kompleksniju verziju, koja pri odabiru najboljeg kvantiziranog vektora u zatvorenoj petlji evaluira LSP mjeru distorzije za svih 32 kvantizirana kandidata (8 komponenti mješavine x 4 ECSQ kvantizatora po svakoj komponenti),
- *2-best* verzija – (slika 5.3), predstavlja jednostavniju alternativu *m*×*q* verziji, koja pri odabiru najboljeg kvantiziranog vektora u zatvorenoj petlji, prvo od 8 komponenti mješavine odabire dvije najbolje primjenom TCG principa, te zatim evaluira LSP mjeru distorzije za samo 8 kvantiziranih kandidata (2 komponente mješavine x 4 ECSQ kvantizatora po svakoj komponenti).

6.4. Odabir izlazne entropije skalarnih kvantizatora

Kako je za simulaciju GMM-temeljenog LSF VQ odabran sustav sa $m = 8$ komponenti mješavine i $q = 4$ ECSQ kvantizatora, entropijsko kodiranje informacije o odabranoj komponenti mješavine i kvantizacijskog koraka zahtijeva aproksimativno 5 bitova po svakom ulaznom vektoru. Broj bitova potrebnih za kodiranje informacije o izabranoj kombinaciji razmjerno umanjuje dostupni broj bitova za kodiranje komponenti transformiranog vektora. Npr., u 10k2 môdu, ako se od maksimalnih 26 bita, dostupnih za kodiranje ovojnice u jednom okviru, 5 bitova upotrijebi za kodiranje informacije o odabranoj kombinaciji, tada samo 21 bit ostaje dostupno za kodiranje komponenti transformiranog vektora.

U svrhu odabira optimalne kombinacije izlaznih entropija ECSQ kvantizatora eksperimentirano je sa 20 različitih kombinacija navedenih u tablici 6.1. Kombinacije su prikazane kao korekcije u odnosu na maksimalno dostupan broj bitova za kodiranje ovojnice u pojedinom môdu, umanjen za broj bitova potrebnih za kodiranje odabrane

kombinacije komponente mješavine i koraka kvantizacije (u provedenim simulacijama: 5 bitova). Drugim riječima, maksimalno dostupni broj bitova u pojedinom môdu, umanjen za 5 bitova potrebnih za kodiranje odabrane kombinacije, predstavlja ciljnu entropiju koja se onda korigira za iznose korekcija prikazanih u tablici 6.1 da bi se dobile ukupne entropije pojedinih ECSQ kvantizatora. Zbog znatno veće rezolucije, za môd 12k2 korištene su dvostruke vrijednosti prikazanih korekcija.

Tablica 6.1 *Kombinacije ukupnih entropija ECSQ kvantizatorâ.*

asimetrične kombinacije					*simetrične kombinacije*				
indeks	*korekcije ciljne entropije*				*indeks*	*korekcije ciljne entropije*			
	q1	*q2*	*q3*	*q4*		*q1*	*q2*	*q3*	*q4*
1	0	-5	-10	-15	13	2	1	-1	-2
2	0	-4	-8	-12	14	3	1	-1	-3
3	0	-3	-6	-9	15	4	2	-2	-4
4	0	-2	-4	-6	16	5	2	-2	-5
5	0	-1	-2	-3	17	6	3	-3	-6
6	1	0	-1	-2	18	9	4	-4	-9
7	2	1	0	-1	19	12	6	-6	-12
8	3	2	1	0	20	15	7	-7	-15
9	6	4	2	0					
10	9	6	3	0					
11	12	8	4	0					
12	15	10	5	0					

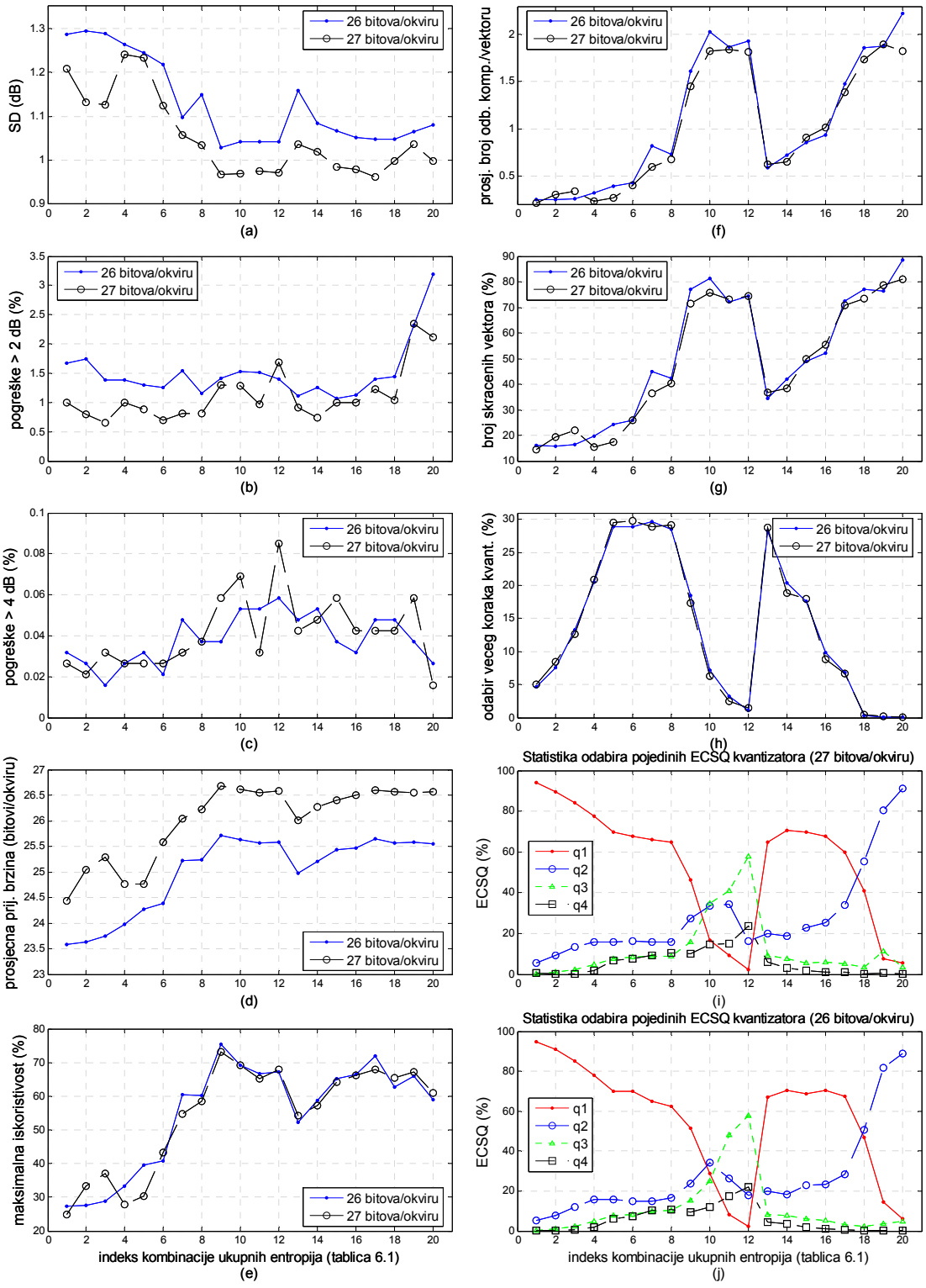
Sve testirane kombinacije mogu se općenito podijeliti u dvije skupine:

- ***asimetrične kombinacije*** (indeksi od 1 do 12), čije su ukupne entropije pojedinih ECSQ kvantizatora asimetrično raspoređene oko ciljne entropije, odnosno maksimalno dostupne duljine kodiranog vektora umanjene za 5 bitova potrebnih za kodiranje odabrane kombinacije komponente mješavine i ECSQ kvantizatora. Asimetrične kombinacije pokrivaju kombinacije čije su ukupne entropije manje ili jednake ciljnoj entropiji (npr., kombinacija s indeksom 1) do kombinacija čije su entropije redom veće ili jednake ciljnoj entropiji (npr., kombinacija s indeksom 12). Tako, npr., za môd 10k2, s ciljnom entropijom od 21 bit, kombinacija s indeksom 1 obuhvaća ECSQ kvantizatore s ukupnim entropijama 21−0, 21−5, 21−10 i 21−15 bitova, dok kombinacija s indeksom 12 obuhvaća ECSQ kvantizatore s ukupnim entropijama 21+15, 21+10, 21+5 i 21+0 bitova.
- ***simetrične kombinacije*** (indeksi od 13 do 20), čije su ukupne entropije pojedinih ECSQ kvantizatora simetrično raspoređene oko ciljne entropije. Npr., u môdu 10k2, kombinacija s indeksom 17 obuhvaća ECSQ kvantizatore s

ukupnim entropijama 21+6, 21+3, 21−3 i 21−6 bitova. Ove kombinacije u pravilu imaju dva ECSQ kvantizatora čije su ukupne entropije veće od ciljne entropije, te dva kvantizatora s ukupnim entropijama manjim od ciljne entropije. Zbog simetrije, s povećanjem entropije je razmak među simetričnim entropijama sve veći.

Simulacijski rezultati za svih 20 kombinacija ukupnih entropija prikazani su na slikama 6.2, 6.3, 6.4 i 6.5. Odgovarajući GMM-temeljeni LSF vektorski kvantizatori projektirani su u tri iteracije, koristeći princip najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni za inicijalno pridruživanje komponenti Gaussove mješavine rezidualnim LSF vektorima izdvojenim iz AMR kodera. Radi bolje razlučivosti i preglednijeg prikaza, simulacijski rezultati za različite rezolucije kodiranja spektralne ovojnice prikazani su odvojeno na spomenutim slikama.

Na slikama 6.2 i 6.3 skupno su prikazani simulacijski rezultati za AMR modove koji koriste 26 bitova (modovi: 10k2, 7k4, 6k7 i 5k9) i 27 bitova (môd 7k95) po okviru za kodiranje spektralne ovojnice. Kako spomenuti modovi koriste bliske rezolucije, simulacijski rezultati su također bliskih vrijednosti, što omogućava njihovo skupno prikazivanje bez značajnijeg gubitka na preglednosti rezultata. Slika 6.2(a) pokazuje najmanju spektralnu distorziju za asimetrične kombinacije, čije su ukupne entropije redom veće ili jednake broju bitova dostupnim za kodiranje komponenti transformiranog vektora (indeksi 9-12). Spomenute kombinacije zapravo najbolje kodiraju komponente transformiranog vektora, jer za kvantizaciju koriste najmanje (najfinije) korake kvantizacije. Posljedično, fini kvantizacijski koraci uzrok su dugim entropijskim kodovima koje je potrebno skratiti kako bi stali u fiksnu duljinu polja za prijenos kodirane ovojnice spektra. Slike 6.2(f) i 6.2(g) potvrđuju da spomenute kombinacije s najmanjom spektralnom distorzijom, u usporedbi s ostalim asimetričnim kombinacijama, imaju najveći broj skraćenih vektora (cca. 75%), te najveći prosječan broj odbačenih komponenti po vektoru (cca. 2 komponente). Očito je da je, kao posljedica KLT transformacije, puno značajnije precizno kodirati prvih nekoliko komponenti transformiranog vektora, nego izbjeći njegovo skraćivanje upotrebom grubog kvantizacijskog koraka. Međutim, kao posljedica intenzivnog skraćivanja transformiranih vektora, slike 6.2(b) i 6.2(c) za spomenute kombinacije u prosjeku pokazuju lagano povećanje broja kodiranih vektora sa spektralnom distorzijom većom od 2 i 4 dB. Također, na slici 6.2(e) vidljivo je da spomenute kombinacije imaju najveći postotak iskoristivosti dostupnih bitova, tj. da im je postotak vektora koji kodiraju maksimalnim brojem bitova najveći (cca. 70%). Kao direktna posljedica te činjenice, kombinacije 9-12 rezultirale su najvećim prosječnim prijenosnim brzinama (slika 6.2(d)). Npr., upotrebom kombinacije s indeksom 9 u modovima koji za kodiranje ovojnice koriste ukupno 26 bitova, 75% vektora kodirano je upotrebom svih 26 dostupnih bitova. Za ostale vektore, kao najbolji kvantizirani reprezentant originalnog vektora odabran je entropijski kôd kraći od 26 bitova. Za istu kombinaciju i AMR modove, modificirani AMR koder je pri kodiranju ovojnice spektra postigao prosječnu prijenosnu brzinu od 25,716 bitova/okviru. To je ujedno, za sve testirane kombinacije, prosječna prijenosna brzina koja je najbliža dostupnoj fiksnoj prijenosnoj brzini od 26 bitova/okviru u odgovarajućim modovima AMR kodera.



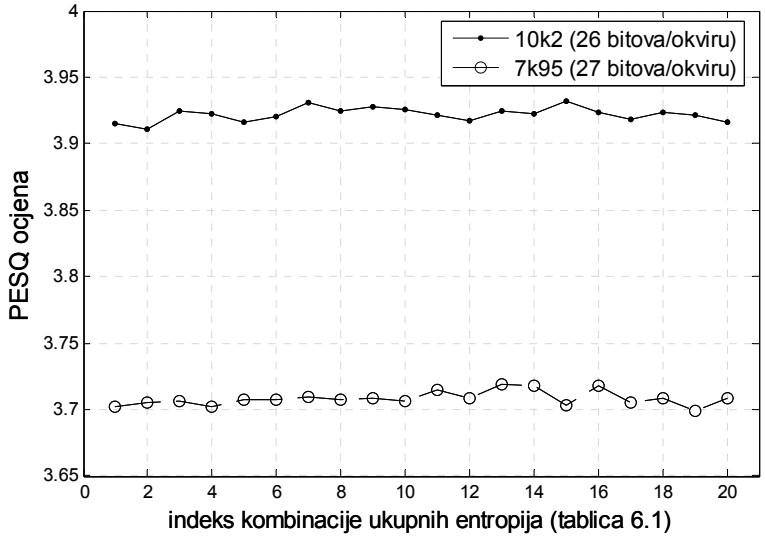
Slika 6.2 *Odabir optimalne kombinacije ukupnih entropija ECSQ kvantizatora. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 26 i 27 bitova/okviru.*

S daljnjim porastom ukupnih entropija (kombinacije 10, 11 i 12) vidljivo je lagano pogoršanje svojstava u smislu veće spektralne distorzije, koja je prvenstveno posljedica intenzivnijeg skraćivanja transformiranih vektora (slika 6.2(f)), te znatno smanjene upotrebe q1 ECSQ kvantizatora (slike 6.2(i) i 6.2(j)), zbog njegovog prefinog kvantizacijskog koraka i razmjerno predugih entropijskih kodova. Kako s porastom ukupne entropije q1 postaje sve manje upotrebljiv, tako se smanjuje broj maksimalno dugih kodiranih vektora (slika 6.2(e)), što rezultira suboptimalnim prosječnim prijenosnim brzinama (slika 6.2(d)).

Simetrične kombinacije (indeksi 13-20) općenito pokazuju nešto lošije rezultate od asimetričnih kombinacija s visokim entropijama, ali i znatno bolje rezultate u usporedbi s asimetričnim kombinacijama s nižim ukupnim entropijama (indeksi 1-8). Prednost nad asimetričnim kombinacijama s nižim ukupnim entropijama zasigurno im daje činjenica da barem dva ECSQ kvantizatora imaju ukupnu entropiju veću od dostupne fiksne dužine kôda, što im omogućava preciznije kodiranje značajnijih komponenti transformiranog vektora. Kao što je i očekivano, simetrične kombinacije također pokazuju bolje rezultate za kombinacije koje koriste manje (finije) kvantizacijske korake. Od simetričnih kombinacija, kombinacija s indeksom 17 pokazuje najmanju spektralnu distorziju, te najveću maksimalnu iskoristivost i prosječnu prijenosnu brzinu. Ponovno je vidljiv kompromis na koji treba paziti pri povećanju ukupnih entropija. Njihovo daljnje povećanje u kombinacijama s indeksom 19 i 20 rezultira znatnijim povećanjem spektralne distorzije, te izrazitim povećanjem broja kodiranih vektora sa spektralnom distorzijom većom od 2 dB. Znatnija degradacija u odnosu na asimetrične kombinacije s visokim entropijama prvenstveno je posljedica manjeg broja kvantizatora s entropijama većim od raspoložive dužine kôda. Sukladno ponašanju asimetričnih kombinacija, s povećanjem entropija, q1 se koristi sve manje. U kombinacijama s indeksima 19 i 20, q1 se koristi izrazito rijetko (5-15%). Kako q3 i q4 imaju veoma malu ukupnu entropiju i velik korak kvantizacije, učestalost njihove upotrebe je također minimalna (0-10%). Posljedično, na raspolaganju ostaje samo q2, čija frekvencija odabira raste na 80-90%. Upotreba q1 u ovim kombinacijama uzrokuje znatna skraćenja transformiranog vektora koja rezultiraju većim brojem kodiranih vektora s pogreškama iznad 2 dB, dok upotreba q4 rezultira većim brojem neiskorištenih bitova, što u konačnici smanjuje maksimalnu iskoristivost i prosječnu brzinu kodera. Oba slučaja odgovarajuće kombinacije čine znatno lošijim izborom za kodiranje ovojnice spektra. Na slikama 6.2(i) i 6.2(j) vidljivo je da kombinacije s najboljim rezultatima (indeks 9 za asimetrične kombinacije, te indeks 17 za simetrične kombinacije) ravnomjernije koriste sve ECSQ kvantizatore s entropijama većim od dostupne duljine kôda.

Zanimljivo je primijetiti da je kod kombinacija s ukupnim entropijama koje su relativno bliske dostupnoj duljini kôda (prvenstveno kombinacije s indeksima 5-8 i 13) relativno velik postotak vektora (cca. 30%) kodiran kvantizatorom s većim kvantizacijskim korakom, iako je kao alternativa postojao vektor kodiran s manjim kvantizacijskim korakom i jednakim ili manjim brojem odbačenih komponenti transformiranog vektora (slika 6.2(h)).

Spektralna distorzija, kao posljedica kvantizacije i kodiranja spektralne ovojnice, nije nužno povezana sa svojstvima cjelokupnog kodera govornog signala, pogotovu kod kodera sa zatvorenom petljom (engl. *closed loop coders*) kao što je to AMR koder. Iz tog je razloga optimalnost testiranih kombinacija ukupnih entropija također evaluirana korištenjem PESQ algoritma na modificiranom AMR koderu sa LSF VQ temeljenim na Gaussovim mješavinama. Iako različite kombinacije ukupnih entropija pokazuju znatno različita svojstva u kodiranju spektralne ovojnice, ta razlika nije uočljiva u PESQ rezultatima modificiranog AMR kodera. Simulacijski rezultati za modove 10k2 i 7k95, koji za kodiranje spektralne ovojnice koriste 26, odnosno 27 bitova po okviru, prikazani su na slici 6.3. PESQ rezultati pokazuju neznatnu varijaciju unutar 0.021 za različite kombinacije ukupnih entropija. Takvi rezultati navode na zaključak da su svojstva AMR kodera uglavnom ograničena pogreškom kvantizacije pobude, a ne kvantizacijskom pogreškom kodiranja spektralne ovojnice.



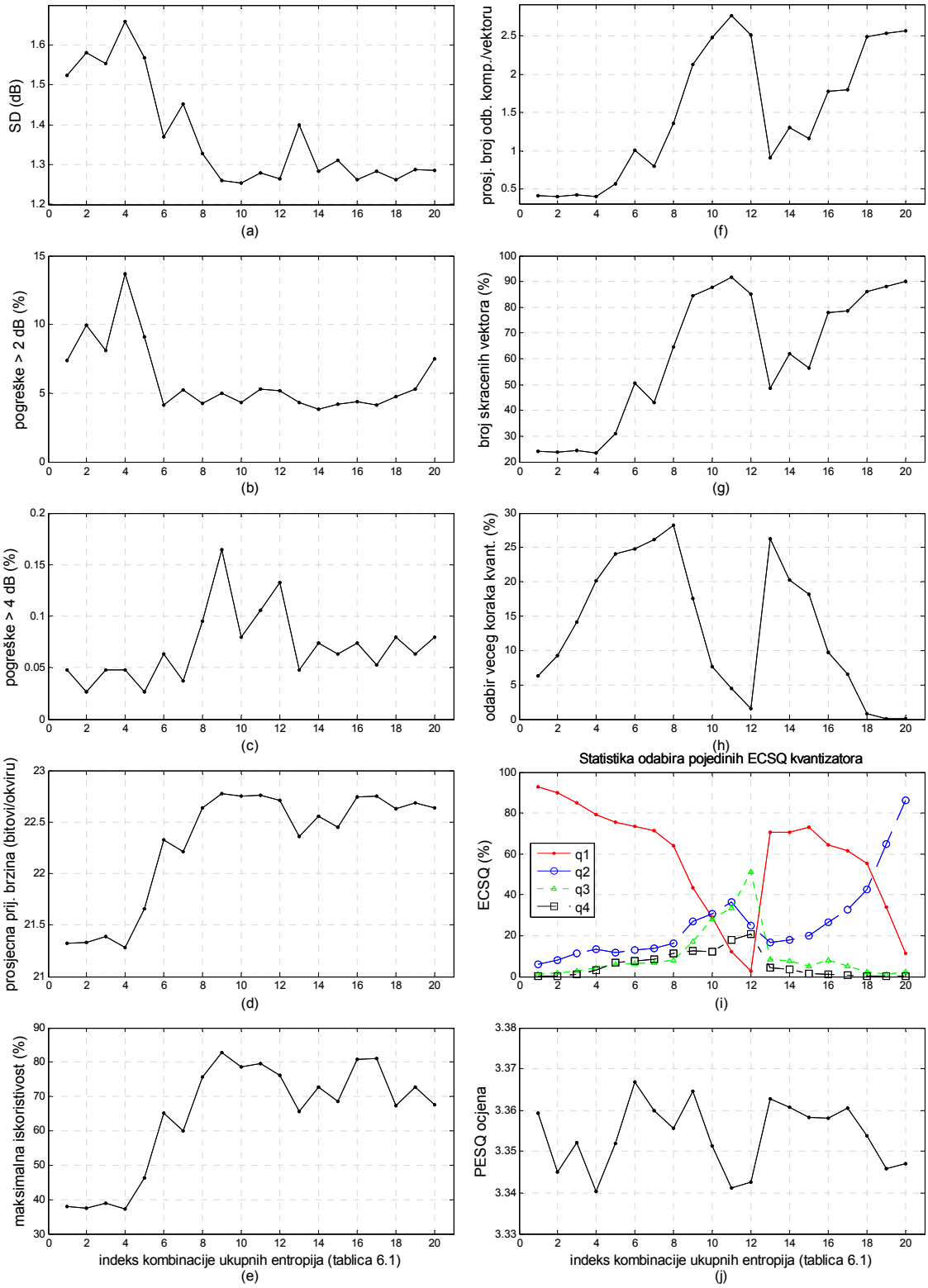
Slika 6.3 *Simulacijski rezultati PESQ ocjene modificiranog AMR kodera pri odabiru optimalne kombinacije ukupnih entropija ECSQ kvantizatora (modovi 10k2 i 7k9, koji koriste 26, odnosno 27 bitova/okviru za kodiranje ovojnice).*

Rezultati simulacija za AMR modove koji koriste 23 bita po okviru za kodiranje spektralne ovojnice (5k15 i 4k75) prikazani su na slici 6.4. Općenito je vidljivo da rezultati za spomenute modove slijede iste trendove, već opisane u prethodnoj analizi rezultata za modove sa 26 i 27 bitova/okviru. Najbolje rezultate opet postižu kombinacije s visokim ukupnim entropijama (indeksi 9-12, 16-20) uz nešto povećani broj kodiranih vektora sa spektralnom distorzijom iznad 2 i 4 dB. Ponovno asimetrična kombinacija s indeksom 9 postiže najveću prosječnu prijenosnu brzinu od 22,774 bitova/okviru, s najvećim brojem vektora (83%) kodiranih s maksimalno dostupnih 23 bita. Spomenuta kombinacija također intenzivno primjenjuje tehniku skraćivanja na 85% vektora s prosječno 2,13 odbačenih komponenti po vektoru. Vidljivi su jednaki trendovi odabira pojedinih ECSQ kvantizatora, gdje kombinacije s ravnomjernijim odabirom postižu bolje rezultate. PESQ rezultati također zanemarivo osciliraju unutar razlike od

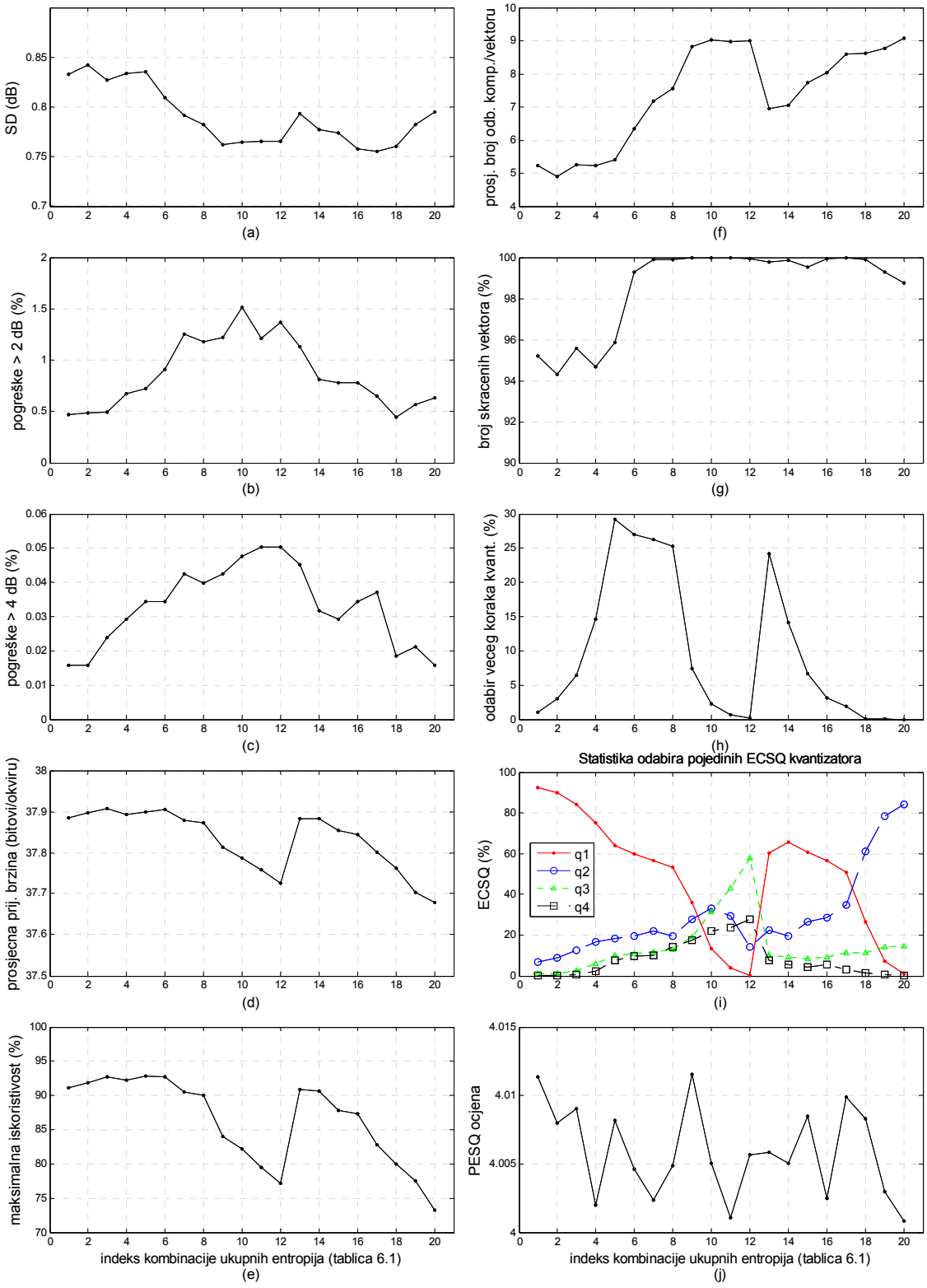
0.027, s tim da bolje rezultate postižu kombinacije s ukupnim entropijama koje su bliske ili nešto više od raspoložive duljine kôda.

Simulacije za AMR môd 12k2 koriste 38 bitova po okviru za kodiranje dva skupa LSF parametara. Odgovarajući rezultati prikazani su na slici 6.5. Slično ranije opisanim trendovima, i ovaj môd najmanju distorziju (SD) pokazuje za kombinacije s visokim ukupnim entropijama (indeksi 9-12, 16-18), koje rezultiraju povećanim brojem kodiranih vektora sa spektralnom distorzijom iznad 2 i 4 dB. Za razliku od ostalih modova, kombinacije s indeksima 19 i 20 pokazuju nešto veći prirast distorzije, popraćen manjim brojem kodiranih vektora sa spektralnom distorzijom iznad 2 i 4 dB. Uzimajući u obzir i iskorištenost raspoložive brzine prijenosa, kombinacija s indeksom 9 predstavlja dobar kompromis između distorzije i postignute prijenosne brzine, odnosno maksimalne iskoristivosti dodijeljenih bitova. Spomenuta kombinacija tehniku skraćivanja primjenjuje na svim vektorima s prosječno 8.81 odbačenih komponenti po vektoru dvostruke duljine (dva skupa LSF parametara). Najbolje rezultate ponovno postižu kombinacije s ravnomjernijim odabirom ECSQ kvantizatora, koji slijedi opisane trendove ostalih modova. PESQ rezultati također zanemarivo osciliraju unutar razlike od 0.012, bez posebno izraženih trendova.

Provedenom analizom, uspoređujući rezultate testiranih kombinacija ukupnih entropija u svim podržanim modovima AMR kodera, zaključeno je da kombinacija s indeksom 9 rezultira najboljim kompromisom među mjerenim parametrima modificiranog AMR kodera s LSF VQ-om temeljenim na Gaussovim mješavinama. Odabrana kombinacija u pravilu postiže najmanju spektralnu distorziju uz nešto povećani broj kodiranih vektora sa spektralnom distorzijom iznad 2 i 4 dB. Također, u većini modova, ona postiže najveće prosječne prijenosne brzine, popraćene najvećom maksimalnom iskoristivošću, pri tome intenzivno primjenjujući tehniku skraćivanja transformiranih vektora. Rezultati spomenute kombinacije popraćeni su relativno ravnomjernim odabirom pojedinih ECSQ kvantizatora. Npr. za 12k2 môd: q1 (s najvećom izlaznom entropijom, odnosno najfinijim korakom kvantizacije) odabran je u 36% okvira, q2 – u 28%, q3 – u 19% i q4 (s najmanjom izlaznom entropijom, odnosno najgrubljim korakom kvantizacije) – u 18% okvira. Neovisno o môdu rada AMR kodera, PESQ ocjene pokazuju neznatne varijacije za različite kombinacije ukupnih entropija, te stoga nisu korištene kao kriterij pri odabiru najbolje kombinacije. Iz navedenih razloga, kombinacija s indeksom 9 i ukupnim entropijama l+6, l+4, l+2 i l+0, odnosno l+12, l+8, l+4 i l+0 za môd 12k2, korištena je u svim ostalim simulacijama čiji opis rezultata slijedi. Pri tome je l jednak maksimalno dostupnom broju bitova za kodiranje ovojnice u pojedinom môdu, umanjen za prosjek od 5 bitova potrebnih za kodiranje odabrane kombinacije komponente mješavine i koraka kvantizacije.



Slika 6.4 *Odabir optimalne kombinacije ukupnih entropija ECSQ kvantizatora. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 23 bita/okviru.*



Slika 6.5 *Odabir optimalne kombinacije ukupnih entropija ECSQ kvantizatora. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 38 bitova/okviru.*

6.5. Utjecaj kriterija inicijalnog pridruživanja odgovarajućih komponenti mješavine izdvojenim rezidualnim vektorima

Pri odabiru najboljeg od tri kriterija inicijalnog pridruživanja (opisanih u poglavlju 5.3) u otvorenoj petlji, za svaki opisani kriterij projektirane su $m \times q$ varijante GMM LSF VQ-ra, čije rezolucije odgovaraju podržanim prijenosnim brzinama kodirane spektralne ovojnice u AMR koderu (38, 27, 26 i 23 bita/okviru). Pri tome je postupak projektiranja proveden u samo jednoj iteraciji, koja parametre projektiranog VQ izračunava direktno iz rezidualnih LSF vektora izdvojenih iz AMR kodera. Na taj način, odabrani kriterij inicijalnog pridruživanja značajnije afektira svojstva kodera, jer o njemu direktno ovise dizajn Huffmanovih koderâ i ECSQ kvantizatora, projektiranih primjenom kombinacije ukupnih entropija s indeksom 9 (tablica 6.1). Svojstva projektiranih vektorskih kvantizatora evaluirana su procesiranjem govornih sekvenci iz evaluacijske baze.

Tablica 6.2 *Ovisnost spektralne distorzije i PESQ ocjene GMM LSF VQ o kriteriju inicijalnog pridruživanja odgovarajućih komponenti mješavine izdvojenim rezidualnim vektorima.*

brzina (bitovi/okviru) / AMR môd	**SD (dB)**			**p2 (%)**			**p4 (%)**			**PESQ**		
	max. P	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***
38 / 12k2	0,780	0,782	0,780	2,344	2,389	2,071	0,064	0,085	0,074	4,002	3,998	3,992
27 / 7k95	1,051	1,044	1,030	5,505	5,495	4,153	0,228	0,212	0,170	3,699	3,695	3,694
26 / 10k2, 7k4, 6k7, 5k9	1,103	1,104	1,092	5,882	6,173	4,683	0,217	0,255	0,154	3,911	3,911	3,909
23 / 5k15, 4k75	1,313	1,345	1,292	9,547	10,925	7,759	0,200	0,440	0,164	3,320	3,308	3,332

Tablica 6.3 *Ovisnost svojstava GMM LSF VQ o kriteriju inicijalnog pridruživanja odgovarajućih komponenti mješavine ulaznim rezidualnim vektorima.*

brzina (bitovi/okviru) / AMR môd	**prosječna brzina (bitovi/okviru)**			**max. iskorištenje (%)**			**prosj. broj odb. komp./vektoru**			**broj skraćenih vektora (%)**		
	max. P	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***	***max. P***	***min. E***	***TCG***
38 / 12k2	37,743	37,755	37,785	78	79	83	9,37	9,18	9,24	100	100	100
27 / 7k95	26,649	26,622	26,645	73	71	76	2,19	2,06	1,92	84	80	78
26 / 10k2, 7k4, 6k7, 5k9	25,688	25,657	25,685	76	73	76	2,30	2,18	2,09	86	82	81
23 / 5k15, 4k75	22,747	22,699	22,752	79	76	81	2,85	2,73	2,56	92	89	88

Rezultati simulacija pri odabiru najboljeg kriterija inicijalnog pridruživanja prikazani su u tablicama 6.2 i 6.3. Za svaki od opisanih kriterija:

- principa najveće vjerojatnosti (*max. P*),
- principa najmanje energije u transformiranoj domeni (*min. E*) i

- principa najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni (*TCG*),

procesiranjem evaluacijske baze mjerene su prosječne vrijednosti SD i odgovarajućih p2 i p4 postotaka kodiranih vektora, PESQ ocjena, postignuta prosječna prijenosna brzina i broj vektora koji u potpunosti iskorištavaju dodijeljene bitove, te postotak skraćenih vektora s prosječnim brojem odbačenih komponenti po vektoru. Spomenuta mjerenja provedena su za sve četiri podržane brzine prijenosa kodirane spektralne ovojnice u AMR koderu.

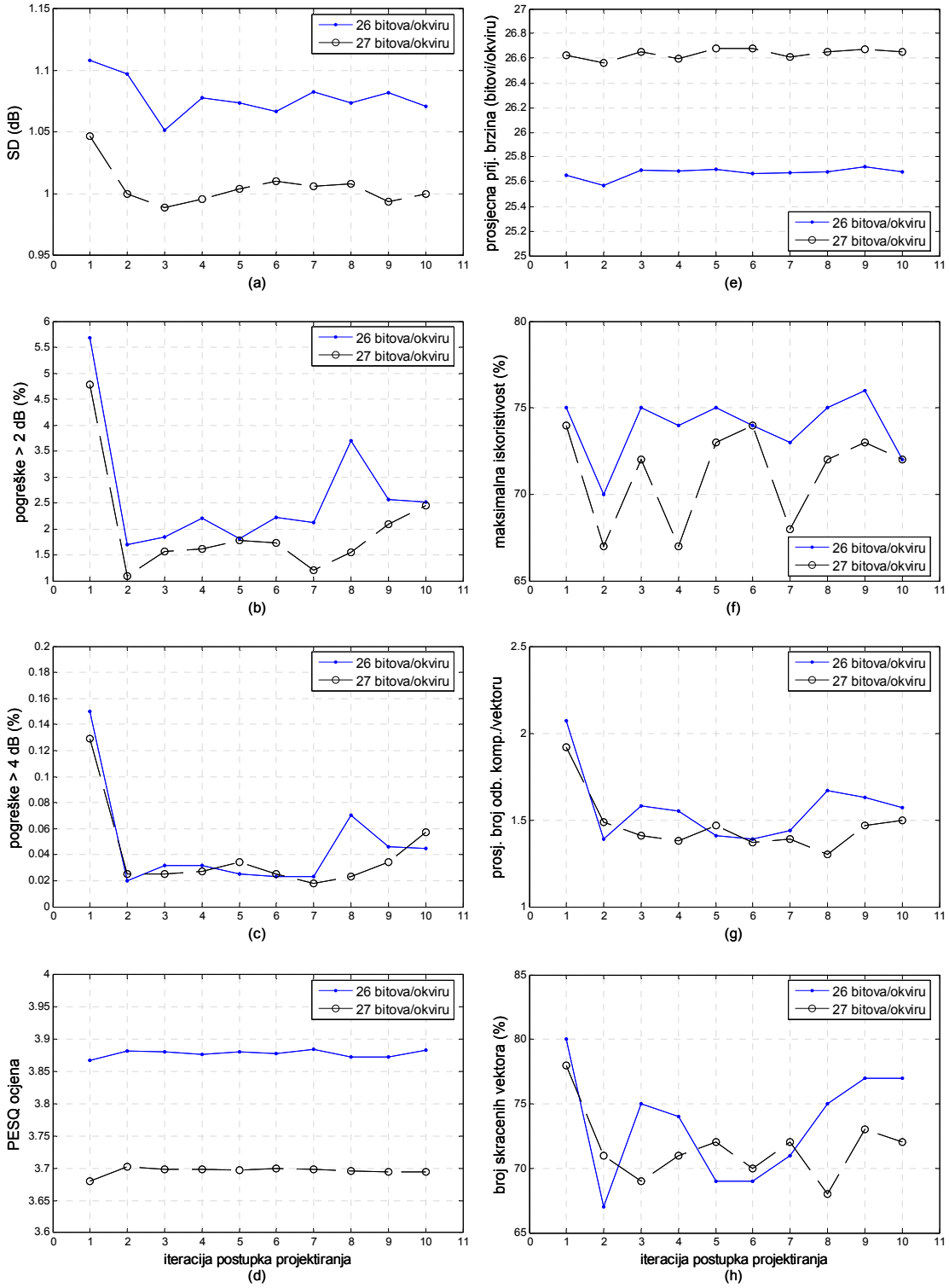
Simulacijski rezultati pokazuju da se primjenom *TCG* principa u pravilu ostvaruje nešto manja (do 0.053 dB) spektralna distorzija, uz značajno manji broj kodiranih vektora sa spektralnom distorzijom većom od 2 i 4 dB (do $\Delta p2 = 3.166\%$, odnosno $\Delta p4 = 0.276\%$ manji u odnosu na ostala dva principa). Vidljivo je da primjena *TCG* principa rezultira s najmanjim brojem skraćenih vektora uz najmanji prosječan broj odbačenih komponenti po vektoru. Uz najveći broj okvira koji ovojnicu kodiraju iskorištavajući sve dodijeljene bitove (max. iskorištenje), primjenom ovog principa uglavnom se postižu nešto veće prosječne prijenosne brzine. Ponovno, PESQ ocjene pokazuju zanemarive oscilacije bez jasnih trendova, te u tom smislu i nemaju težinu pri određivanju najboljeg kriterija inicijalnog pridruživanja.

S obzirom na opisane rezultate simulacija, princip najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni smatra se boljim u odnosu na ostala dva principa, te je korišten za inicijalno pridruživanje komponenti Gaussove mješavine izdvojenim rezidualima u svim ostalim simulacijama, čiji opis rezultata slijedi.

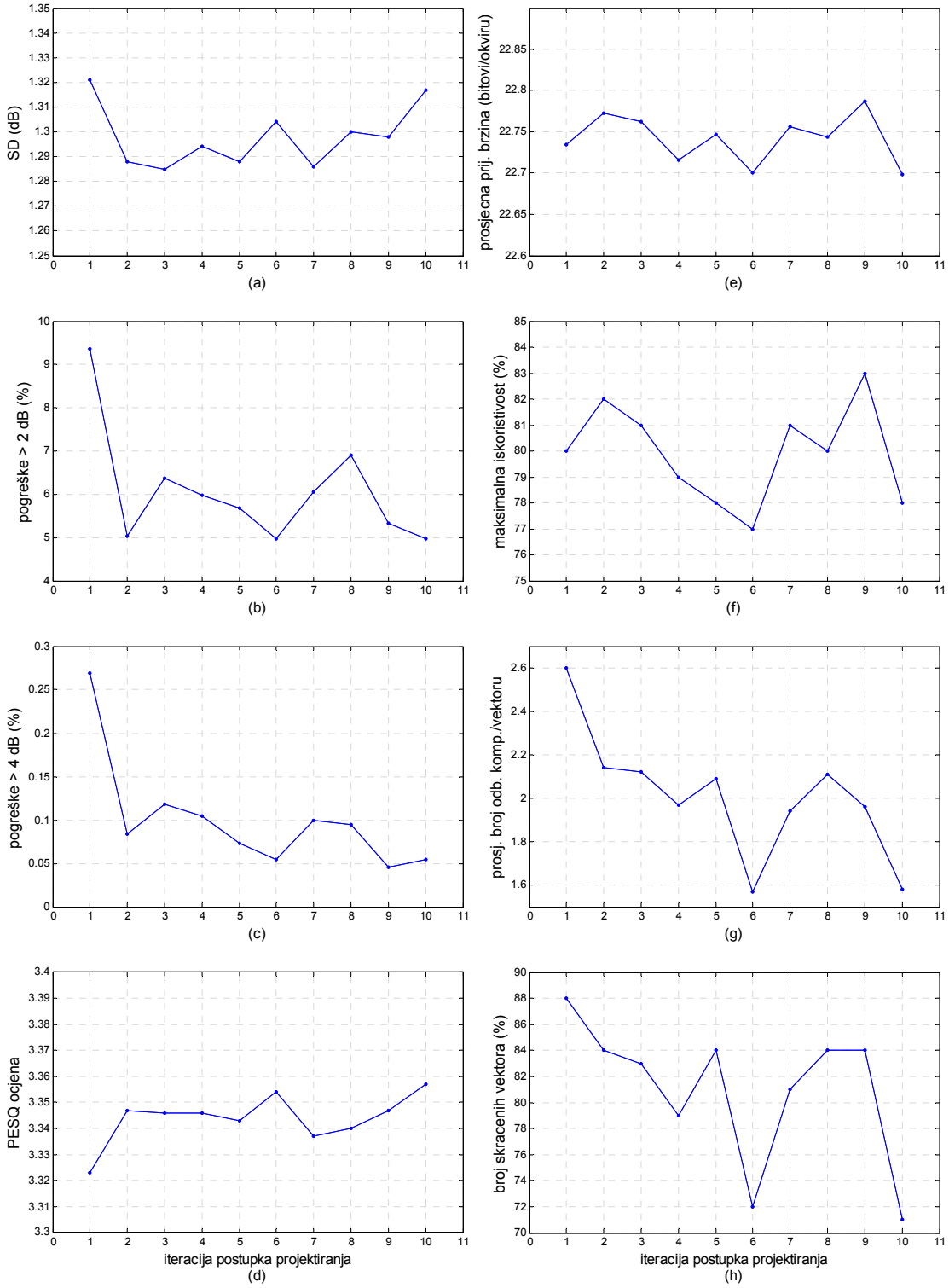
6.6. Utjecaj iteriranja postupka projektiranja kvantizatora

Kako je i opisano u poglavlju 5.2, postupak projektiranja GMM-temeljenog LSF VQ iterativno se ponavlja nekoliko puta u svrhu optimizacije parametara kvantizatora. Pri tome, svaka sljedeća iteracija (osim prve) kao ulazne parametre koristi reziduale istih LSF vektora i indekse najboljih kombinacija komponente mješavine i ECSQ kvantizatora, reasocirane kvantizatorom iz prethodne iteracije. U svakoj iteraciji, promijenjeni parametri kvantizatora rezultiraju drugačijim rezidualima i reasociranim indeksima, koji onda u sljedećoj iteraciji opet mijenjaju parametre projektiranog kvantizatora.

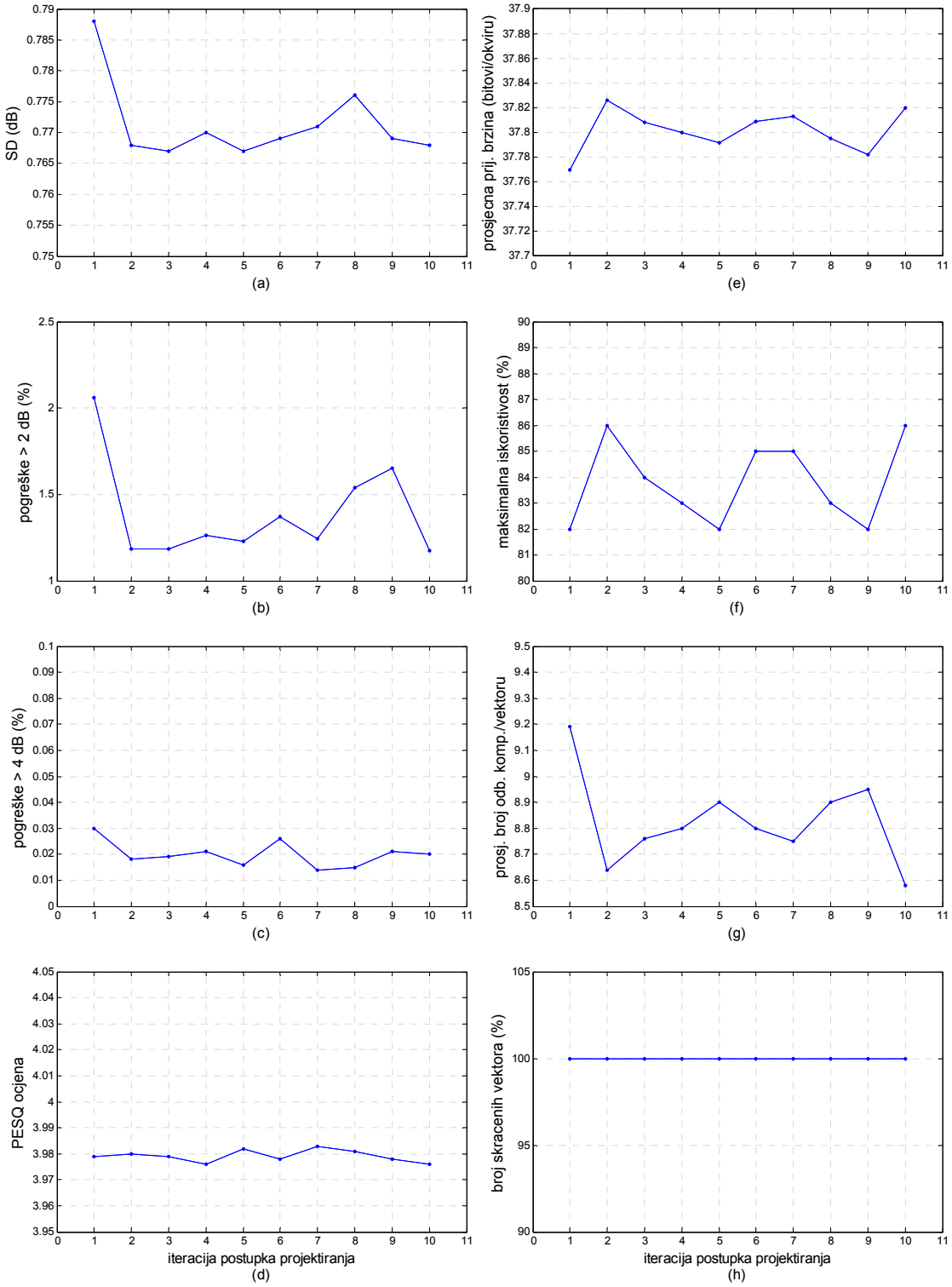
Učinkovitost takvog iterativnog pristupa promatrana je na simulacijskim rezultatima, prikazanim na slikama 6.6 do 6.11. Navedene slike prikazuju svojstva i statističke podatke modificiranog AMR kodera, u ovisnosti o broju iteracija (1-10) postupka projektiranja $m \times q$ varijante GMM LSF VQ, koristeći kombinaciju ukupnih entropija pod indeksom 9 (tablica 6.1) i princip najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni za inicijalno pridruživanje komponenti Gaussove mješavine rezidualnim LSF vektorima izdvojenim iz AMR kodera.



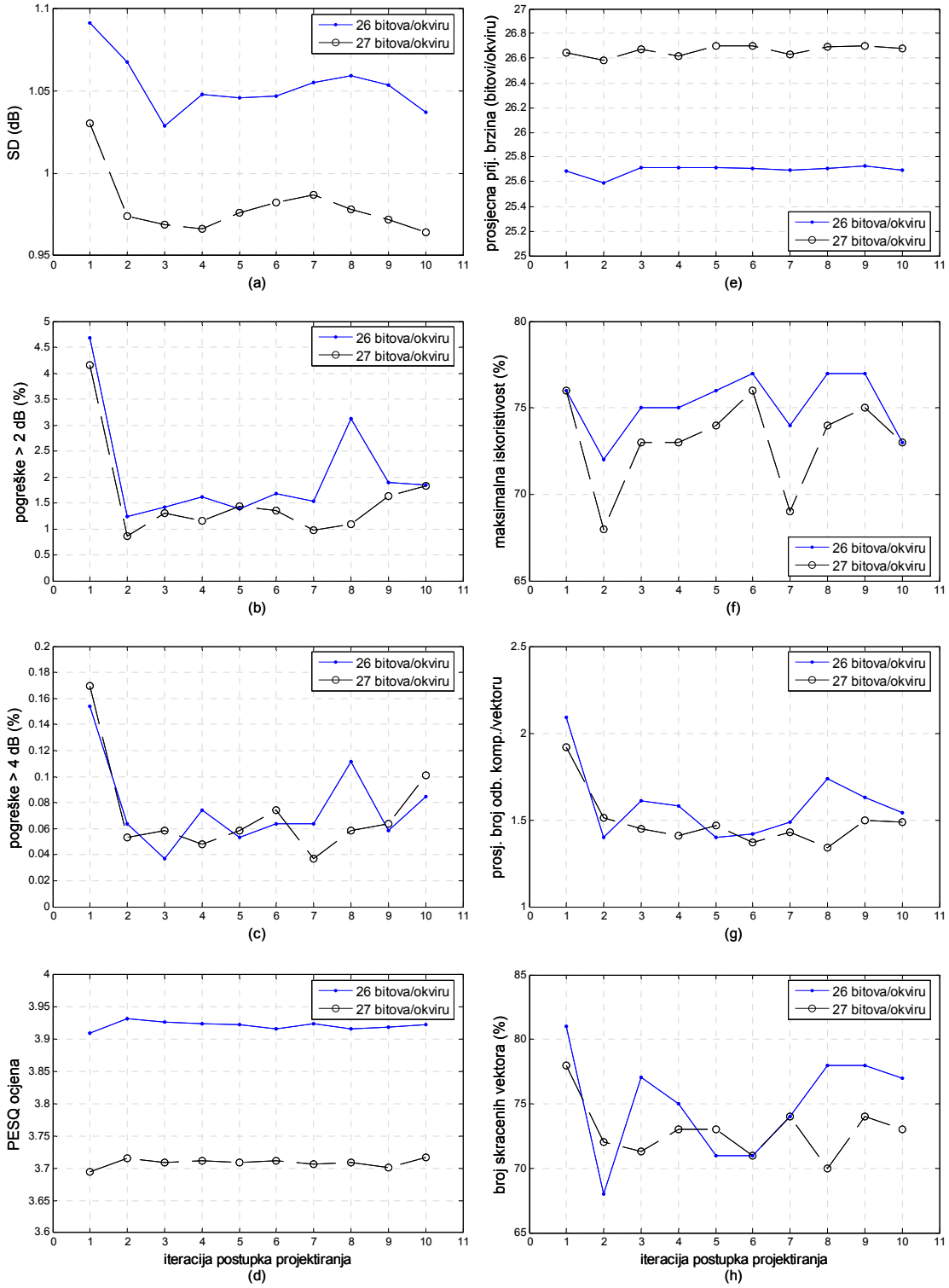
Slika 6.6 *Svojstva modificiranog AMR kodera, evaluirana nad trening bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 26 i 27 bitova/okviru.*



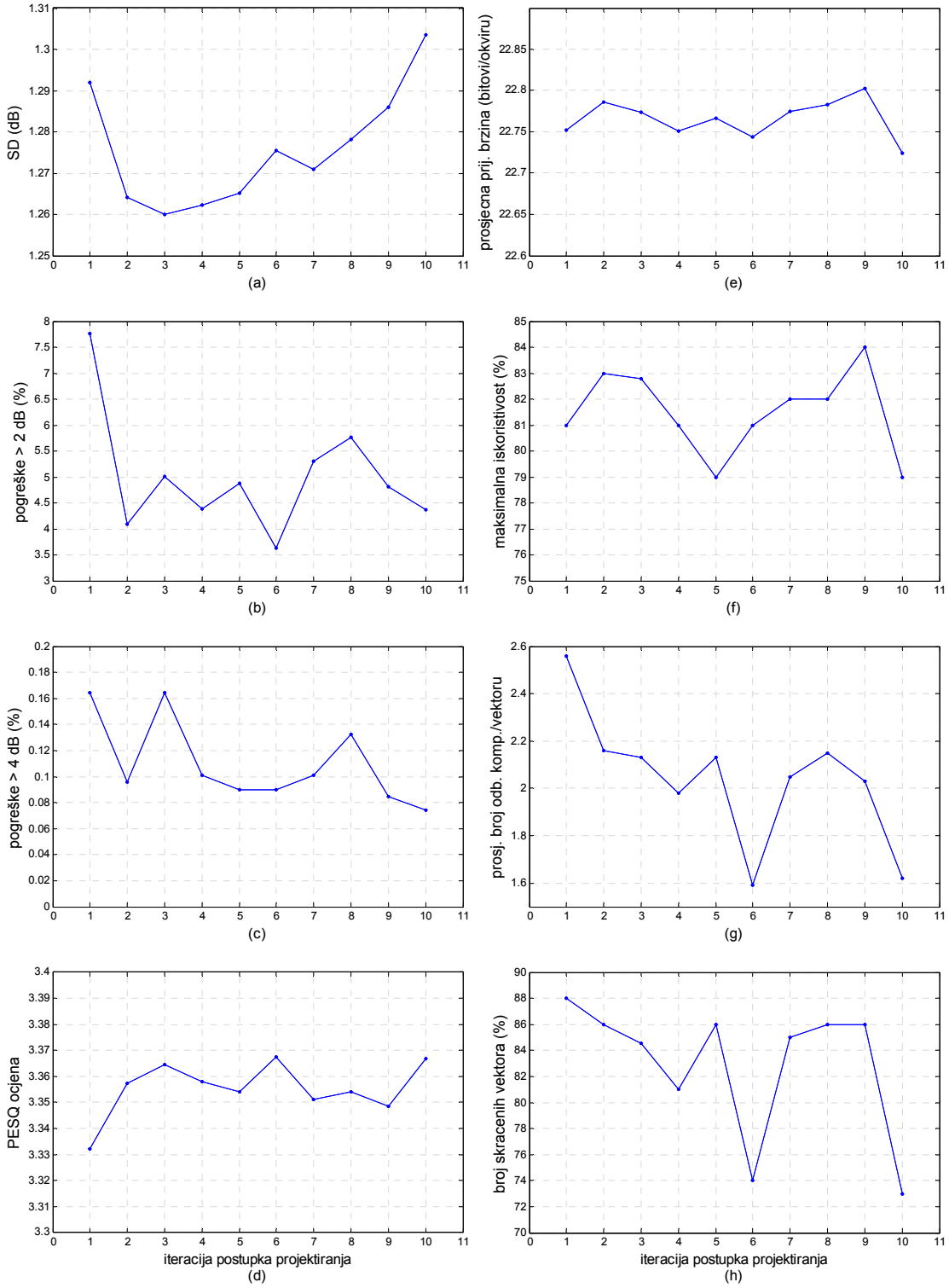
Slika 6.7 *Svojstva modificiranog AMR kodera, evaluirana nad trening bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 23 bita/okviru.*



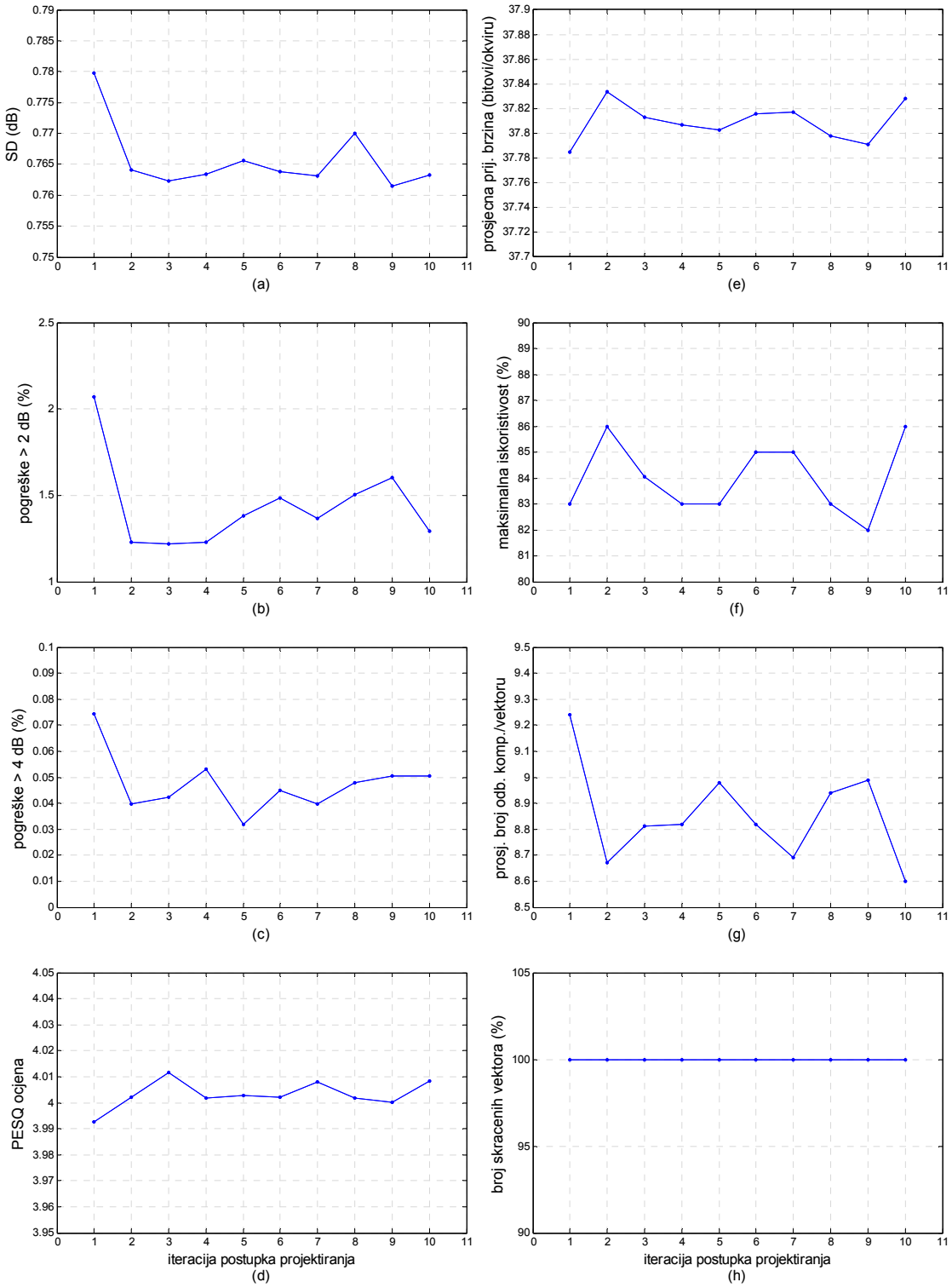
Slika 6.8 *Svojstva modificiranog AMR kodera, evaluirana nad trening bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 38 bitova/okviru.*



Slika 6.9 *Svojstva modificiranog AMR kodera, evaluirana nad evaluacijskom bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 26 i 27 bitova/okviru.*



Slika 6.10 *Svojstva modificiranog AMR kodera, evaluirana nad evaluacijskom bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 23 bita/okviru.*



Slika 6.11 *Svojstva modificiranog AMR kodera, evaluirana nad evaluacijskom bazom govornih sekvenci, u ovisnosti o broju iteracija postupka projektiranja GMM LSF VQ. Simulacijski rezultati za modove koji za kodiranje ovojnice koriste 38 bitova/okviru.*

Naravno, postupak projektiranja iteriran je nad trening bazom, te su svojstva rezultirajućih kvantizatora evaluirana na obje baze govornih sekvenci. Na slikama 6.6, 6.7 i 6.8 prikazana su svojstva modificiranog AMR kodera evaluirana nad trening bazom, dok slike 6.9, 6.10 i 6.11 prikazuju svojstva istog kodera evaluirana nad evaluacijskom bazom govornih sekvenci. Zbog preglednosti, rezultati pojedinih môdova ponovno su prikazani odvojeno, na zasebnim slikama.

Prikazani simulacijski rezultati, evaluirani nad obje baze, pokazuju kako je već tri iteracije dovoljno za izračun optimalnih parametara kvantizatora koji minimiziraju SD. Pri tome je i odgovarajući broj pogrešaka iznad 2 i 4 dB redovito minimalan ili među nižim vrijednostima u testiranom rasponu iteracija. Daljnje iteriranje postupka projektiranja pokazuje blago oscilatorno ponašanje bez dosljedne konvergencije, te ne rezultira daljnjim poboljšanjem svojstava modificiranog AMR kodera. To je i razumljivo, jer svaka iteracija istovremeno mijenja i model Gaussove mješavine (zbog nove trening baze LSF reziduala) i ECSQ kvantizatore.

Također je vidljivo da rezultati simulacija s tri iteracije postupka projektiranja pokazuju najbolji kompromis s obzirom na postignutu prijenosnu brzinu i njoj odgovarajuću maksimalnu iskoristivost dodijeljenih bitova. Npr. na slikama 6.7 i 6.10 vidljivo je da kvantizator projektiran u devet iteracija postiže nešto višu prijenosnu brzinu i više iskorištava dodijeljene bitove, ali rezultira nešto većom spektralnom distorzijom, što je posljedica nešto većeg broja skraćenih vektora. Slično vrijedi i za kompromis s obzirom na broj skraćenih vektora i, sa njim povezanim, prosječnim brojem odbačenih komponenti po vektoru. Na istim slikama, vidljivo je da kvantizator projektiran u šest iteracija skraćuje vrlo mali broj vektora, uz najniži prosječan broj odbačenih komponenti po vektoru, ali zbog toga postiže manju maksimalnu iskoristivost i prosječnu brzinu prijenosa, što rezultira nešto većom spektralnom distorzijom. Kao i kod ranije opisanih simulacijskih rezultata, PESQ ocjene pokazuju zanemarive oscilacije bez jasnih trendova, te u tom smislu i nemaju težinu pri određivanju optimalnog broja iteracija postupka projektiranja.

S obzirom na navedeno, svi GMM LSF VQ-ri u simulacijama čiji opis rezultata slijedi projektirani su u tri iteracije postupka projektiranja.

6.7. Usporedba referentnog i modificiranog kodera koji koriste jednak broj bitova po LSF vektoru

Nakon odabira najbolje kombinacije ukupnih entropija ECSQ kvantizatora (kombinacija s indeksom 9 u tablici 6.1), zatim najboljeg kriterija inicijalnog pridruživanja komponenti mješavine izdvojenim rezidualima (princip najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni), te optimalnog broja iteracija postupka projektiranja GMM LSF VQ-a (3), izabrani parametri korišteni su za usporedbu svojstava referentnog i modificiranih verzija AMR kodera, u svim podržanim modovima rada. Pri tome su modificirani AMR koderi za kvantizaciju spektralne ovojnice u odgovarajućem môdu rada koristili prijenosnu brzinu jednaku brzini koju koristi referentni AMR koder (tablice 3.1 i 3.2). Naravno, unutar spomenute prijenosne brzine, modificirane verzije također prenose i entropijski kôd kombinacije indeksa

odabrane komponente mješavine i broja koji indeksira jedan od četiri skupa Huffmanovih tablica.

Parametri učinkovitosti referentnog AMR kodera, te rezultati simulacija za *m*×*q* verziju modificiranog kodera, prikazani su u tablici 6.4. Spomenuta verzija, u svrhu odabira najboljeg kandidata u zatvorenoj petlji, izračunava težinsku mjeru LSP distorzije za svih 32 kvantizirana kandidata (8 komponenti mješavine × 4 skupa Huffmanovih kodera). Vidljivo je da se razlika u PESQ ocjeni, ovisno o odabranom môdu rada, kreće u granicama od -0.012 do 0.013. Kako prosječna razlika u PESQ ocjeni za sve modove rada iznosi 0.003 u korist *m*×*q* verzije AMR kodera, može se zaključiti da promjena u načinu kodiranja spektralne ovojnice nije utjecala na PESQ ocjenu cjelokupnog kodera. Za razliku od PESQ ocjene, vrijednosti SD pokazuju znatno veće razlike. Ovisno o odabranom môdu rada, predložena *m*×*q* verzija GMM LSF VQ reducira spektralnu distorziju za 0.105 – 0.292 dB, što u prosjeku daje 0.208 dB manju SD u odnosu na referentni AMR koder. Također je vidljiva značajna redukcija postotka kodiranih vektora sa spektralnom distorzijom većom od 2 dB (0.644% - 10.65%) i 4 dB (do 0.032%). Samo modovi s najmanjom prijenosnom brzinom (5k15 i 4k75) pokazuju nešto veći postotak p4 pogrešaka u odnosu na referentni koder. Općenito je vidljivo da je poboljšanje svojstava koje donosi GMM LSF VQ veće za modove koji za kodiranje ovojnice koriste niže brzine prijenosa. Primjerice, redukcija SD je najveća (0.292 dB) za modove koji koriste 23 bita/okviru za kodiranje ovojnice (5k15 i 4k75).

Tablica 6.4 *Usporedba učinkovitosti polaznog i modificiranog sustava (m×q verzija) koji koriste jednaku prijenosnu brzinu za kodiranje spektralne ovojnice.*

AMR môd (kbit/s)	***Referentni AMR koder***					***Modificirani AMR koder (m×q)***								
	bitovi / okviru	**PESQ**	**prosj. SD (dB)**	**pogreške (%)**		**bitovi / okviru**	**PESQ**		**prosj. SD (dB)**		**pogreške (%)**			
				2-4 dB	> 4 dB			Δ		Δ	2-4 dB	Δ	> 4 dB	Δ
12.2	38	4.002	0.983	1.864	0.072	38	4.012	0.010	0.762	0.221	1.220	0.644	0.042	0.029
7.95	27	3.721	1.074	2.164	0.058	27	3.709	-0.012	0.969	0.105	1.305	0.859	0.058	0
10.2	26	3.914	1.217	4.100	0.069	26	3.927	0.013	1.029	0.188	1.416	2.684	0.037	0.032
7.4	26	3.704				26	3.714	0.010						
6.7	26	3.614				26	3.619	0.005						
5.9	26	3.488				26	3.488	0						
5.15	23	3.365	1.552	15.656	0.106	23	3.365	0	1.260	0.292	5.007	10.650	0.164	-0.058
4.75	23	3.306				23	3.307	0.001						

Kako je i očekivano, uz jednake trendove kroz podržane modove rada, *2-best* verzija modificiranog AMR kodera, u usporedbi sa *m*×*q* verzijom, pokazuje blago inferiorne rezultate, prikazane u tablici 6.5. Smanjena kompleksnost, u odnosu na *m*×*q* verziju, rezultira u prosjeku 0.018 dB manjom PESQ ocjenom, te 0.107 dB većom prosječnom SD. Iako nešto manja, razlika u prosječnoj PESQ ocjeni u odnosu na referentni koder,

ovisno o môdu rada, kreće se u granicama od -0.031 do 0.002. S prosječnom razlikom od 0.015 za sve modove rada, u korist referentnog AMR kodera, može se također zaključiti da promjena u načinu kodiranja spektralne ovojnice nije bitno utjecala na PESQ ocjenu cjelokupnog kodera. Također, iako nešto lošije u usporedbi s $m \times q$ verzijom, vrijednosti SD ipak jasno nadmašuju svojstva referentnog AMR kodera (za 0.018 – 0.162 dB). Istovremeno, postotak p2 i p4 pogrešaka je nešto veći u usporedbi s referentnim AMR koderom, osim za modove s najmanjom prijenosnom brzinom, koji pokazuju znatno manji postotak p2 pogrešaka.

Tablica 6.5 *Usporedba učinkovitosti polaznog i modificiranog sustava (2-best verzija) koji koriste jednaku prijenosnu brzinu za kodiranje spektralne ovojnice.*

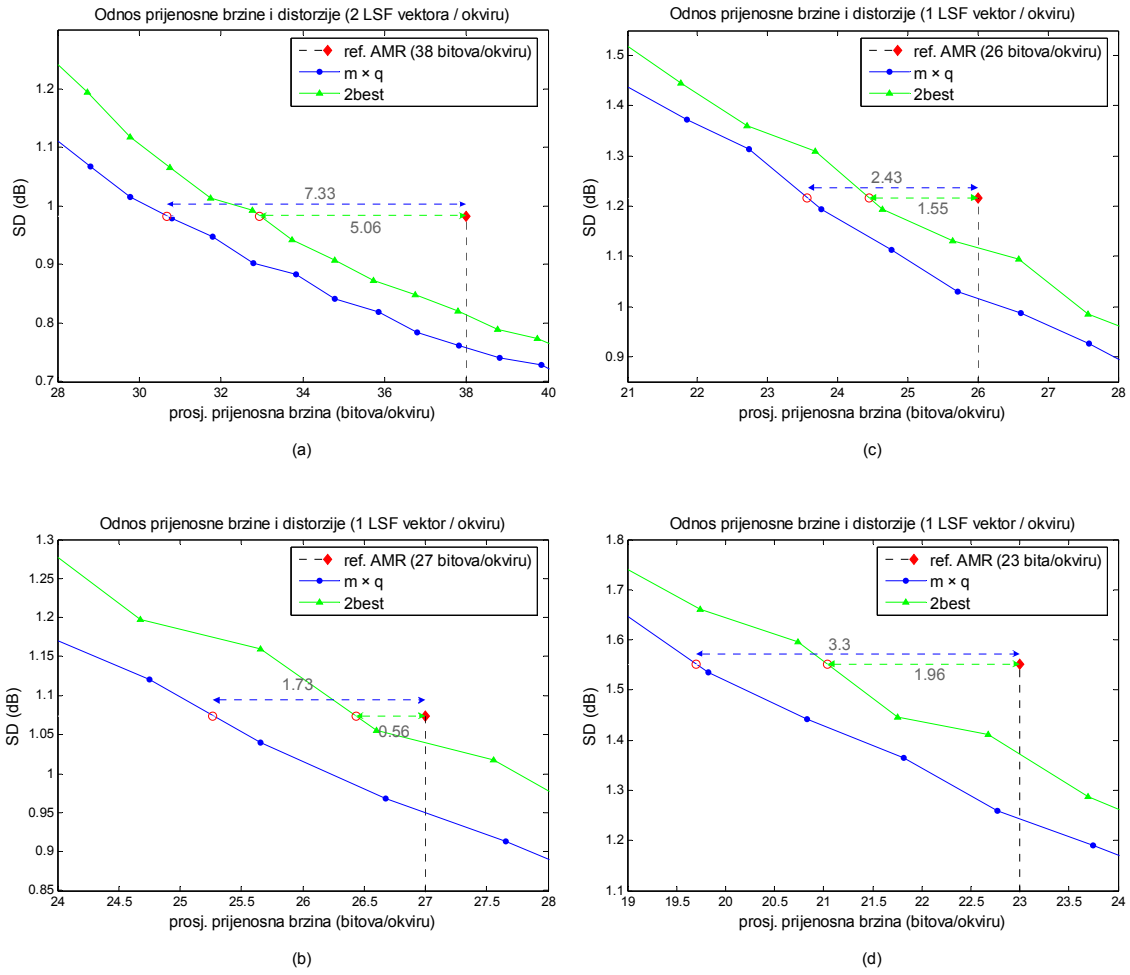
AMR môd (kbit/s)	***Referentni AMR koder***					***Modificirani AMR koder (2-best)***								
	bitovi / okviru	**PESQ**	**prosj. SD (dB)**	**pogreške (%)**		**bitovi / okviru**	**PESQ**		**prosj. SD (dB)**		**pogreške (%)**			
											2-4 dB		> 4 dB	
				2-4 dB	> 4 dB			Δ		Δ		Δ		Δ
12.2	38	4.002	0.983	1.864	0.072	38	3.989	-0.013	0.821	0.162	2.286	-0.422	0.106	-0.034
7.95	27	3.721	1.074	2.164	0.058	27	3.701	-0.020	1.055	0.018	3.447	-1.283	0.186	-0.127
10.2	26	3.914	1.217	4.100	0.069	26	3.916	0.002	1.130	0.087	5.060	-0.960	0.324	-0.255
7.4	26	3.704				26	3.691	-0.014						
6.7	26	3.614				26	3.583	-0.031						
5.9	26	3.488				26	3.466	-0.022						
5.15	23	3.365	1.552	15.656	0.106	23	3.349	-0.016	1.411	0.141	7.324	8.332	0.149	-0.042
4.75	23	3.306				23	3.295	-0.011						

Jasno je vidljivo da obje verzije predloženog algoritma spektralnu ovojnicu kodiraju točnije u usporedbi s referentnim AMR koderom. Međutim, sukladno ranije opisanim simulacijama, poboljšanje u kodiranju ovojnice nije vidljivo u PESQ ocjenama cjelokupnog kodera, što potvrđuje zaključak da su svojstva cjelokupnog kodera zapravo ograničena pogreškom kvantizacije pobude (algebarska i adaptivna kodna knjiga).

6.8. Usporedba referentnog i modificiranog kodera koristeći GMM VQ s reduciranom brzinom prijenosa

Kako učinkovitost cjelokupnog AMR kodera nije ograničena pogreškom kvantizacije spektralne ovojnice, već pogreškom kvantizacije pobude, točnije kodiranje ovojnice korištenjem jednake prijenosne brzine nema velikog smisla. Očito se dio raspoloživih bitova za kodiranje LSF parametara troši na poboljšanje parametara koji ne doprinose svojstvima cjelokupnog AMR kodera. Za provjeru ove pretpostavke, projektirani su GMM LSF VQ-ri koji, u usporedbi s referentnim AMR koderom, za kodiranje spektralne ovojnice koriste manju prijenosnu brzinu.

Mogućnost redukcije brzine prijenosa istražena je izračunom empirijskih krivulja odnosa prijenosne brzine i distorzije (engl. *rate-distortion*, RD). RD krivulje dobivene su variranjem željene fiksne prijenosne brzine kodirane spektralne ovojnice. Pri tome je, za svaku željenu brzinu izračunata odgovarajuća vrijednost spektralne distorzije, te prosječna PESQ ocjena. Za môd 12k2, koji dva LSF vektora iz istog okvira kodira kao jednu cjelinu, fiksna prijenosna brzina varirana je u rasponu od 28 – 42 bita/okviru, dok je za ostale modove korišten raspon od 16 – 30 bitova/okviru.



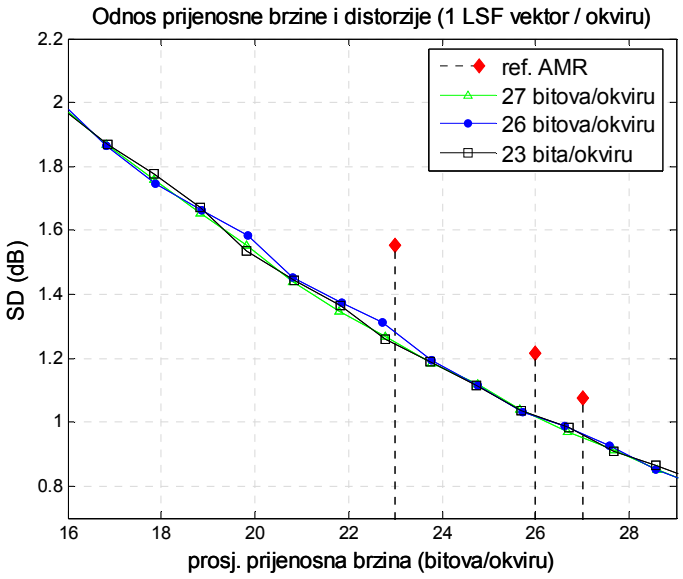
Slika 6.12 *Usporedba SD referentnog i modificiranog AMR kodera. Odnos prosječne prijenosne brzine i distorzije na pojedinim slikama prikazan je za GMM LSF VQ čiji je postupak projektiranja inicijaliziran rezidualima izdvojenim iz referentnog AMR kodera u odgovarajućem môdu rada.*

Empirijske RD krivulje za obje verzije modificiranog AMR kodera prikazane su na slikama 6.12(a)-(d). Slika 6.12(a) prikazuje RD krivulje koje se odnose na môd rada s najvećom prijenosnom brzinom (12k2). Izdvajanjem rezidualnih LSF vektora u 12k2 môdu referentnog AMR kodera (38 bitova/okviru za kodiranje ovojnice), za svaku

željenu fiksnu prijenosnu brzinu projektiran je odgovarajući GMM LSF VQ. Evaluacijom spektralne distorzije svih VQ-ra, dobiva se odgovarajuća RD krivulja. Vidljivo je da $m \times q$ verzija ima SD jednaku distorziji AMR kodera pri brzini prijenosa manjoj za 7.33 bita/okviru. Slične rezultate pokazuje i *2-best* verzija, koja jednaku prosječnu SD postiže sa prosječno 5.06 bitova/okviru manje od referentnog kodera.

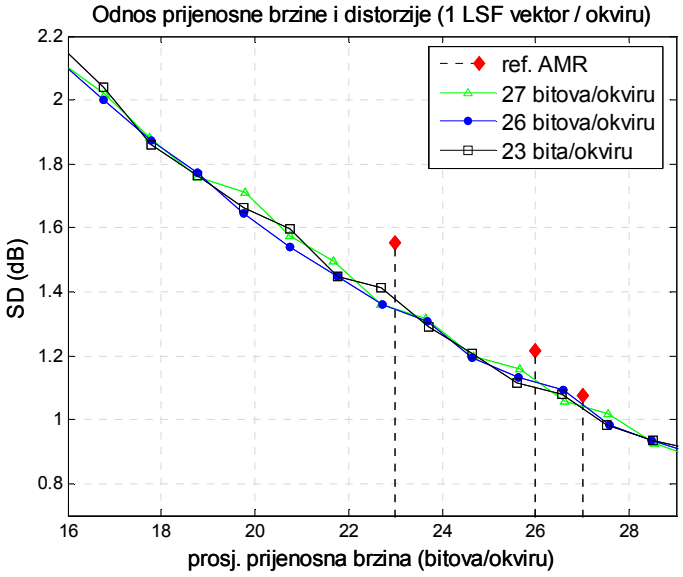
RD krivulje ostalih modova, koji kodiraju 1 LSF vektor po okviru, prikazane su na slikama 6.12(b)-(d). Slično gore opisanom postupku, i ove RD krivulje dobivene su evaluacijom spektralne distorzije GMM LSF VQ-ra, projektiranih za različite fiksne prijenosne brzine korištenjem reziduala izdvojenih u odgovarajućem môdu referentnog AMR kodera. Na slikama je vidljivo da, ovisno o môdu rada (prijenosnoj brzini), $m \times q$ verzija nadmašuje referentni koder za 1.73–3.3 bita/okviru, dok *2-best* verzija treba u prosjeku 0.56–1.96 bita/okviru manje od referentnog AMR kodera za istu prosječnu SD. Slično ranijem zapažanju kod usporedbe koderâ s jednakom prijenosnom brzinom, vidljivo je da se veće redukcije prijenosne brzine postižu za AMR modove s nižim prijenosnim brzinama.

RD krivulje na slikama 6.12(b)-(d) predstavljaju simulacijske rezultate GMM LSF vektorskih kvantizatora, projektiranih istovjetnim postupkom, korištenjem iste trening baze LSF vektora. Naime, bez obzira na odabrani môd rada (koji kodira jedan LSF vektor po okviru), referentni AMR koder izračunava i kvantizira iste LSF vektore, koji se onda prilikom projektiranja GMM VQ izdvajaju i upotrebljavaju za izračun reziduala postupkom MA predikcije. Međutim, izdvojeni LSF reziduali, čija se funkcija gustoće vjerojatnosti modelira u prvoj iteraciji postupka projektiranja, razlikovat će se u ovisnosti o odabranoj prijenosnoj brzini (môdu) referentnog kodera. To je posljedica činjenice da SVQ pojedine pod-vektore različito kodira u ovisnosti o prijenosnoj brzini odabranog môda, koristeći različite kvantizacijske razine (kodne tablice) i alokacije raspoloživih bitova (tablica 3.2). Posljedica razlike u inicijalnoj bazi izdvojenih LSF reziduala je blago rasipanje RD krivulja vektorskih kvantizatora za različite prijenosne brzine spektralne ovojnice. Slike 6.13 i 6.14 skupno prikazuju RD krivulje obje verzije GMM LSF vektorskih kvantizatora, čiji je postupak projektiranja inicijaliziran rezidualima izdvojenim iz referentnog AMR kodera u različitim modovima rada (27, 26 i 23 bita/okviru). Iz izloženog je jasno da su RD krivulje različitih modova i iste prijenosne brzine (npr. 5k15 i 4k75) identične.

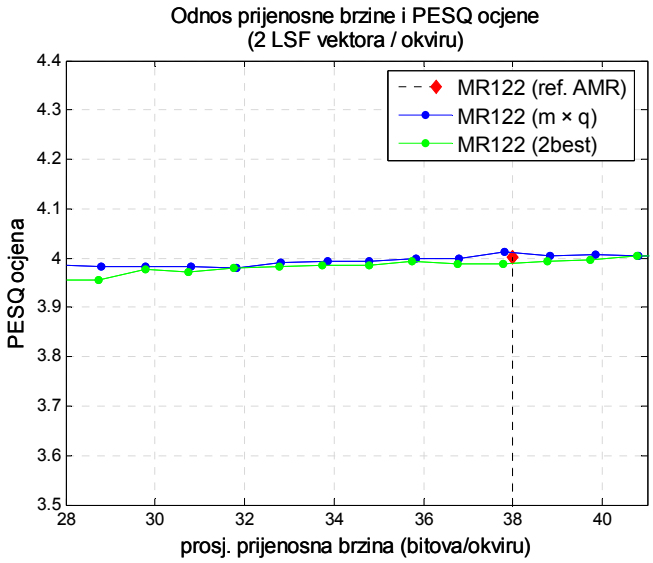


Slika 6.13 *Usporedba RD krivulja različito inicijaliziranih m×q verzija modificiranog AMR kodera pri kodiranju jednog LSF vektora/okviru*

.



Slika 6.14 *Usporedba RD krivulja različito inicijaliziranih 2-best verzija modificiranog AMR kodera pri kodiranju jednog LSF vektora/okviru.*

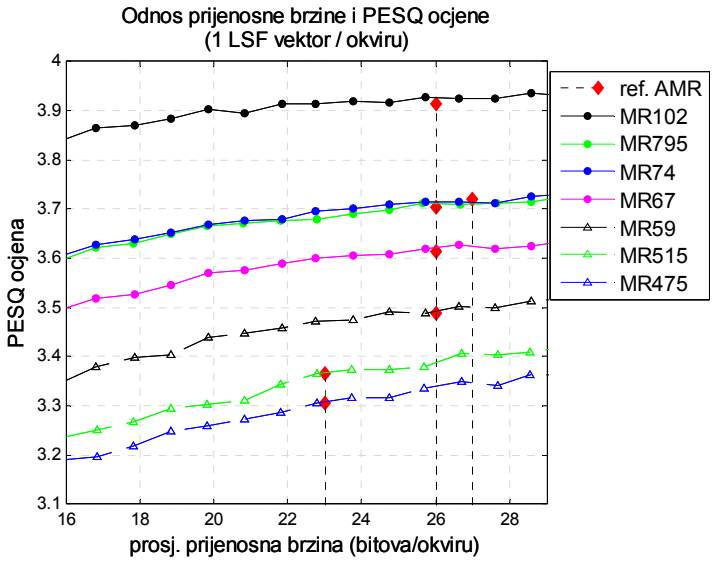


Slika 6.15 *Usporedba PESQ ocjene referentnog i modificiranih verzija AMR kodera (môd 12k2).*

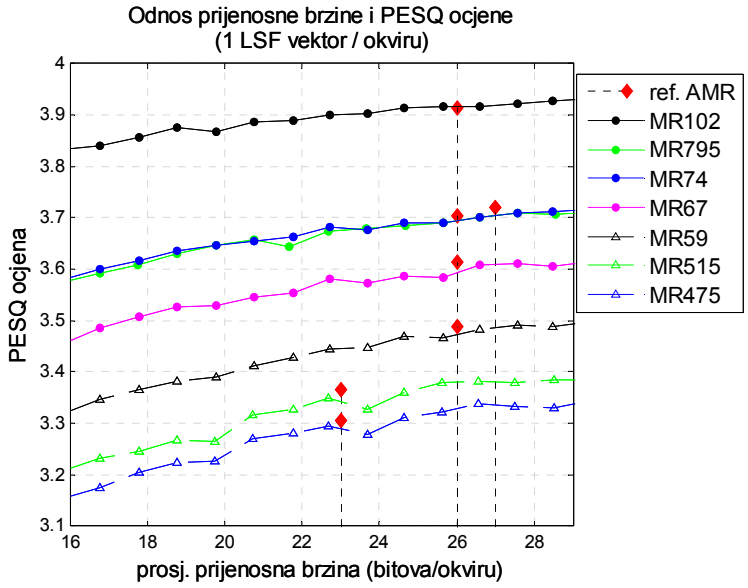
Kako je i pretpostavljeno u prethodnom poglavlju, redukcija prijenosne brzine kodirane ovojnice ne utječe bitno na PESQ ocjenu cjelokupnog kodera kod obje modificirane verzije. Odnos prijenosne brzine i PESQ ocjene za obje verzije modificiranog kodera u môdu 12k2 prikazan je na slici 6.15. Prikazane krivulje pokazuju da učinkovitost cjelokupnog kodera gotovo ne ovisi o pogrešci kvantizacije spektralne ovojnice u testiranom rasponu prijenosnih brzina. Kod redukcije prijenosne brzine $m \times q$ verzije modificiranog kodera u 12k2 môdu za 7 bitova, kao i kod redukcije za 5 bita kod *2-best* verzije, PESQ ocjena je u prosjeku za -0.018 manja u usporedbi s referentnim koderom.

Kod ostalih modova, čije su krivulje odnosa prijenosne brzine i PESQ ocjene prikazane na slici 6.16 za $m \times q$ verziju, odnosno slici 6.17 za *2-best* verziju modificiranog kodera, u usporedbi s 12k2 môdom, u testiranom rasponu prijenosnih brzina vidljiva je nešto izraženija ovisnost učinkovitosti cjelokupnog kodera o prijenosnoj brzini kodirane spektralne ovojnice. U ovom slučaju, $m \times q$ verzije ekvivalentnih (po spektralnoj distorziji) GMM vektorskih kvantizatora postižu do -0.061 manje PESQ ocjene u odnosu na referentni koder standardizirane brzine, dok ta razlika za *2-best* verzije raste do -0.098. Navedene razlike potvrđuju pretpostavku o mogućnosti redukcije prijenosne brzine s početka ovog poglavlja.

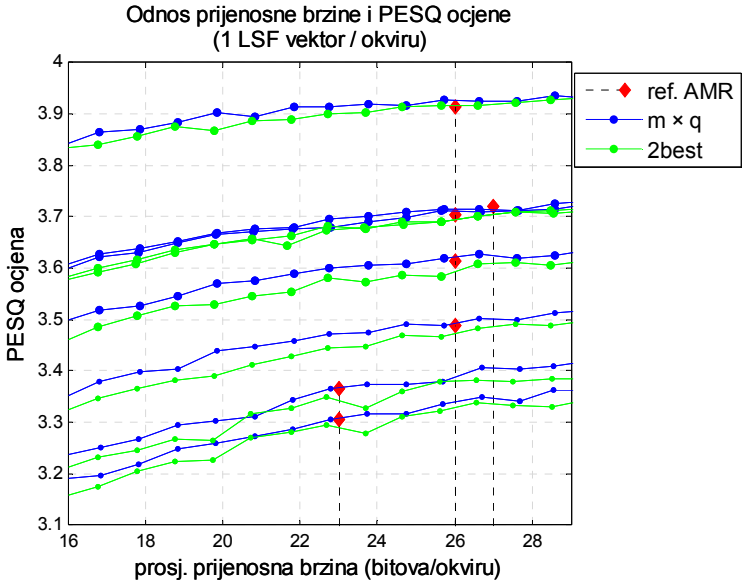
Radi usporedbe i kompletnijeg prikaza simulacijskih rezultata, krivulje odnosa prijenosne brzine i PESQ ocjene za obje verzije modificiranog AMR kodera prikazane su skupa na slici 6.18. Iz skupnog prikaza kvalitativno je vidljivo pogoršanje cjelokupne učinkovitosti kodera koje, u odnosu na $m \times q$ verziju, unosi redukcija računske kompleksnosti u *2-best* verziji modificiranog AMR kodera.



Slika 6.16 *Usporedba PESQ ocjene referentnog i m×q verzije modificiranog AMR kodera (svi modovi osim 12k2).*



Slika 6.17 *Usporedba PESQ ocjene referentnog i 2-best verzije modificiranog AMR kodera (svi modovi osim 12k2).*



Slika 6.18 *Preklopljene m×q i 2-best PESQ krivulje za modove s jednim LSF vektorom/okviru.*

6.9. Računska složenost i memorijski zahtjevi

Osnovni cilj analize računske složenosti i memorijskih zahtjeva obje verzije predloženog LSF VQ temeljenog na GMM-u je njihova usporedba sa složenošću i zahtjevima SMQ i SVQ algoritama koje koristi referentni AMR koder. Iz tog razloga, analiza računske složenosti ne obuhvaća složenost MA predikcije, jer je na istovjetan način koristi i modificirani i polazni sustav. Računska složenost pojedinog kvantizatora izražena je brojem operacija s pomičnim zarezom (engl. *floating point operations,* flops) potrebnih za kodiranje reziduala u jednom okviru, flops/okviru, gdje se svako zbrajanje, množenje, dijeljenje i uspoređivanje ubraja kao jedan flop.

Memorijski zahtjevi izražavaju se jedinicom „float“, koja predstavlja memoriju potrebnu za pohranu bilo kojeg realnog broja.

6.9.1. Računska složenost i memorijski zahtjevi SVQ

Korištenjem težinske mjere LSP distorzije prema izrazu 3.6, za kvantizaciju *d*-dimenzionalnog vektora SVQ algoritmom potrebno je

$$kompleksnost_{SVQ} = \sum_{i=1}^{s} (4 d_i 2^{b_i} - 1) \quad flops\,, \tag{6.1}$$

gdje s predstavlja broj pod-vektora na koje je razlomljen vektor koji se kvantizira, d_i predstavlja dimenziju i-tog pod-vektora, a b_i odgovara broju bitova alociranih za njegovo kodiranje. Pri tome vrijedi

$$d = \sum_{i=1}^{s} d_i\,, \tag{6.2}$$

$$b_{tot} = \sum_{i=1}^{s} b_i \,, \qquad (6.3)$$

gdje b_{tot} odgovara broju bitova raspoloživom za kodiranje LSF parametara jednog okvira u odgovarajućem môdu AMR kodera. Alokacija bitova za kodiranje spektralne ovojnice po pojedinim modovima AMR kodera, skupa s njihovom raspodjelom po odgovarajućim pod-vektorima, prikazana je u tablici 3.2. Iz izraza 6.1 vidljivo je da je računska kompleksnost SVQ jednaka zbroju kompleksnosti kvantizatora koji kvantiziraju pojedine pod-vektore. Kompleksnost svakog od pod-vektorskih kvantizatora određena je umnoškom broja reprezentanata odgovarajuće kodne knjige (2^{b_i}) i broja operacija potrebnih za evaluaciju pogreške komponenti pojedinog reprezentanta, te njenu usporedbu s trenutno najboljim kandidatom ($4d_i$). Pri tome je, prema izrazu 3.6, za svaku komponentu pod-vektora potrebno 4 operacije: jedna operacija oduzimanja, dvije operacije množenja i jedna operacija zbrajanja koja težinsku pogrešku komponente pribraja akumuliranoj pogrešci ostalih komponenti. Prva komponenta ne treba operaciju zbrajanja, ali cjelokupan pod-vektor treba dodatnu operaciju uspoređivanja za usporedbu s trenutno najboljim kandidatom. Na kraju, radi se korekcija za -1, za svaki pod-vektor, jer pogreška za prvi reprezentant ne treba usporedbu.

Memorijski zahtjevi SVQ općenito su dani izrazom

$$memorija_{SVQ} = \sum_{i=1}^{s} d_i 2^{b_i} \quad floats \,, \qquad (6.4)$$

gdje s, d_i i b_i odgovaraju prethodno navedenim značenjima kod izračuna računske kompleksnosti. Vidljivo je da memorijski zahtjevi SVQ odgovaraju veličini memorije potrebnoj za pohranu s kodnih knjiga, gdje svaka kodna knjiga odgovara kvantizatoru određenog pod-vektora. Veličina memorije potrebna za pojedinu kodnu knjigu određena je rezolucijom (b_i) i dimenzijom pod-vektora (d_i) odgovarajućeg kvantizatora.

Vidljivo je da i računska kompleksnost i memorijski zahtjevi SVQ eksponencijalno rastu s rezolucijom kvantizatora, te linearno ovise o dimenzionalnosti procesa koji se kvantizira.

6.9.2. Računska složenost i memorijski zahtjevi SMQ

Slično SVQ, uz korištenje iste težinske mjere LSP distorzije prema izrazu 3.6, SMQ algoritam za skupnu kvantizaciju n d-dimenzionalnih vektora koji zajedno čine matricu dimenzija $d \times n$ treba:

$$kompleksnost_{SMQ} = \sum_{i=1}^{s} (4 d_i n_i 2^{b_i} - 1) \quad flops \,, \qquad (6.5)$$

gdje je s predstavlja broj pod-matrica na koje se dijeli matrica susjednih vektora koji se skupno kvantiziraju, $d_i \times n_i$ predstavlja dimenzije i-te pod-matrice, a b_i odgovara broju bitova alociranih za njeno kodiranje. Pri tome vrijedi

$$d = \sum_{i=1}^{s} d_i; \quad n = \sum_{i=1}^{s} n_i , \tag{6.6}$$

$$b_{tot} = \sum_{i=1}^{s} b_i , \tag{6.7}$$

gdje b_{tot} odgovara raspoloživom broju bitova za kodiranje LSF parametara jednog okvira u 12k2 môdu AMR kodera. Alocirani broj bitova za kodiranje spektralne ovojnice u 12k2 môdu AMR kodera, skupa s njihovom raspodjelom po odgovarajućim pod-matricama, opisana je u poglavlju 3.2. Izraz 6.5 izveden je sličnom argumentacijom kao i izraz 6.1, od kojeg se razlikuje samo u broju elemenata pod-matrice ($d_i n_i$), odnosno komponenti pod-vektora (d_i), koji se istovremeno kvantiziraju odgovarajućim kvantizatorom.

Memorijski zahtjevi SMQ također se izračunavaju slično zahtjevima SVQ, zbrajajući memoriju potrebnu za kodne knjige svih kvantizatora pojedinih pod-matrica. Ako se broj komponenti pod-vektora (d_i) u SVQ zamijeni brojem elemenata pod-matrice ($d_i n_i$) u SMQ, dobiva se izraz za memorijske zahtjeve SMQ

$$memorija_{SMQ} = \sum_{i=1}^{s} d_i n_i 2^{b_i} \quad floats , \tag{6.8}$$

gdje s, d_i, n_i i b_i odgovaraju prethodno navedenim značenjima kod izračuna kompleksnosti.

Kao i kod SVQ, računska kompleksnost i memorijski zahtjevi SMQ također eksponencijalno rastu s rezolucijom kvantizatora, te linearno ovise o dimenzionalnosti procesa i broju vektora koji se skupno kvantiziraju.

6.9.3. Računska složenost LSF VQ temeljenog na GMM-u

Računska složenost GMM LSF VQ na strani kodera, odnosno broj operacija s pomičnim zarezom potrebnih za kodiranje spektralne ovojnice jednog okvira, analizirana je u tablici 6.6. U tablici je prikazana i komentirana računska složenost svih operacija koje obuhvaća postupak kodiranja LSF reziduala. Analiza obuhvaća obje verzije *pd*-dimenzionalnog GMM LSF VQ s *m* komponenti mješavine, gdje *p* predstavlja broj *d*-dimenzionalnih LSF reziduala koji se skupno kodiraju. Tako je za môd 12k2 $p = 2$, dok je za sve ostale modove $p = 1$. Varijacija koraka kvantizacije, u svrhu prilagodbe entropijski kodiranih reziduala na fiksnu prijenosnu brzinu AMR kodera, ostvaruje se primjenom *q* ECSQ kvantizatora.

Iz rezultata analize vidljivo je jedno od značajnijih svojstava GMM VQ: ukupna računska složenost GMM VQ, kako je već spomenuto u [28], ne ovisi o prijenosnoj brzini, odnosno rezoluciji kodirane spektralne ovojnice. To je u potpunoj suprotnosti s matematičkom složenošću vektorskih kvantizatora s potpunom pretragom kodne knjige, te vektorskih kvantizatora s ograničenom strukturom (SMQ, SVQ), čija kodna knjiga, a zbog toga i memorijski zahtjevi i računska složenost, eksponencijalno raste s

povećanjem rezolucije kvantizatora [13]. Iz tablice je općenito vidljivo da ukupna računska složenost linearno ovisi o broju komponenti mješavine, te raste s kvadratom dimenzije (pd) GMM VQ-a. Računska složenost $m\times q$ i *2-best* verzije razlikuje se za $(m/2)(q[2(pd)^2+8pd-p])$ flops-a, umanjena za $2mpd-1$ flops-a potrebnih za odabir dvije najbolje komponente mješavine u *2-best* verziji GMM VQ. Složenost postupka kodiranja ECSQ kvantizatorima posebno je analizirana u tablici 6.7, iz koje je vidljiva njena linearna ovisnost o broju koraka kvantizacije (q) i dimenzionalnosti ECSQ kvantizatora (pd).

Tablica 6.6 *Računska složenost obje verzije GMM LSF VQ na koderskoj strani.*

operacija	***računska složenost (flops)***	
	m×q verzija	***2-best verzija***
oduzimanje vektora srednje vrijednosti – translacija reziduala u obrnutom smjeru odgovarajućeg centroida za svaku komponentu mješavine	mpd	mpd
dekorelacija transformacijskom matricom – matrično množenje reziduala s transformacijskom matricom svake komponente mješavine	$m[2(pd)^2-pd]$	$m[2(pd)^2-pd]$
odabir dvije najbolje komponente mješavine – primjenom TCG principa: izračun umnoška apsolutnih vrijednosti komponenti za m transformiranih reziduala ($m(2pd-1)$), te odabir dva najmanja umnoška ($m-1$ usporedba)	-	$2mpd-1$
entropijsko kodiranje – obuhvaća kvantizaciju, Huffmanovo kodiranje i dekvantizaciju svakog transformiranog kandidata (tablica 6.7)	$mq(4pd-1)$	$2q(4pd-1)$
koreliranje inverznom transformacijskom matricom – inverzna transformacija dekodiranih i dekvantiziranih kandidata	$mq[2(pd)^2-pd]$	$2q[2(pd)^2-pd]$
dodavanje vektora srednje vrijednosti – translacija u smjeru centroida odgovarajuće komponente mješavine	$mqpd$	$2qpd$
izračun težinske mjere LSP distorzije – za svakog inverzno procesiranog kandidata	$mqp(4d-1)$	$2qp(4d-1)$
usporedba pogreški svih kandidata – odabir kandidata (indeksa komponente mješavine i ECSQ kvantizatora) koji ulazni rezidual reprezentira uz najmanju distorziju	$mq-1$	$2q-1$
Ukupna računska složenost:	$\mathbf{2m(pd)^2+mq[2(pd)^2+8pd-p]-1}$	$\mathbf{2m[(pd)^2+pd]-1+2q[2(pd)^2+8pd-p]-1}$

Tablica 6.7 *Računska složenost postupka kodiranja ECSQ kvantizatorima.*

operacija	***računska složenost (flops)***	
	m×q verzija	***2-best verzija***
kvantizacija transformiranog reziduala – dijeljenjem njegovih komponenti s korakom kvantizacije svakog od q ECSQ kvantizatora	$mqpd$	$2qpd$
kodiranje kvantiziranih komponenti reziduala – zamjenom izlaznog indeksa kvantizatora s odgovarajućim entropijskim kôdom iz odgovarajuće Huffmanove tablice (engl. *table lookup operation*)	$mqpd$	$2qpd$
zbrajanje duljine entropijskih kodova svih komponenti kodiranog reziduala – radi provjere duljine kodiranog kandidata	$mq(pd-1)$	$2q(pd-1)$
rekonstrukcija transformiranog reziduala – množenjem izlaznih indeksa kvantizatora s odgovarajućim korakom kvantizacije	$mqpd$	$2qpd$
Ukupna računska složenost:	$\boldsymbol{mq(4pd-1)}$	$\boldsymbol{2q(4pd-1)}$

Tablica 6.8 *Računska složenost GMM LSF VQ na dekoderskoj strani.*

operacija	***računska složenost (flops)***
dekodiranje entropijskog kôda komponenti kodiranog reziduala – obuhvaća dekodiranje Huffmanovog kôda svih komponenti kodiranog reziduala (prosječno h_c operacija po komponenti), te njihovu dekvantizaciju množenjem s odgovarajućim korakom kvantizacije	$pd(h_c+1)$
dekodiranje entropijskog kôda odabrane kombinacije indeksa najbolje komponente mješavine i indeksa najboljeg ECSQ kvantizatora – prosječno h_i operacija za dekodiranje Huffmanovog kôda kombiniranog indeksa	h_i
koreliranje odgovarajućom inverznom transformacijskom matricom – inverzna transformacija dekodiranog i dekvantiziranog reziduala	$2(pd)^2-pd$
dodavanje odgovarajućeg vektora srednje vrijednosti – translacija u smjeru centroida odgovarajuće komponente mješavine	pd
Ukupna računska složenost:	$\boldsymbol{2(pd)^2+(h_c+1)pd+h_i}$

Složenost GMM LSF VQ na strani dekodera, naravno, ne ovisi o varijanti kvantizatora. Broj operacija potreban za dekodiranje entropijski kodiranih komponenti reziduala, njihovo množenje odgovarajućim korakom kvantizacije, inverznu transformaciju i translaciju parametrima odgovarajuće komponente mješavine analiziran je u tablici 6.8. Vidljivo je da složenost dekodiranja raste s kvadratom dimenzije kodiranog vektora (pd), što je posljedica inverzne transformacije, te linearno raste s prosječnim brojem operacija potrebnih za dekodiranje jedne komponente kodiranog vektora (h_c).

6.9.4. Memorijski zahtjevi LSF VQ temeljenog na GMM-u

Memorijski zahtjevi GMM LSF VQ općenito su neovisni o njegovoj verziji. Tablica 6.9 prikazuje analizu memorijskih zahtjeva na strani kodera. Kao i kod analize računske složenosti, vidljivo je da memorijski zahtjevi ne ovise o rezoluciji GMM LSF VQ-a [28]. Ovo vrijedi pod pretpostavkom neovisnosti veličine Huffmanovih tablica o prijenosnoj brzini kodera, što naravno nije točno. Veća prijenosna brzina kodera omogućava primjenu finijih koraka kvantizacije, što rezultira Huffmanovim tablicama s većim brojem simbola. Međutim, zbog jednostavnosti, memorijski zahtjevi za spremanje Huffmanovih tablica izračunati su umnoškom ukupnog broja tablica (pdq) s prosječnim brojem simbola po tablici (h_s), empirijski dobiven prebrojavanjem simbola u Huffmanovim tablicama projektiranih GMM-temeljenih vektorskih kvantizatora, čije su prijenosne brzine u pojedinim modovima jednake prijenosnim brzinama referentnog AMR kodera. Prosječan broj simbola po tablici posebno je određen za môd sa $p = 2$, te posebno za modove s $p = 1$. Pri tome je na strani kodera po svakom simbolu potrebno spremiti tri vrijednosti: ulazni kôd simbola, pripadajući izlazni kôd, te njegova varijabilna duljina.

Tablica 6.9 *Memorijski zahtjevi obje verzije GMM LSF VQ na koderskoj strani.*

parametri	***memorijski zahtjevi (floats)***
vektor srednjih vrijednosti (μ_i) – duljine pd, za svaku od m komponenti Gaussove mješavine	mpd
transformacijska matrica ($\mathbf{T}_i$) – dimenzija $pd \times pd$, za svaku od m komponenti Gaussove mješavine	$m(pd)^2$
Huffmanove tablice za kodiranje kvantiziranih komponenti transformiranih reziduala – $pd \times q$ Huffmanovih tablica, za svaki od q ECSQ kvantizatora po jedna tablica za svaku od d komponenti transformiranog reziduala – ulazni kôd simbola, te pripadajući izlazni kôd i njegova varijabilna duljina za svaki od prosječno h_s simbola po tablici	$3h_sqpd$
Huffmanova tablica za entropijsko kodiranje odabrane kombinacije komponente mješavine i ECSQ kvantizatora – ulazni kôd simbola, te pripadajući izlazni kôd i njegova varijabilna duljina za svaku od mq kombinacija	$3mq$
korak kvantizacije – za svaki od q ECSQ kvantizatora	q
Ukupni memorijski zahtjevi:	$\boldsymbol{m[(pd)^2+pd]+3q(m+h_spd)+q}$

Iz tablice 6.9 također je vidljivo da memorijski zahtjevi GMM LSF VQ, slično računskoj složenosti, linearno ovise o broju komponenti mješavine (m) i rastu s kvadratom dimenzije GMM VQ-a (pd). Osim navedenog, memorijski zahtjevi GMM VQ linearno se povećaju s brojem ECSQ kvantizatora (q) i, naravno, s povećanjem prosječnog broja simbola u Huffmanovim tablicama.

Sve navedeno u analizi memorijskih zahtjeva GMM LSF VQ na strani kodera, također vrijedi i za dekodersku stranu. Izuzetak predstavlja jedino memorija potrebna za pohranu Huffmanovih tablica. Naime, za razliku od kodera, u dekoderu je osim parametara za listove Huffmanovih tablica potrebno pohraniti i parametre međučvorova, uključujući i korijenski čvor. Za svaki simbol, potrebno je pohraniti vrijednost simbola, te indekse oca i sinova, dok je za međučvorove potrebno pohraniti samo spomenute indekse. Drugim riječima, umjesto tri vrijednosti po simbolu, u dekoderu je potrebno pohraniti četiri vrijednosti po svakom simbolu, te po tri vrijednosti za svaki međučvor. U konačnici, određivanjem prosječnog broja međučvorova po tablici (h_m), kojih je u pravilu za jedan manje od prosječnog broja simbola (h_s), izraz za memoriju (tablica 6.9) potrebnu za pohranu Huffmanovih tablica na dekoderskoj strani mijenja se u

$$(4h_s + 3h_m)qpd = \left[4h_s + 3(h_s - 1)\right]qpd = (7h_s - 3)qpd\,, \qquad (6.9)$$

umjesto $3h_s qpd$ na strani kodera.

6.9.5. Usporedba računske složenosti polaznog i modificiranog sustava

Za usporedbu matričnih/vektorskih kvantizatora iz referentnog AMR kodera (SMQ, SVQ) sa predloženim verzijama GMM LSF VQ, izrazi 6.1 i 6.5, te izrazi u tablicama 6.6 i 6.8 upotrijebljeni su za izračun konkretnih vrijednosti za računsku složenost odgovarajućih LSF kvantizatora. Rezultati izračuna za kodersku i dekodersku stranu prikazani su u tablicama 6.10 i 6.11. Vrijednosti za SMQ i SVQ izračunate su samo za odgovarajuće modove AMR kodera koji koriste spomenute kvantizatore.

Tablica 6.10 *Usporedba računske složenosti polaznog i modificiranog sustava na koderskoj strani.*

brzina (bitovi/okviru) / AMR môd	***p***	***računska složenost (flops)***			
		SMQ	***SVQ***	***m×q verzija***	***2-best verzija***
38 / 12k2	2	19451	-	37055	14382
27 / 7k95	1	-	20477	10527	3990
26 / 10k2, 7k40, 6k70, 5k90	1	-	17405		
23 / 5k15, 4k75	1	-	8189		

Iz tablice 6.10, na strani kodera vidljiva je redukcija računske složenosti *2-best* verzije za faktor 2.6, u usporedbi sa *m*×*q* verzijom. Također je vidljivo da je računska složenost *m*×*q* verzije za modove koji kodiraju jedan LSF vektor po okviru ($p = 1$) usporediva sa složenošću SVQ za modove s najmanjom rezolucijom (23 bita/okviru), dok u odnosu na môd s najvećom rezolucijom (27 bitova/okviru) ostvaruje redukciju složenosti za faktor 2. Naravno, taj se faktor za *2-best* verziju množi sa navedenim faktorom redukcije 2.6, što u konačnici daje redukciju složenosti *2-best* verzije za faktor 5.2 u odnosu na složenost SVQ u 7k95 môdu. Odnos složenosti nešto je drugačiji za môd 12k2 ($p = 2$),

gdje je računska složenost $m \times q$ verzije za faktor 1.9 veća od složenosti SMQ, što je posljedica činjenice da složenost GMM VQ raste proporcionalno s kvadratom dimenzionalnosti kvantizatora. Kako spomenuti môd skupno kodira $p = 2$ vektora po okviru, dvostruko veća dimenzionalnost povećava složenost za faktor 4 u odnosu na modove sa $p = 1$. I u ovom slučaju, množenjem sa faktorom 2.6 dobiva se faktor redukcije složenosti *2-best* verzije, čija je složenost 1.3 puta manja u odnosu na računsku složenost SMQ algoritma.

Konkretne vrijednosti računske složenosti na strani dekodera prikazane su u tablici 6.11. U usporedbi s trivijalnom kompleksnošću dekodiranja vektora kodiranih SMQ/SVQ metodom, koje se svodi na jednostavno čitanje indeksiranih komponenti pod-matrica/pod-vektora iz odgovarajućih tablica reprezentanata, dekodiranje vektora kodiranih GMM LSF VQ-om je znatno složenije. Za dekodiranje vektora kodiranog u môdu s najvećom prijenosnom brzinom potrebno je 930 flops/okviru, dok je za ostale modove, zahvaljujući dva puta manjoj dimenzionalnosti kvantizatora, za isti posao potrebno skoro četverostruko manje operacija. Pri izračunu ovih vrijednosti prema izrazima u tablici 6.8, korištene su empirijski određene vrijednosti za prosječan broj operacija potrebnih za dekodiranje Huffmanovog kôda jedne komponente kodiranog vektora ($h_c = 5$), odnosno prosječan broj operacija potrebnih za dekodiranje kombiniranog indeksa najbolje komponente mješavine i ECSQ kvantizatora ($h_i = 10$). Navedene empirijske vrijednosti dobivene su prebrojavanjem operacija potrebnih za dekodiranje svih simbola u svim Huffmanovim tablicama GMM LSF VQ. Množenje prebrojanih operacija sa vjerojatnošću odgovarajućeg simbola, te njihovo zbrajanje po odgovarajućim tablicama rezultira prosječnim brojem operacija po svakoj komponenti vektora. Njihovim usrednjavanjem po svim komponentama, za sva četiri ECSQ kvantizatora, dobiva se gore navedena prosječna vrijednost po komponenti kodiranog vektora. Na sličan način dobivena je i vrijednost h_i. Iako modificirana verzija znatno povećava složenost dekodiranja, to povećanje posve je zanemarivo u usporedbi s kompleksnošću cijelog AMR dekodera.

Tablica 6.11 *Računska složenost modificiranog sustava na dekoderskoj strani.*

brzina (bitovi/okviru) / AMR môd	***p***	***računska složenost (flops)***	
		m×q verzija	***2-best verzija***
38 / 12k2	2	930	
27 / 7k95	1	270	
26 / 10k2, 7k40, 6k70, 5k90	1		
23 / 5k15, 4k75	1		

6.9.6. Usporedba memorijskih zahtjeva polaznog i modificiranog sustava

Slično usporedbi računske složenosti, za usporedbu memorijskih zahtjeva odgovarajućih kvantizatora polaznog i modificiranog sustava korišteni su izrazi 6.4, 6.8 i 6.9, te izrazi u tablici 6.9. Rezultati izračuna prikazani su u tablici 6.12 za obje strane odgovarajućih kvantizatora. Vrijednosti za SMQ i SVQ izračunate su samo za odgovarajuće modove AMR kodera koji koriste spomenute kvantizatore.

Tablica 6.12 *Usporedba memorijskih zahtjeva polaznog i modificiranog sustava.*

brzina (bitovi/okviru) / AMR môd	***p***	***memorijski zahtjevi (floats)***					
		koder/dekoder		***koder***		***dekoder***	
		SMQ	***SVQ***	***m×q verzija***	***2-best verzija***	***m×q verzija***	***2-best verzija***
38 / 12k2	2	3840	-	7780		13300	
27 / 7k95	1	-	5120	2660		4780	
26 / 10k2, 7k40, 6k70, 5k90	1	-	4352				
23 / 5k15, 4k75	1	-	2048				

Iz tablice 6.12 vidljiva je neovisnost memorijskih zahtjeva GMM-temeljenog VQ o verziji i rezoluciji kvantizatora. Pod pretpostavkom neovisnosti memorijskih zahtjeva potrebnih za spremanje Huffmanovih tablica o brzini prijenosa, za izračun memorijskih zahtjeva za modove s $p = 1$ korišten je empirijski određen prosječan broj od 14 simbola po tablici, dok za môd sa $p = 2$ taj broj iznosi 18. Vidi se da linearna ovisnost o kvadratu dimenzije VQ-a, skupa s povećanim prosječnim brojem simbola po tablici, otprilike trostruko povećava memorijske zahtjeve môda koji kodira dva LSF vektora po okviru. Na strani kodera, slično računskoj složenosti $m \times q$ verzije, memorijski zahtjevi modova koji kodiraju jedan LSF vektor po okviru usporedivi su sa zahtjevima SVQ za modove s najmanjom rezolucijom (23 bita/okviru), dok u odnosu na môd s najvećom rezolucijom (27 bitova/okviru) ostvaruju redukciju memorijskih zahtjeva za faktor 1.9. Međutim, brojnost Huffmanovih tablica, zbog dvostruke dimenzionalnosti kvantizatora u môdu 12k2, povećava memorijske zahtjeve GMM-temeljenog VQ na strani kodera za faktor 2 u odnosu na zahtjeve SMQ.

Kako memorija potrebna za pohranu Huffmanovih tablica predstavlja oko 60% ukupnih memorijskih zahtjeva na strani kodera, porast memorijskih zahtjeva na strani dekodera povećava njihov udio na oko 80% ukupno potrebne memorije. Tako su memorijski zahtjevi GMM LSF VQ na strani dekodera usporedivi s memorijskim zahtjevima SVQ, dok u môdu 12k2 trostruko premašuju memorijske zahtjeve SMQ.

Međutim, uzimajući u obzir činjenicu da Huffmanove tablice sadrže samo cjelobrojne brojeve, koji uglavnom predstavljaju simbole i indekse odgovarajućih redaka, njihovo

spremanje je u većini slučajeva moguće napraviti korištenjem kraćih memorijskih jedinica. Npr., ako pretpostavimo spremanje vektorskih komponenti SMQ/SVQ kodnih knjiga u 32-bitne lokacije (float), spremanjem elemenata Huffmanovih tablica u 8-bitne lokacije ostvaruje se približno dvostruka redukcija gore spomenutih omjera memorijskih zahtjeva u korist modificiranog sustava.

Nadalje, kako je prosječan broj simbola po Huffmanovoj tablici empirijski dobiven prebrojavanjem simbola u tablicama projektiranih GMM LSF vektorskih kvantizatora čije su prijenosne brzine u pojedinim modovima jednake prijenosnim brzinama referentnog AMR kodera, redukcija prijenosne brzine GMM LSF VQ pri zadržanim svojstvima standardiziranog kodera će, smanjenjem prosječnog broja simbola po tablici, dodatno reducirati memorijske zahtjeve predloženog kvantizatora.

7. ZAKLJUČAK

U ovom radu sustavno je uveden i opisan postupak vektorske kvantizacije LSF parametara koji se temelji modelima s Gaussovim mješavinama. Istražena je mogućnost primjene i doprinos takvog postupka kvantizacije u jednom od standardiziranih kodera govornog signala poput AMR kodera, isključivo uz izmjenu dijela algoritma koji se odnosi na kvantizaciju LSF vektora. Opisana je dekorelacijska uloga KLT, čijom se primjenom u transformacijskom kodiranju bolje iskorištavaju korelacije vektorskih komponenti LSF reziduala, što omogućava bolju kvantizaciju LSF vektora. Problem neuniformnosti statističkih zavisnosti realnih signala kroz cijeli vektorski prostor riješen je primjenom adaptivnog transformacijskog kodiranja, odnosno projektiranjem transformacijskih kodera adaptiranih na statistiku svake pojedine regije particioniranog vektorskog prostora.

Kao glavni doprinos, u svrhu bolje prilagodbe lokalnoj statistici izvora, ovaj rad razmatra primjenu i prilagodbu entropijski ograničene tehnike kodiranja dekoreliranih LSF reziduala. Varijabilna izlazna brzina entropijskih kodera prilagođena je fiksnoj prijenosnoj brzini AMR kodera kombiniranjem tehnika varijacije koraka kvantizacije i skraćivanja transformiranih vektora, koje omogućavaju ograničavanje duljine entropijskog kôda. Kako odabir ukupnih entropija (koraka kvantizacije) značajno afektira svojstva predloženog LSF kvantizatora, eksperimentirano je s nekoliko različitih strategija za njihov optimalan odabir. Pokazano je da, zahvaljujući KLT, fina (točnija) kvantizacija nekoliko prvih komponenti transformiranih reziduala više doprinosi optimalnosti svojstava od izbjegavanja skraćivanja transformiranih reziduala korištenjem grubljeg kvantizacijskog koraka. Posljedično, kombinacije ukupnih entropija, koje rezultiraju nižom SD i boljim iskorištenjem dostupne fiksne prijenosne brzine, pokazuju relativno veći postotak p2 i p4 pogrešaka, indicirajući kompromis među spomenutim parametrima, koji treba zadovoljiti u procesu odabira ukupnih entropija ECSQ kvantizatora. Pokazano je da najbolje rezultate postiže asimetričan odabir ukupnih entropija većih od dostupne prijenosne brzine (kombinacija s indeksom 9 – tablica 6.1), iako je zadovoljavajuće rezultate moguće postići i odabirom simetričnih ukupnih entropija. U oba slučaja treba zadovoljiti kompromis, imajući na umu da prevelike ukupne entropije degradiraju prosječnu učinkovitost kodera.

Na osnovu provedenog istraživanja predložene su dvije verzije GMM temeljenog kvantizatora spektralne ovojnice. Predloženi parametri $m \times q$ verzije prilagođeni su ostvarenju najvećeg mogućeg stupnja sažimanja (odnos distorzije i brzine prijenosa),

dok je struktura 2-*best* verzije prilagođena smanjenju broja potrebnih matematičkih operacija za provođenje postupka kvantizacije. Pokazano je da je kvantizator optimalno projektirati u tri iteracije, upotrebljavajući kriterij najmanjeg umnoška apsolutnih vrijednosti komponenti reziduala u transformacijskoj domeni pri inicijalnom pridruživanju odgovarajućih komponenti Gaussove mješavine izdvojenim rezidualima.

Usporedba SD i odgovarajućih postotaka p2 i p4 pogrešaka pokazuju da kvantizacijska svojstva modificiranih AMR kodera značajno nadmašuju polazni referentni AMR koder. Međutim, činjenica da se spomenuto poboljšanje u kodiranju spektralne ovojnice ne reflektira u PESQ ocjenama pokazuje da je cjelokupna učinkovitost AMR kodera ograničena pogreškom kvantizacije pobude. Simulacijom predloženih algoritama u Matlabu, pokazano je da je učinak referentnog AMR kodera moguće ostvariti primjenom GMM LSF VQ, koristeći manje brzine prijenosa. Na taj način ostvarena je redukcija prijenosne brzine, pri tome zadržavajući učinkovitost polaznog sustava. U môdu s najvećom prijenosnom brzinom, $m \times q$ verzija modificiranog AMR kodera ostvaruje prosječnu redukciju prijenosne brzine za 7.33 bita/okviru, dok u ostalim modovima svojstva referentnog AMR kodera dostiže s brzinom prijenosa reduciranom do 3.3 bita/okviru. Pri tome je redukcija prijenosne brzine u 12k2 môdu plaćena dvostrukom računskom složenošću i memorijskim zahtjevima, dok je u ostalim modovima neovisna o prijenosnoj brzini i usporediva sa računskom složenošću i memorijskim zahtjevima AMR modova s najmanjom prijenosnom brzinom. Pojednostavljena *2-best* verzija modificiranog AMR kodera, u usporedbi s $m \times q$ verzijom, uz nepromijenjene memorijske zahtjeve donosi za faktor 2.6 reduciranu računsku složenost. Smanjena složenost rezultira nešto manjim redukcijama prijenosnih brzina kodirane spektralne ovojnice (5.06 bitova/okviru u môdu 12k2, te do 1.96 bitova/okviru u ostalim modovima rada). Na strani dekodera, GMM-temeljeni kvantizator, neovisno o varijanti i brzini prijenosa, neznatno doprinosi složenosti cjelokupnog modificiranog sustava. Memorijski zahtjevi po dekoder slični su polaznom sustavu za sve modove, osim najbržeg, u kojem su zbog entropijskog kodiranja utrostručeni. Činjenica da je elemente Huffmanovih tablica, u usporedbi s vektorskim komponentama kodnih knjiga standardiziranog kodera, moguće pohraniti u znatno manje memorijske jedinice, te smanjenje prosječnog broja simbola po tablici pri reduciranim brzinama prijenosa GMM LSF vektorskih kvantizatora koji postižu svojstva standardiziranog kodera, dodatno reduciraju omjere memorijskih zahtjeva i računske složenosti u korist predloženog kvantizatora.

Kako je pokazano, memorijski zahtjevi predloženog kvantizatora ne ovise o njegovoj varijanti, te su pod pretpostavkom neovisnosti veličine Huffmanovih tablica o prijenosnoj brzini kodera, skupa s računskom složenošću, neovisni o njegovoj rezoluciji. Ta činjenica omogućava relativno jednostavnu prilagodbu na novu izlaznu brzinu podatkovnog toka, bez promjene njegove strukture, računske složenosti ili memorijskih zahtjeva. Promjena rezolucije kvantizatora zahtijevat će projektiranje novih ECSQ kvantizatora i odgovarajućih entropijskih kodera, dok će zbog neovisnosti GMM modela vektorskog procesa o rezoluciji kvantizatora, kvantizator s novom rezolucijom koristiti iste transformacijske matrice i vektore srednjih vrijednosti komponenti Gaussove mješavine.

Sve simulacije opisane u ovom istraživanju provedene su primjenom modela s osam komponenti Gaussove mješavine, te primjenom četiri ECSQ kvantizatora. Broj komponenti mješavine i ECSQ kvantizatora odabran je kao kompromis između broja dodatnih bitova, potrebnih za kodiranje njihovih indeksa, i poželjne fleksibilnosti koju unose u sustav. U smislu smjernica za buduće radove, provedeno bi se istraživanje moglo nastaviti variranjem broja komponenti mješavine i ECSQ kvantizatora. Nadalje, kako projektirani GMM-temeljeni vektorski kvantizatori za svaki ECSQ kvantizator i svaku komponentu vektora koriste zasebne tablice entropijskih kodera (ukupno $d \times q$ tablica), istraživanje mogućnosti njihovog združivanja, odnosno korištenja zajedničkih tablica, moglo bi rezultirati značajnom redukcijom memorijskih zahtjeva kvantizatora. Osim navedenog, buduća bi istraživanja mogla obuhvatiti potvrdu dobivenih rezultata nekom od subjektivnih metoda mjerenja kvalitete govora. Na taj bi se način verificirali rezultati dobiveni objektivnim mjerenjem PESQ modelom, u smislu postojanja artefakata koji su ostali skriveni iza njegove neidealnosti.

Učinkovitost pristupa kodiranju spektralne ovojnice, primjenjenog u ovom radu, ovisna je o sličnosti razdiobe ulaznog vektorskog procesa (pogreške predikcije LSF vektora), odnosno razdiobe svakog od dekompozicijom dobivenih izvora, s Gaussovom razdiobom. Za Gaussovu razdiobu izvorâ, koji odgovaraju komponentama Gaussove mješavine, KLT transformacija rezultirat će statistički nezavisnim komponentama transformiranog vektorskog procesa, koji onda omogućavaju njihovu optimalnu kvantizaciju primjenom skalarnih kvantizatora. Međutim, za razdiobe koje ne odgovaraju Gaussovoj, KLT transformacija nije u stanju ukloniti sve statističke zavisnosti, te rezultira statistički zavisnim komponentama transformiranog vektorskog procesa, što uzrokuje njihovu suboptimalnu kvantizaciju skalarnim kvantizatorima. Radi se o (nelinearnim) statističkim zavisnostima višeg reda (engl. *higher-order dependencies*) za čije uklanjanje postoji više rješenja. Među njih spada i tzv. jezgrena analiza osnovnih komponenata (engl. *kernel principal component analysis*, kernel PCA), koja zapravo predstavlja nelinearnu verziju KLT. Jedan od očekivanih dobitaka bila bi manja osjetljivost rezultata transformacije o pretpostavci Gaussove razdiobe ulaznog vektorskog procesa [37]. Kao alternativa kernel PCA, postoji skupina algoritama nazvana *analiza nezavisnih komponenata* (engl. *indepenendent component analysis*, ICA) [38], koji nakon transformacije rezultiraju statistički nezavisnim komponentama vektorskog procesa. Optimalnost ove transformacije pretpostavlja da najviše jedna komponenta ulaznog vektorskog procesa ima Gaussovu razdiobu. Spomenute bi alternative (primjeni kombinacije GMM i KLT), unatoč vjerojatno većoj računskoj složenosti, primjenom na realnim vektorskim procesima trebale rezultirati statistički nezavisnom (manje redudantnom) pogreškom predikcije, te samim tim i efikasnijim kodiranjem. U tom bi se smislu provedeno istraživanje u budućim radovima moglo nastaviti kroz primjenu kernel PCA, odnosno ICA transformacije, kao alternativnom mehanizmu za uklanjanje statističkih zavisnosti među komponentama realnog vektorskog procesa, čija razdioba ne odgovara Gaussovoj.

Na kraju, kao smjernica za buduće radove, treba ukazati i na alternativni način izračuna LPC koeficijenata minimizacijom entropije pogreške predikcije [39] umjesto minimizacije njene srednje kvadratne pogreške, primjenjene u ovom radu. Minimizacija

entropije pogreške predikcije (estimirane neparametarskim estimatorom s Gaussovom jezgrom) odgovara slučaju kada je *pdf* funkcije pogreške delta funkcija centrirana u nuli, tj. najvjerojatniji ishod minimizacije je nulta pogreška i to za bilo koju razdiobu (prilagodljivost na razdiobu dobije se neparametarskom procjenom entropije iz samih podataka).

A. Opis strukture izvornog kôda simulacijskog modela

Simulacijski model, prikazan na slici 6.1, obuhvaća standardnu implementaciju AMR kodera u **C** programskom jeziku, izvršni kôd PESQ algoritma, te simulaciju GMM-temeljenih LSF vektorskih kvantizatora u programskom paketu MATLAB.

A.1. Priprema govornih datoteka

Sve datoteke koje sačinjavaju trening i evaluacijsku bazu govornih sekvenci nalaze se u *orig_wav* direktoriju. Radi se o *.wav* datotekama koje sadrže 16-bitne uzorke govornih sekvenci u linearnom PCM (engl. *Pulse Code Modulation*) formatu, uzorkovanih frekvencijom 8 kHz. Kako AMR koder na ulazu predviđa uzorke 13-bitne rezolucije u linearnom PCM formatu, lijevo poravnate u 16-bitnoj riječi, prije procesiranja se tri najmanje značajna bita svih uzoraka u ulaznoj govornoj datoteci pozivom MATLAB funkcije *wav2inp* postavljaju na vrijednost nula. Ovim pozivom se ulazna *.wav* datoteka prevodi u datoteku s *.INP* ekstenzijom i sprema u *INP* direktorij, te se kao takva prosljeđuje u AMR koder kao ulazni argument. Nakon procesiranja AMR koderom, bitovni niz kodiranih parametara sprema se u datoteku s *.COD* ekstenzijom, koja se pohranjuje u *COD* direktorij. Pokretanjem izvršnog koda AMR dekodera, iz kodiranih se parametara *.COD* datoteke sintetizira govorni signal koji odgovara originalnom govornom signalu u *.INP* datoteci. Sintetizirani govorni signal sprema se u datoteku s *.OUT* ekstenzijom i pohranjuje u *OUT* direktorij. Govorne datoteke s *.OUT* ekstenzijom moguće je radi eventualnog preslušavanja prevesti u *.wav* format pozivom *out2wav* MATLAB funkcije.

A.2. Funkcija *sim_model*

Sve simulacije pokreću se pozivom matlab funkcije *sim_model* sa sljedećim ulaznim argumentima:

- *lsf_vq* – obavezan argument, koji određuje vektorski kvantizator koji se koristi za kvantizaciju LSF parametara. Argument prihvaća sljedeće vrijednosti:
    - *amr* – ovisno o môdu rada AMR kodera, koristi se SMQ/SVQ metoda referentnog AMR kodera,
    - *gmm_vq_mxq* – za kvantizaciju LSF parametara koristi se $m \times q$ varijanta GMM-temeljenog VQ,

- *gmm_vq_mbest* – za kvantizaciju LSF parametara koristi se *mbest* varijanta GMM-temeljenog VQ, pri čemu je za *2-best* varijantu potrebno *Mbest* varijabli u *sim_model* funkciji pridijeliti vrijednost 2.

- *amr_mode* – obavezan argument, koji određuje môd rada AMR kodera. Argument prihvaća sljedeće vrijednosti:
  - *1* – odgovara 12k2 môdu rada AMR kodera (12.2 kbit/s),
  - *2* – odgovara 10k2 môdu rada AMR kodera (10.2 kbit/s),
  - *3* – odgovara 7k95 môdu rada AMR kodera (7.95 kbit/s),
  - *4* – odgovara 7k4 môdu rada AMR kodera (7.4 kbit/s),
  - *5* – odgovara 6k7 môdu rada AMR kodera (6.7 kbit/s),
  - *6* – odgovara 5k9 môdu rada AMR kodera (5.9 kbit/s),
  - *7* – odgovara 5k15 môdu rada AMR kodera (5.15 kbit/s),
  - *8* – odgovara 4k75 môdu rada AMR kodera (4.75 kbit/s).
- *db* – obavezan argument, određuje bazu govornih sekvenci koje se procesiraju u pokrenutoj simulaciji. Argument prihvaća sljedeće vrijednosti:
  - *training* – pokrenuta simulacija procesira trening bazu govornih sekvenci. Ova vrijednost koristi se u postupku projektiranja GMM-temeljenog vektorskog kvantizatora, odnosno za mjerenje svojstava referentnog AMR kodera nad govornim sekvencama iz trening baze.
  - *evaluation* – pokrenuta simulacija procesira evaluacijsku bazu govornih sekvenci. Ova vrijednost koristi se za objektivno vrednovanje svojstava projektiranog GMM-temeljenog vektorskog kvantizatora, odnosno za mjerenje svojstava referentnog AMR kodera nad govornim sekvencama iz evaluacijske baze.
- *M* – opcijski argument, određuje broj komponenti mješavine koje koristi GMM model.
- *q* – opcijski argument, određuje broj ECSQ kvantizatora korištenih u tehnici varijacije koraka kvantizacije.
- *HIcor* – opcijski argument, određuje indeks kombinacije ukupnih entropija ECSQ kvantizatorâ (tablica 6.1).
- *num_qd_it* – opcijski argument, određuje broj iteracija postupka projektiranja GMM-temeljenog LSF VQ.
- *bps* – opcijski argument, određuje rezoluciju GMM-temeljenog LSF VQ.

Svi opcijski argumenti koriste se za simulacije u kojima se za kvantizaciju LSF parametara koristi GMM-temeljen VQ (za *gmm_vq_mxq* i *gmm_vq_mbest* vrijednosti *lsf_vq* ulaznog argumenta). Odgovarajuće ulazne argumente potrebno je, osim pri

projektiranju GMM LSF VQ (*db = 'training'*), navesti i prilikom objektivnog vrednovanja njegovih svojstava nad evaluacijskom bazom (*db = 'evaluation'*), zbog učitavanja odgovarajućih parametara projektiranog kvantizatora.

Po završetku simulacije, funkcija *sim_model* rezultate privremeno sprema u *print_params.mat* datoteku, koji se onda pozivom MATLAB funkcije *print_to_excell* ponovno učitavaju i zapisuju u odgovarajuću *excell* datoteku.

A.3. Pokretanje simulacija

Ovisno o broju i vrijednostima gore opisanih ulaznih argumenata, pozivanjem funkcije *sim_model* moguće je pokrenuti nekoliko vrsta simulacija:

- poziv s prva tri ulazna argumenta:

sim_model (lsf_vq, amr_mode, db)

na mjestu prvog argumenta očekuje vrijednost *amr*. Ova simulacija zapravo pozivom izvršnog koda standardiziranog AMR kodera (*encoder.exe*), u odabranom môdu rada (*amr_mode*) procesira odabranu bazu (*db*) govornih sekvenci. Tijekom procesiranja, AMR koder u odgovarajuće datoteke izdvaja *a*-koeficijente i odgovarajuće LSF vektore prije i poslije kvantizacije, koji se onda upotrebljavaju za izračun SD parametara pozivom *calc_SD* funkcije. Kodirani parametri prosljeđuju se u standardizirani AMR dekoder, pozivom *decoder.exe* izvršnog kôda, te se reproducirane govorne sekvence (*.OUT*) uspoređuju s polaznim sekvencama (*.INP*) pozivom *pesq.exe* programa. Rezultati PESQ evaluacije spremaju u *_pesq_results.txt* datoteku, koja se za svaki poziv programa nadopunjuje sa po jednim retkom rezultata. Pozivom *decon_out_fils* MATLAB funkcije, reproducirane sekvence se upotrebom *out2wav* funkcije konvertiraju u *.wav* format i spremaju u *wav_out/*<môd>*/amr_orig* direktorij radi mogućeg preslušavanja. Pri tome, <môd> označava odabrani môd rada AMR kodera za trenutnu simulaciju. Na kraju simulacije, izračunate vrijednosti SD i PESQ ocjene spremaju se u odgovarajuće *excell* datoteke kao referentne vrijednosti polaznog sustava.

- poziv s prvih sedam ulaznih argumenata:

sim_model (lsf_vq, amr_mode, db, M, q, HIcor, num_qd_it)

na mjestu prvog argumenta očekuje vrijednost *gmm_vq_mxq* ili *gmm_vq_mbest*, što ovisno o *db* ulaznom argumentu podrazumijeva postupak projektiranja specificirane varijante GMM LSF VQ (*db = 'training'*), odnosno kvantizaciju vektora već projektiranim GMM LSF VQ (*db = 'evaluation'*). Iz MATLAB okruženja poziva se izvršni kôd AMR kodera (*encoder.exe*), koji onda, procesiranjem ulaznih govornih sekvenci u *amr_mode* môdu rada, iz odgovarajuće baze (*db*) izdvaja LSF vektore skupa s odgovarajućim rezidualnim vektorima.

Kod projektiranja kvantizatora (*db = 'training'*), nakon izdvajanja vektora iz trening baze govornih sekvenci slijedi iteriranje postupka projektiranja specificirane varijante GMM LSF VQ. Broj iteracija određen je ulaznim argumentom *num_qd_it*. Poziv *design_gmmvq* funkcije rezultira transformacijskim matricama i srednjim vrijednostima

M komponenti Gaussove mješavine, te koracima kvantizacije i Huffmanovim tablicama *q* ECSQ kvantizatora, čije su ukupne entropije određene indeksom *HIcor*. Projektirani GMM LSF VQ spektralnu ovojnicu kodira jednakim brojem bitova kao što to radi i referentni AMR koder u *amr_mode* môdu rada. Ovisno o varijanti projektiranog kvantizatora, pozivom *quant_mxq*, odnosno *quant_mbest* funkcije, parametri projektiranog kvantizatora koriste se kvantizaciju izdvojenih LSF vektora, čiji se novi rezidualni vektori (*tr_vec*) i indeksi pridruženih komponenti mješavine (*s_gmc*) i ECSQ kvantizatora (*s_ecsq*) prosljeđuju u sljedeću iteraciju postupka projektiranja. Nakon zadnje iteracije, pozivom *recnstr_vecs* funkcije radi se rekonstrukcija kvantiziranih rezidualnih vektora u originalnu domenu, koji onda pozivanjem funkcije *supstitute_to_amr*, ponovnim izvršavanjem AMR kodera zamjenjuju originalne rezidualne vektore, kvantizirane SMQ/SVQ metodama. Na taj se način izračun ostalih parametara AMR kodera u drugom pozivu temelji na eksterno kvantiziranim rezidualnim vektorima, čime se uklanja utjecaj kvantizacijskih algoritama referentnog AMR kodera. U izlazni podatkovni tok AMR kodera i dalje se spremaju indeksi pod-matrica/pod-vektora kvantiziranih SMQ/SVQ metodom. Na strani dekodera se, nakon dekodiranja (*decoder.exe*), rezidualni LSF vektori jednostavno zamjenjuju eksterno kvantiziranim vektorima, što omogućava ispravnu rekonstrukciju kodiranog govornog signala. Kao i kod referentnog kodera, svojstva modificiranog kodera nad trening bazom mjere se pozivom *pesq.exe* programa i *calc_SD* funkcije. Izračunate vrijednosti spremaju se u odgovarajuće *excell* datoteke pozivom *print_to_excell* funkcije. Svojstva *m×q* varijante spremaju se u *AMR_bps-Mxq.xls* datoteku, dok se svojstva *2-best* varijante spremaju u *AMR_bps-2best.xls* datoteku. *AMR_bps* u imenu datoteke označava da se radi o svojstvima GMM LSF VQ čija je rezolucija jednaka rezoluciji referentnog AMR kodera u odgovarajućem môdu rada. Na kraju se, pozivom *decon_out_fils* MATLAB funkcije, reproducirane sekvence konvertiraju u *.wav* format i spremaju u *wav_out/*<môd>*/M*<*M*>*_q*<*q*>*_bps*<bps>*_HIcor*<*HIcor*>*_qdIt*<*num_qd_it*>_<varija nta> direktorij radi mogućeg preslušavanja. Pri tome, varijable u < > zagradama napisane kurzivom odgovaraju ulaznim argumentima simulacije, dok ostale varijable redom predstavljaju:

- môd – odabrani môd rada AMR kodera za trenutnu simulaciju,
- bps – broj bitova/okviru korišten za kodiranje spektralne ovojnice, u ovom slučaju odgovara broju bitova korištenom u odgovarajućem môdu referentnog AMR kodera,
- varijanta – poprima *Mxq* ili *Mbest2* vrijednosti, ovisno o simuliranoj varijanti GMM LSF VQ specificiranoj *lsf_vq* ulaznim argumentom.

Parametri projektiranog kvantizatora spremaju se u *GMM_VQs* direktorij, u datoteku pod imenom:

<môd>*_M*<*M*>*_q*<*q*>*_bps*<bps>*_HIcor*<*HIcor*>*_qdIt*<*num_qd_it*>_<varijanta>*.mat*,

gdje varijable u < > zagradama imaju gore opisana značenja.

Kod objektivnog vrednovanja svojstava već projektiranog GMM LSF VQ (*db = 'evaluation'*), nakon izdvajanja vektora iz evaluacijske baze govornih sekvenci slijedi

učitavanje parametara specificirane varijante GMM LSF VQ. Pozivom *quant_mxq*, odnosno *quant_mbest* funkcije, učitani parametri kvantizatora koriste se kvantizaciju izdvojenih LSF vektora. Kao i kod projektiranja kvantizatora, kvantizirani rezidualni vektori se pozivom *recnstr_vecs* funkcije transformiraju u originalnu domenu, te pozivanjem *supstitute_to_amr* funkcije prosljeđuju u AMR koder i dekoder, gdje zamjenjuju originalne rezidualne vektore kvantizirane SMQ/SVQ metodama. Svojstva modificiranog kodera nad evaluacijskom bazom evaluiraju se i spremaju istovjetno opisanom postupku nad trening bazom. Na kraju se rekonstruirane govorne sekvence evaluacijske baze konvertiraju u *.wav* format i spremaju u isti direktorij u kojem su spremljene rekonstruirane govorne sekvence trening baze.

Iz opisanog se može zaključiti da je za kvantizaciju LSF parametara u MATLAB okruženju potrebno dva poziva izvršnog koda AMR kodera nad istom bazom govornih sekvenci. U prvom se pozivu, dakle, râdi izdvajanje LSF vektora radi njihove kvantizacije u MATLAB okruženju, kako bi u drugom pozivu eksterno kvantizirani vektori zamijenili vektore kvantizirane metodama koje koristi standardizirani koder.

- za poziv sa svih osam ulaznih argumenata:

  sim_model (lsf_vq, amr_mode, db, M, q, HIcor, num_qd_it, bps)

vrijedi sve što je već rečeno za poziv sa prvih sedam ulaznih argumenata uz dva izuzetka:

  - rezolucija projektiranog kvantizatora jednaka je vrijednosti ulaznog argumenta *bps*. Navođenjem ovog ulaznog argumenta u pozivu *sim_model* funkcije moguće je projektirati GMM LSF VQ čija se brzina prijenosa razlikuje od brzine prijenosa referentnog AMR kodera u odgovarajućem môdu rada.
  - rezultati objektivnog vrednovanja svojstava modificiranog AMR kodera ne zapisuju se u *AMR_bps-Mxq.xls* i *AMR_bps-2best.xls* datoteke, nego u *CORR_bps-Mxq.xls* ili *CORR_bps-2best.xls* datoteku, ovisno o varijanti GMM LSF VQ, specificiranog *lsf_vq* ulaznim argumentom.

Pokretanje simulacija s različitim ulaznim argumentima moguće je automatizirati pozivom *mrun* funkcije s jednim ulaznim argumentom. Ovisno o vrijednosti *sim_type* ulaznog argumenta, funkcija uzastopno poziva *sim_model* funkciju, pri tome mijenjajući ulazne argumente:

- za *sim_type = 'ref'*, *sim_model* funkcija poziva se s prva tri ulazna argumenta za sve vrijednosti *amr_mode* argumenta (1 do 8) nad obje baze govornih sekvenci.
- za *sim_type = 'amr_bps'*, *sim_model* funkcija poziva se s prvih sedam ulaznih argumenata. Za obje varijante GMM LSF VQ, nad obje baze govornih sekvenci, za svaki môd rada AMR kodera, *sim_model* funkcija poziva se za svaki indeks kombinacije ukupnih entropija ECSQ kvantizatora (1 do 20). Pri tome se postupak projektiranja provodi u *num_qd_it = 3* iteracije, koristeći GMM model s *M = 8* komponenti, te *q = 4* različita ECSQ kvantizatora.

- za *sim_type = 'corr_bps'*, *sim_model* funkcija poziva se sa svih osam ulaznih argumenata. Za obje varijante GMM LSF VQ, nad obje baze govornih sekvenci, za svaki môd rada AMR kodera, *sim_model* funkcija poziva se za sve cjelobrojne rezolucije kvantizatora u rasponu od 28 do 42 bita za môd 12k2, odnosno u rasponu od 16 do 30 bitova za sve ostale modove rada. Pri tome se postupak projektiranja provodi u *num_qd_it = 3* iteracije, koristeći GMM model s *M = 8* komponenti, te *q = 4* različita ECSQ kvantizatora s *HIcor = 9* indeksom kombinacije ukupnih entropija.

A.4. Hijerarhija kôda

Struktura poziva funkcija korištenih u simulacijskom modelu prikazana je u tablici A.1. Svaki stupac predstavlja jednu razinu poziva, dok svaka ćelija reprezentira jednu MATLAB funkciju ili izvršni kôd programa u **C** programskom jeziku. Navedene funkcije sadrže pozive funkcija koje im se nalaze s desne strane. Vremenski slijed pozivanja funkcija odgovara redoslijedu od vrha ka dnu, s tim da se ponovljeni pozivi ne navode.

Tablica A.1 *Struktura poziva funkcija korištenih u simulacijskom modelu.*

mrun	*sim_model*	*encoder.exe*		
		decoder.exe		
		pesq.exe		
		extract_from_amr	*encoder.exe*	
			read_extr_vecs	
		design_gmmvq	*train_GMM*	*Lbg_WED_nut*
				EM_update
			kltm	
			design_ecsq	*sel_gmc*
				trans_vecs
				diff_entr
			huff_code_design	*huffman*
		quant_mxq	*tr_w_klt*	
			huff_enc1	
			huff_enc2	
			recnstr_vecs	
			lsf_wt	
		quant_mbest	*tr_w_klt*	
			ord_gmc	
			huff_enc1	
			huff_enc2	
			recnstr_vecs	
			lsf_wt	
		recnstr_vecs		
		supstitute_to_amr	*encoder.exe*	
			decoder.exe	
			pesq.exe	
		calc_SD	*cep_dist*	
			lsf2a	
		decon_out_fils		
		print_to_excell		

U tablici je radi preglednosti izostavljen poziv standardnih MATLAB funkcija, te dodanih funkcija koje obavljaju rutinske zadatke, nebitne za opis simulacijskog modela. Sivom bojom označeni su programi koji se pozivaju prilikom mjerenja svojstava standardiziranog AMR kodera, zelenom bojom označene su funkcije i programi koji se izvršavaju u postupku objektivnog vrednovanja svojstava projektiranog GMM LSF VQ, dok se unija funkcija i programa označenih zelenom i žutom bojom izvršava prilikom projektiranja kvantizatora. Neoznačene funkcije koriste se pri izvođenju svih simulacija, bez obzira na vrijednosti ulaznih argumenata *sim_model* funkcije.

Opis MATLAB funkcija i njihovih ulaznih i izlaznih argumenata može se pronaći u zaglavlju odgovarajućih *.m* datoteka.

A.5. Prilagodba standardne implementacije AMR kodera

Za primjenu u simulacijskom modelu, standardizirana implementacija AMR kodera i dekodera prilagođena je u svrhu obavljanja nekoliko zadataka:

- za izdvajanje odgovarajućih parametara iz govornih sekvenci tijekom procesiranja AMR koderom, u njegov izvorni kôd dodana je funkcija:

*void SavePar(Float32 k[], unsigned int ParSize, FILE** ParFile, char* ParFileName)*

koja u datoteku pod imenom na koje pokazuje *ParamFileName*, na mjesto pokazivača *ParFile* sprema polje brojeva, odnosno vektor *k*, veličine *ParSize*. Pozivom ove funkcije s odgovarajućim argumentima, pri svakom izvođenju AMR kodera u *params_extr* direktorij spremaju se sljedeće datoteke koje sadrže odgovarajuće izdvojene parametre:

- *mid.lsf* – sadrži nekvantizirane LSF vektore koji se odnose na drugi pod-okvir;
- *mid_r.lsf* –sadrži nekvantizirane rezidualne LSF vektore koji se odnose na drugi pod-okvir;
- *new.lsf* – sadrži nekvantizirane LSF vektore, za môd 12k2 vektori se odnose na četvrti pod-okvir;
- *new_r.lsf* – sadrži nekvantizirane rezidualne LSF vektore, za môd 12k2 vektori se odnose na četvrti pod-okvir;
- *mid_q.lsf* –sadrži kvantizirane LSF vektore koji se odnose na drugi pod-okvir;
- *new_q.lsf* – sadrži kvantizirane LSF vektore, za môd 12k2 vektori se odnose na četvrti pod-okvir;
- *a.a* – sadrži nekvantizirane *a*-koeficijente;
- *aq.a* – sadrži kvantizirane *a*-koeficijente, dobivene pretvorbom iz kvantiziranih LSF vektora.

Svi kvantizirani i nekvantizirani LSF i odgovarajući rezidualni vektori izdvojeni su u *Q_plsf_5* funkciji u môdu rada 12k2, odnosno u *Q_plsf_3* funkciji standardiziranog AMR kodera u svim ostalim modovima rada. Datoteke koje počinju sa *mid* postoje samo u 12k2 môdu rada i sadrže vektore koje se odnose na drugi pod-okvir. *a*-koeficijenti izdvojeni su u *cod_amr* funkciji, nakon provedene kvantizacije u LSP domeni.

- za zamjenu originalnih rezidualnih LSF vektora (kvantiziranih SMQ/SVQ metodama) rezidualnim vektorima koji su kvantizirani GMM LSF VQ u MATLAB okruženju, u izvorni kôd AMR kodera dodana je funkcija:

*void ReplacePar (Float32 OrigPar[], unsigned int ParSize, FILE** ParFile, char* ParFileName)*

koja iz datoteke pod imenom na koje pokazuje *ParamFileName*, sa mjesta na koje pokazuje pokazivač *ParFile* iščitava polje brojeva, odnosno vektor veličine *ParSize* i sprema ga u polje, odnosno vektor *OrigPar*. Pozivom MATLAB funkcije *supstitute_to_amr*, eksterno kvantizirani rezidualni vektori razdvajaju se i spremaju u *params_mod* direktorij, u zasebne datoteke (*enc_<ime_ulazne_govorne_datoteke>_mod.lsf*) koje odgovaraju obrađenim ulaznim govornim datotekama. Iz ovih se datoteka, pozivom *ReplacePar* funkcije s odgovarajućim argumentima, originalni rezidualni vektori zamjenjuju vektorima koji su kvantizirani u MATLAB okruženju. Da bi AMR koder tijekom izvođenja napravio zamjenu, njegov izvršni kôd potrebno je pozvati sa pet ulaznih argumenata, gdje peti argument specificira ime datoteke koja sadrži eksterno kvantizirane rezidualne LSF vektore. U tu je svrhu izvorni kôd standardiziranog AMR kodera modificiran na način da prihvati dodatni ulazni argument

encoder [-dtx] mode speech_file bitstream_file lsf_rq_file

Po pokretanju izvršnog koda AMR kodera, eksterno kvantizirani vektori se kopiraju u *matlab.lsf* datoteku u korijenskom direktoriju, čije se ime onda univerzalno koristi u ostatku izvornog kôda. U suprotnom, ako AMR koder prilikom izvođenja ne treba napraviti zamjenu parametara, njegov izvršni kôd potrebno je također pozvati s pet argumenata, gdje se kao peti ulazni argument prosljeđuje prazna datoteka u korijenskom direktoriju pod imenom *empty_file.lsf*.

- sve navedeno za zamjenu originalnih rezidualnih LSF vektora u izvornom kôdu AMR kodera vrijedi i za AMR dekoder, s izuzetkom da dekoder umjesto 32-bitnih varijabli s pomičnim zarezom (*Float32*) koristi 32-bitne cjelobrojne varijable (*Word32*), te je deklaracija u dekoder dodane funkcije dana sa

*void ReplacePar (Word32 OrigPar[], unsigned int ParSize, FILE** ParFile, char* ParFileName)*

Iz istog je razloga eksterno kvantizirane rezidualne LSF vektore potrebno podijeliti s frekvencijom uzorkovanja (8000 Hz) i pomnožiti s 32768 (2^{15}), te spremiti u zasebne datoteke (*dec_<ime_ulazne_govorne_datoteke>_mod.lsf*)

koje će se proslijediti AMR dekoderu na zamjenu. Za zamjenu parametara, izvršni kôd AMR dekodera potrebno je onda pozvati s dodatnim, trećim argumentom, koji sadrži ime datoteke s eksterno kvantiziranim rezidualnim LSF vektorima

decoder input_file output_file dec_lsf_rq_file

U suprotnom, izvršni kôd AMR dekodera potrebno je također pozvati s tri ulazna argumenata, gdje se kao treći ulazni argument prosljeđuje prazna datoteka u korijenskom direktoriju pod imenom *empty_file.lsf*.

Literatura

[1] 3GPP TS 26.090 V6.0.0 Adaptive Multi-Rate (AMR) speech codec; Transcoding Functions, (2004).

[2] 3GPP TS 26.104 V5.2.0 ANSI C code for the floating-point Adaptive Multi Rate (AMR) speech codec, (2003-06).

[3] C. ARCHER, T. K. LEEN From mixtures of mixtures to adaptive transform coding. *Proceedings of International Joint Conference on Neural Networks*, July 1999, pp. 925-931.

[4] A. DEMPSTER, N. LAIRD, D. RUBIN Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39 (1977), 1–38.

[5] E. EKUDDEN, R. HAGEN, I. JOHANSSON, J. SVEDBERG The Adaptive Multi-rate Speech Coder. *Proceedings of the IEEE Workshop on Speech Coding,* (1999) Porvoo, Finland, 117-119.

[6] N. FARVARDIN, R. LAROIA Efficient encoding of speech LSP parameters using the discrete cosine transformation. *Proceedings ICASSP*, (1989), 168-171.

[7] W. R. GARDNER, B. D. RAO Theoretical analysis of the high-rate vector quantization of LPC parameters. *IEEE Transactions on Speech and Audio Processing*, 3 (1995), 367–381.

[8] A. GERSHO, R. M. GRAY *Vector Quantization and Signal Compression*. Massachusetts: Kluwer, 1992.

[9] V.K. GOYAL Theoretical foundations of transform coding. *IEEE Signal Processing Mag*. 18 (5) (September 2001).

[10] A. H. GRAY, J. D. MARKEL Quantization and bit allocation in speech processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-24, pp. 459-473, Dec (1976 a).

[11] A. H. GRAY, R. M. GRAY, J. D. MARKEL Comparison of optimal quantizations of speech reflection coefficients. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-25, pp. 9-23, Feb. 1977.

[12] R. M. GRAY *Source coding theory*. Kluwer Academic Publisher, 1990.

[13] R. M. GRAY, D. L. NEUHOFF Quantization. *IEEE Transactions on Information Theory* 44 (1998), 2325-2384.

[14] A. GYÖRGY, T. LINDER On the structure of optimal entropy constrained scalar quantizers. *IEEE Transactions on Information Theory*, 48 (2002), 416-427.

[15] P. HEDELIN, J. SKOGLUND Vector quantization based on Gaussian mixture models. *IEEE Transactions on Speech and Audio Processing*, 8 (2000), 385–401.

[16] Y. HUANG, P. M. SCHULTHEISS Block quantization of correlated Gaussian random variables. *IEEE Transactions on Communications Systems*, 11 (1963), 289–296.

[17] D. A. HUFFMAN A method for the construction of minimum-redundancy codes. *Proceedings of the IRE,* 40 (1952), 1098-1101.

[18] F. ITAKURA Line spectrum representation of linear predictive coefficients of speech signals. *J. Acoust. Soc. Am.,* vol. 57, Suppl. no. 1, pp. S35, Apr. 1975.

[19] N. S. JAYNANT, P. NOLL *Digital Coding of Waveforms*. Prentice-Hall, 1984.

[20] P. KABAL, R. P. RAMACHANDRAN Computation of line spectral frequencies using Chebyshev polynomials. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-34, pp. 1419-1426, Dec. 1986.

[21] W. B. KLEIJN, K. K. PALIWAL *Speech Coding and Synthesis*. Elsevier, Amsterdam, 1995.

[22] Y. LINDE, A. BUZO, R. M. GRAY An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 1 (1980), 84–95.

[23] S. SO, K. K. PALIWAL Multi-frame GMM-based block quantization of line spectral frequencies. *Speech Communication*, 47 (2005), 265-276.

[24] K. K. PALIWAL, B. S. ATAL Efficient vector quantization of LPC parameters at 24 bits/frame. *IEEE Transactions on Speech and Audio Processing*, 1 (1993), 3–14.

[25] D. A. REYNOLDS, R. C. ROSE Robust text-independent speaker identification using Gaussian mixture models. *IEEE Trans. Speech Audio Processing*, vol. 3, no. 1, pp 72–83, Jan. 1995.

[26] F. SOONG, B. JUANG Line Spectrum Pair (LSP) and speech data compression. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, San Diego, 1984, pp 1.10.1–1.10.4.

[27] A. D. SUBRAMANIAM, B. D. RAO Speech lsf quantization with rate independent complexity, bit scalability and learning. *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2 (2001), 705–708.

[28] A. D. SUBRAMANIAM, B. D. RAO PDF optimized parametric vector quantization of speech line spectral frequencies. *IEEE Transactions on Speech and Audio Processing*, 11 (2003), 130–142.

[29] N. SUGAMURA, F. ITAKURA Speech analysis and synthesis methods developed at ECL in NTT – from LPC to LSP. *Speech Communication*, vol. 5, No. 2, June 1986.

[30] T. TADIĆ, D. PETRINOVIĆ Adapting Entropy Constrained Coding of Spectral Envelope for Fixed-Rate Coding in AMR Speech Codec. *Proceedings of MIPRO 2010*, vol. II, pp. 340-345, May 2010.

[31] T. TADIĆ, D. PETRINOVIĆ Gaussian Mixture Model Based Quantization of Line Spectral Frequencies for Adaptive Multirate Speech Codec. Submitted to *Journal of Computing and Information Technology*, University Computing Centre, Zagreb, December 2009.

[32] R. VISWANATHAN, J. MAKHOUL Quantization properties of transmission parameters in linear predictive systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-23, pp. 309-321, June 1975.

[33] R. C. WOOD On optimum quantization. *IEEE Transactions on Information Theory*, vol. IT-15, no. 2, pp. 248-252, Mar. 1969.

[34] D. XIAOYAN, Z. YONGGANG, T. KUN Improved Memoryless GMM VQ for Speech Line Spectral Frequencies. Presented at the *8th International* Conference *on Signal Processing*, (2006) Beijing.

[35] Y. ZHANG, C. J. S. DESILVA An isolated word recognizer using the EM algorithm for vector quantization. *IREECON 1991*, Sydney, Australia, pp. 289-292.

[36] D. Y. ZHAO, J. SAMUELSSON, M. NILSSON GMM-based Entropy-Constrained Vector Quantization. *Proceedings of ICASSP 2007*, (2007) Hawaii, USA.

[37] F. R. BACH, M. I. JORDAN Kernel independent component analysis. *The Journal of Machine Learning Research*, 3, January 2003, p. 1-48.

[38] P. COMON Independent component analysis: a new concept? *Signal Processing*, 36(3), 1994, pp. 287-314.

[39] J. C. PRINCIPE *Information Theoretic Learning*. Springer, 2010.

Sažetak

KVANTIZACIJA FREKVENCIJA SPEKTRALNIH LINIJA U ADAPTIVNOM KODERU GOVORNOG SIGNALA S VIŠE BRZINA PRIJENOSA PRIMJENOM MODELA S GAUSSOVIM MJEŠAVINAMA

Ključne riječi: *model s Gaussovim mješavinama, Karhunen-Loève transformacija (KLT), frekvencije spektralnih linija (LSF), adaptivni koder govornog signala s više brzina prijenosa (AMR), kodiranje govornog signala, transformacijsko kodiranje, vektorska kvantizacija (VQ), skalarna kvantizacija s ograničenom entropijom (ECSQ).*

U ovom radu istražena je mogućnost primjene modela sa Gaussovim mješavinama (engl. *Gaussian Mixture Model*, GMM) u svrhu kvantizacije frekvencija spektralnih linija (engl. *Line Spectral Frequencies*, LSFs) u adaptivnom koderu govornog signala s više brzina prijenosa (engl. *Adaptive Multi-Rate codec*, AMR). Primjenom GMM-a estimirana je parametarska reprezentacija funkcije gustoće vjerojatnosti (engl. *probability distribution function*, *pdf*) pogreške predikcije LSF parametara, koji reprezentiraju spektralnu ovojnicu govornog signala. Predložen je kvantizator spektralne ovojnice, koji za svaku komponentu mješavine koristi transformacijski koder temeljen na Karhunen-Loève transformaciji (KLT) i skalarnoj kvantizaciji (engl. *Scalar Quantization*, SQ) transformiranih reziduala. Istražena je mogućnost korištenja predloženog postupka kvantizacije u postojećem AMR koderu govornog signala isključivo uz izmjenu dijela algoritma koji se odnosi na kvantizaciju LSF vektora. Kao glavni doprinos, u svrhu bolje prilagodbe lokalnoj statistici izvora, ovaj rad razmatra prilagodbu entropijski ograničenog (engl. *Entropy Constrained*, EC) kodiranja dekoreliranih LSF reziduala za primjenu u AMR koderu fiksne brzine prijenosa. Napravljeno je objektivno vrednovanje i usporedba učinkovitosti sažimanja, računske složenosti i memorijskih zahtjeva polaznog i modificiranog sustava. Rezultati simulacija pokazuju da predloženi kvantizator ostvaruje bolje sažimanje od algoritama korištenih u referentnom AMR koderu, reducirajući prosječnu brzinu prijenosa do 7.33 bita/okviru, pri značajno manjoj računskoj složenosti i memorijskim zahtjevima, koji ne ovise o brzini prijenosa.

Abstract

GAUSSIAN MIXTURE MODEL BASED QUANTIZATION OF LINE SPECTRAL FREQUENCIES FOR ADAPTIVE MULTIRATE SPEECH CODEC

Key words: *Gaussian mixture models (GMMs), Karhunen-Loève transform (KLT), line spectral frequency (LSF), Adaptive Multi-Rate (AMR), speech coding, transform coding, vector quantization (VQ), entropy constrained scalar quantizer (ECSQ).*

In this thesis, the use of a Gaussian Mixture Model (GMM) based quantizer for quantization of Line Spectral Frequencies (LSFs) in the Adaptive Multi-Rate (AMR) speech codec is investigated. A parametric GMM model is estimated, modeling the probability density function (*pdf*) of the prediction error (residual) of mean-removed LSF parameters that are used in the AMR codec for speech spectral envelope representation. The studied GMM vector quantizer is based on transform coding using Karhunen-Loève transform (KLT) and transform domain scalar quantizers (SQ), individually designed for each Gaussian mixture. The applicability of such a quantization scheme in the existing AMR codec has been investigated by solely replacing the AMR LSF quantization algorithm segment. The main novelty in this thesis lies in applying and adapting the entropy constrained (EC) coding for fixed-rate scalar quantization of transformed residuals thereby allowing for better adaptation to the local statistics of the source. The compression efficiency, computational complexity and memory requirements of the proposed algorithm are studied and evaluated. Experimental results show that the GMM-based EC quantizer provides better rate/distortion performance than the quantization schemes used in the referent AMR codec by saving up to 7.33 bits/frame at much lower rate-independent computational complexity and memory requirements.

Životopis

Rođen sam 16. svibnja 1974. u St. Gallen-u, Švicarska. Osnovnu školu završio sam u Sarajevu, Bosna i Hercegovina. Od 1989. do 1993. zbog ratnih sam okolnosti srednjoškolsko obrazovanje prošao kroz tri srednje škole. Dvije godine elektrotehničke srednje škole nastavio sam u trećoj godini matematičke gimnazije u Sarajevu, te sam četvrtu godinu i srednjoškolsko obrazovanje završio kao Ekonomski tehničar u Srednjoj školi Ivan Goran Kovačić, Kiseljak, Bosna i Hercegovina. Po završenom srednjoškolskom obrazovanju, 1994. upisujem Fakultet elektrotehnike i računarstva u Zagrebu. Diplomirao sam 1999., smjer Radiokomunikacije i profesionalna elektronika. Diplomski rad izradio sam na Zavodu za elektroničke sustave i obradbu informacija (ZESOI) na temu „Višepojasno pobuđeni koder govornog signala“. Od kolovoza 1999. zaposlen sam u kompaniji Ericsson Nikola Tesla d.d. gdje trenutno obavljam poslove programskog inženjera u Institutu za telekomunikacije.

Objavljeni radovi: *„Adapting Entropy Constrained Coding of Spectral Envelope for Fixed-Rate Coding in AMR Speech Codec“*, MIPRO'10, Opatija, Croatia, 2010.