

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Danijel Domazet

**PROGRAMSKA IZVEDBA OPĆENITOG
AUDIO KODERA PREMA NORMI MPEG-4**

MAGISTARSKI RAD

Zagreb, 2004.

Magistarski rad je izrađen na:

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
Sveučilište u Zagrebu, Republika Hrvatska
Zavodu za Automatiku i Procesno Računarstvo
Grupa Računarski Sustavi i Procesi

Mentor: Prof.dr.sc. Mario Kovač

Magistarski rad ima 225 stranica.

Rad br. _____ .

Povjerenstvo za ocjenu magistarskog rada:

1. Prof.dr.sc. Mario Žagar, predsjednik
2. Prof.dr.sc. Mario Kovač, mentor
3. Doc.dr.sc. Davor Antonić, član (ETF Osijek)

Povjerenstvo za obranu magistarskog rada:

1. Prof.dr.sc. Bojan Ivančević, predsjednik
2. Prof.dr.sc. Mario Kovač, mentor
3. Doc.dr.sc. Davor Antonić, član (ETF Osijek)

Datum obrane 27.02.2004.

Zahvale

Zahvaljujem Danijeli na neizmjernoj ljubavi i strpljenju.

Zahvaljujem mami Miri i tati Josipu na svemu.

Rad posvećujem Luki (6) i Almi (1), Vujinim uzdanicama!

Sadržaj

Zahvale	3
Sadržaj	4
1 Multimedija	12
1.1 Uvod	12
1.2 Audio i Internet.....	13
1.3 Metode audio kompresije	13
1.3.1 Kompresija bez gubitaka	14
1.3.2 Kompresija sa gubitcima	14
1.4 Zaključak	15
2 Ljudski slušni sistem	16
2.1 Uvod	16
2.2 Građa uha.....	16
2.2.1 Vanjsko uho.....	17
2.2.2 Srednje uho	17
2.2.3 Unutrašnje uho.....	18
2.3 Slušni proces.....	19
2.4 Osobine slušnog sistema.....	19
2.4.1 Karakteristična frekvencija.....	19
2.4.2 Nelinearnost slušnog sistema	20
2.4.3 Binauralno slušanje	21
2.4.4 Efekt prvenstva	22
2.4.5 Efekt <i>zabave</i>	23
2.4.6 BMLD efekt.....	23
2.5 Zaključak	23
3 Osnove psihoakustike	24
3.1 Uvod	24

3.2	Apsolutni prag čujnosti.....	24
3.3	Kritični pojasevi bazilarne membrane	25
3.4	Simultano maskiranje	27
3.5	Širenje maskiranja	29
3.6	Vremensko maskiranje	31
3.7	Zaključak	31
4	Percepcijsko audio kodiranje	33
4.1	Uvod	33
4.2	Opći audio koder.....	33
4.2.1	Vremensko-frekvencijska transformacija.....	35
4.2.2	Psihoakustički modul.....	39
4.2.2.1	Posebna vremensko-frekvencijska transformacija	40
4.2.2.2	Funkcija širenja maskiranja.....	41
4.2.2.3	Rezultati psihoakustičke analize.....	41
4.2.2.3.1	Omjer <i>signal/prag maskiranja</i>	41
4.2.2.3.2	Tonalitet.....	42
4.2.2.3.3	Tranzijentnost	43
4.2.2.4	Efekt <i>pred-jeke</i>	45
4.2.3	Uobličavanje frekvencijskog spektra	46
4.2.4	Kvantizacija.....	47
4.2.5	Kompresija bez gubitaka	48
4.2.6	Kodirani tok.....	48
4.3	Povijesni razvoj audio kodiranja	49
4.3.1	Rezultati istraživačkog rada	50
4.3.1.1	OCF koder	50
4.3.1.2	PXFM koder	50
4.3.1.3	Brandenburg – Johnson hibridni koder	50
4.3.1.4	CNET koder.....	50
4.3.1.5	ASPEC koder.....	51

4.3.1.6	DPAC koder	51
4.3.1.7	MASCAM koder	51
4.3.1.8	MUSICAM koder	51
4.3.2	Standardizirani audio koderi	52
4.3.2.1	ISO MPEG-1	52
4.3.2.2	ISO MPEG-2	53
4.3.2.3	ISO MPEG-4	54
4.3.2.4	PASC koder	54
4.3.2.5	ATRAC koder	54
4.3.2.6	PAC koder	55
4.3.2.7	AC-2 i AC-3 koderi	55
4.4	Zaključak	56
5	Međunarodni standard ISO 14496 (MPEG-4)	57
5.1	Uvod	57
5.2	MPEG-4 Audio	58
5.2.1	Prirodni audio	59
5.2.2	Sintetički audio	59
5.2.3	Profili, audio objekti i moduli	60
5.2.3.1	Profili	60
5.2.3.2	Audio objekti i moduli	62
5.2.3.2.1	Govor	62
5.2.3.2.2	Strukturirani audio	62
5.2.3.2.3	Složeni audio	63
5.3	MPEG-4 <i>General Audio</i>	65
5.3.1	<i>General Audio</i> moduli	66
5.3.1.1	Vremensko-frekvencijska transformacija	66
5.3.1.1.1	MDCT filtarski slog	66
5.3.1.1.2	Filtarski slog sa malim kašnjenjem	72
5.3.1.1.3	Hibridni filtarski slog	72
5.3.1.2	Obrada spektra	73

5.3.1.3	Predviđanje	73
5.3.1.4	Kvantizacije	74
5.3.1.4.1	AAC kvantizacija.....	74
5.3.1.4.2	Twin VQ kvantizacija.....	76
5.3.1.4.3	BSAC kvantizacija.....	77
5.3.1.5	Skalabilnost	77
5.3.1.6	Korekcije i prevencije grešaka	79
5.4	Zaključak	80
6	RAAC izvedba MPEG-4 općenitog audio koda	81
6.1	Uvod	81
6.2	Proces kodiranja.....	83
6.3	Inicijalizacija koda.....	85
6.3.1	Ulazne veličine	85
6.3.2	Analiza ulaznog signala.....	85
6.3.3	Prosječna veličina bloka	86
6.4	Kodiranje bloka od 1024 uzorka.....	87
6.4.1	Formiranje ulaznog bloka.....	87
6.4.2	Psihoakustička analiza.....	87
6.4.3	Izbor dužine bloka	88
6.4.4	Izbor oblika prozora	89
6.4.5	<i>Prozorovanje</i> i MDCT transformacija.....	92
6.4.6	Limitiranje frekvencijskog raspona.....	96
6.4.7	Obrada spektra.....	96
6.4.8	Kvantizacija.....	97
6.4.9	Huffmanovo kodiranje.....	101
6.4.10	Grupiranje kratkih prozora	105
6.4.11	<i>Bit-rezervoar</i> i bitovi za popunjavanje	110
6.4.12	Kodirani tok.....	111
6.4.12.1	Zaglavlje	112
6.4.12.2	Single_channel_element.....	113
6.4.12.3	Channel_pair_element.....	113

6.4.12.4	Individual_channel_stream.....	114
6.4.12.5	Fill_element.....	115
6.5	Zaključak	115
7	Psihoakustički model izveden u RAAC koderu	116
7.1	Uvod	116
7.2	Cilj psihoakustičke analize	116
7.3	Psihoakustički model 2.....	118
7.4	Proces psihoakustičke analize.....	118
7.5	Tok psihoakustičke analize.....	120
7.5.1	Ulazni vremenski signal	120
7.5.2	Frekvencijska domena.....	121
7.5.3	Energija po maskirnim pojasevima	122
7.5.4	Maskirna energija i funkcija širenja maskiranja.....	124
7.5.5	Proračun tonaliteta.....	126
7.5.6	Omjer <i>signal/šum</i>	128
7.5.7	Energetski prag maskiranja	130
7.5.8	Percepcijska entropija.....	135
7.5.9	Detekcija tranzijentnosti signala.....	137
7.6	Zaključak	139
8	MPEG-4 <i>alati</i> izvedeni u RAAC koderu	140
8.1	<i>Vremensko uobličavanje šuma</i>	140
8.1.1	Općenito	140
8.1.2	Linearno predviđanje u vremenskoj domeni	141
8.1.3	Linearno predviđanje u frekvencijskoj domeni.....	142
8.1.4	Dualnost.....	142
8.1.5	Predviđanje metodama otvorene i zatvorene petlje.....	143
8.1.6	Zaključak	144
8.1.7	TNS realizacija u RAAC koderu.....	145
8.2	<i>Združeni-stereo</i> kodiranje.....	149
8.2.1	Općenito	149
8.2.2	<i>Zbroj/razlika</i> kodiranje.....	149

8.2.2.1	Općenito	149
8.2.2.2	RAAC izvedba <i>zbroj/razlika</i> kodiranja.....	150
8.2.3	<i>Intenzitet-stereo</i> kodiranje	153
8.2.3.1	Općenito	153
8.2.3.2	Problemi <i>intenzitet-stereo</i> kodiranja.....	154
8.2.3.3	RAAC izvedba <i>intenzitet-stereo</i> kodiranja.....	155
8.2.3.4	<i>Intenzitet-stereo</i> i čisti tonalni signali u RAAC koderu	157
8.3	<i>Percepcijska zamjena šuma</i>	158
8.3.1	Općenito	158
8.3.2	RAAC izvedba PNS kodiranja	159
8.3.3	Problemi pri upotrebi PNS kodiranja	160
8.3.3.1	Niske frekvencije	160
8.3.3.2	Pre-upotreba.....	160
8.3.3.3	PNS i čisti tonalni signali u RAAC koderu	161
8.3.3.4	PNS i BMLD efekt	162
8.3.4	Primjer PNS kodiranja.....	162
8.3.5	Zaključak	163
9	Ocjena RAAC koda	164
9.1	Uvod	164
9.2	Subjektivna analiza.....	164
9.2.1	<i>Double-blind triple-stimulus with hidden reference</i> test metoda	166
9.2.2	Izbor audio uzoraka	168
9.2.3	Karakteristike audio sistema.....	170
9.2.4	Akustičke karakteristike okoline	170
9.2.5	Izbor individualnih slušatelja	170
9.2.6	Skala za ocjenjivanje	170
9.2.7	Konfiguracije koda	171
9.2.7.1	<i>Normal</i>	172
9.2.7.2	<i>Full</i>	172

9.2.7.3	<i>Basic</i>	172
9.2.7.4	<i>Extra PNS</i>	172
9.2.7.5	<i>Extra TNS</i>	172
9.2.8	Rezultati testiranja	176
9.2.8.1	Ocjene pojedinačnih audio uzorka	177
9.2.8.1.1	<i>Adam</i>	179
9.2.8.1.2	<i>Cast</i>	180
9.2.8.1.3	<i>Gspi</i>	181
9.2.8.1.4	<i>Harp</i>	182
9.2.8.1.5	<i>Horn</i>	183
9.2.8.1.6	<i>Moza</i>	184
9.2.8.1.7	<i>Pump</i>	185
9.2.8.1.8	<i>Spfe</i>	186
9.2.8.1.9	<i>Tehn</i>	187
9.2.8.1.10	<i>Zeph</i>	188
9.2.8.2	Analiza uspješnosti	189
9.2.8.3	Nominalni bitski protok	189
9.2.8.4	Frekvencijski spektar	189
9.2.8.5	Detekcija tonalnosti	190
9.2.8.6	Detekcija tranzijentnosti	191
9.2.8.7	TNS upotreba	191
9.2.8.8	Izbor oblika prozora	192
9.2.8.9	Upotreba <i>zbroj/razlika</i> kodiranja	193
9.2.8.10	PNS upotreba	194
9.2.8.11	Upotreba <i>intenzitet-stereo</i> kodiranja	196
9.3	Objektivna analiza	197
9.3.1	Standard ITU-R BS. 1387	197
9.3.2	OPERA program	199
9.3.3	Rezultati	199
9.3.4	Usporedba rezultata objektivnog sa subjektivnim testiranjem	204
9.3.5	Zaključak	205
9.4	Brzina RAAC koda	205

9.5	Zaključak	206
10	Zaključak	208
10.1	Cilj razvoja RAAC koda i postignuto	208
10.2	Nadogradnje RAAC koda.....	210
10.3	Budućnost MPEG-4 standarda	210
A	Sadržaj priloženog CD-a	212
B	Popis korištenih alata	213
	Reference i bibliografija	214
	Sažetak	223
	Abstract	224
	Životopis	225

Poglavlje 1

1 Multimedija

1.1 Uvod

U stvarnom svijetu čovjek svoju okolinu doživljava pomoću pet čula: vidom, sluhom, okusom, mirisom i dodirom. Sa pojavom računala pojavila se i posebna grana računarskih znanosti koja se bavi modeliranjem i simuliranjem spomenutih čula na računalu. Naziva se *multimedija* (lat. *multus* – više i lat. *media* – medij za prijenos). Glavni cilj multimedije je da što bolje opiše stvarni svijet i predstavi ga virtualno, putem računalne opreme. Za sada multimedija uspješno modelira dva osjeta, vid i sluh, kroz dvije glavne multimedijalne komponente: video i audio.

Zadnjih desetak godina svjedoci smo velikog rasta popularnosti multimedijalnih sadržaja na računalima. Razloga za ovakav trend ima nekoliko, a jedan od glavnih je sveopća prisutnost i dostupnost računala, koja su zahvaljujući velikom tehnološkom napretku, performansama dostigla vrlo visoku razinu. Drugi bitan razlog nagle popularizacije multimedije je *Internet* koji je u samo nekoliko godina svoga postojanja pružio neslućene mogućnosti širenja, distribucije, razmjene i kopiranja multimedijalnih sadržaja.

Jedna od osnovnih komponenti multimedijalnih tehnologija, koja je aktualizirana pojavom Interneta, jeste audio kompresija. Audio kompresija je danas sinonim za *audio kodiranje* što u široj interpretaciji znači: predstavljanje kompleksnih audio signala sa što manjim zauzećem medija za pohranjivanje i transport. Audio kodiranje je danas već našlo svoju primjenu. Jedne od osnovnih primjena audio kodiranja su: digitalno emitiranje radio signala (engl. *Digital Broadcasting*, *Digital Radio Mondiale*), audio za digitalnu televiziju (engl. *High Definition Digital Television*), čuvanje glazbe na tvrdim diskovima, prijenos govora putem Interneta (engl. *voice over IP*), razmjena datoteka (engl. *file-sharing*) itd.

Pogledajmo najprije kratak pregled zapisa audio signala kroz povijest, trenutne tendencije i smjerove budućeg razvoja.

1.2 Audio i Internet

Osnovni zapis audio signala na računalu zasnovan je na tzv. pulsno-kodnoj modulaciji (engl. *Pulse Code Modulation – PCM*). Digitalni PCM audio signal se sastoji od diskretno predstavljenog analognog signala u vremenu (uzorkovanje) i u amplitudi (kvantizacija). Moglo bi se reći da je PCM najsiroviji oblik digitalnog audio zapisa. Kako čovjek čuje frekvencije do 20 kHz, frekvencija uzorkovanja prema Shannonovom teoremu bi trebala biti barem dva puta veća, pa se obično uzima 44.1 kHz (npr. *Compact Disk* zapis). Ako jedan uzorak u zapisu zauzme 16 bita, lako se izračuna da je za jednu sekundu digitalnog PCM audio signala potrebno osigurati 706 kB memorije. Ovakav način zapisa audio signala obzirom na zauzeće resursa vrlo je skup i nepraktičan, a pogotovo je neprikladan za prijenos audio sadržaja putem Interneta, obzirom da je propusnost Internet veza u većini slučajeva vrlo ograničena.

Pogledamo li Tablicu 1.1, zaključujemo da je novi način digitalnog zapisa audio signala postao imperativ, pogotovo nakon popularizacije Interneta.

Tablica 1.1: Različiti mediji imaju različitu kvalitetu signala i proporcionalno zauzeće memorije. Očito je da niti jedan od formata nije prihvatljiv za prijenos putem Interneta. Prosječna brzina modema od 56 kb/s nije dovoljna čak niti za prijenos telefonskog signala.

Medij	Uzoraka po sekundi	Biti po uzorku	Kanala	Frekvencijski opseg [kHz]	Omjer <i>signal/šum</i> [dB]	Zauzeće [kb/s]
DVD	96000	24	6	48	144	13824
CD	44100	16	2	22	96	1411
FM radio	32000	12	2	16	72	768
Telefon	8000	8	1	3.4	48	64

1.3 Metode audio kompresije

Novi način zapisa audio signala je primarno trebao osigurati što manje zauzeće resursa pri pohrani i prijenosu, pa je to i bila glavna ideja svih poduzetih istraživanja koja su uslijedila.

Također, audio signal ne bi smio imati izraženo lošiju kvalitetu od originalnog signala, što je s druge strane bio i najveći izazov. Rezultat poduzetih istraživanja bilo je naglo aktualiziranje *audio kodiranja* ili *audio kompresije*, koje je postalo jedno od prominentnijih područja u multimediji.

1.3.1 Kompresija bez gubitaka

Određenu kompresiju audio signala moguće je postići i standardnim, odavno poznatim metodama kompresije bez gubitaka, gdje se nakon rekonstrukcije postiže signal istovjetan originalu. Algoritmi kompresije bez gubitaka zasnivaju se na metodama eliminacije ponavljanja (redundancije). Eliminacija ponavljanja se postiže upotrebom takozvanih *variable length* entropijskih kompresijskih algoritama koji rade na sljedećem principu: simbolima koji se češće pojavljuju pridružuju se kraći kodovi, a onima koji se pojavljuju rijetko duži, pa se tako postižu uštede. Očito je da efikasnost ovakvih algoritama direktno ovisi o svojstvu ulaznog signala: što više ponavljanja (manja entropija) to je algoritam efikasniji. Huffmanovo kodiranje, *run-length* kodiranje i aritmetičko kodiranje tipični su predstavnici ove kategorije algoritama kompresije.

Praktično prihvatljiv rezultat audio kompresije za Internet upotrebu bi bio omjer kompresije veći od 1:10, pa su svi pokušaji kompresije bez gubitaka unaprijed bili osuđeni na neuspjeh, ili barem nisu mogli zasebno izazvati nikakav revolucionaran pomak. Njihov maksimalni omjer kompresije je oko 1:3. Zbog ovog razloga je ozbiljnije pristupljeno razvoju kompresijskih algoritama sa gubiticima, koji bi komprimirali audio signal na način da, nakon rekonstrukcije, novo-dobiveni audio signal ne bude identičan originalu, ali i da u najboljem slučaju ne sadrži razlike koje bi ljudski slušni sistem percipirao. Novi način kodiranja je nazvan *percepcijsko audio kodiranje* jer se u potpunosti zasniva na subjektivnoj percepciji audio signala.

1.3.2 Kompresija sa gubiticima

U razvoju percepcijskog audio kodiranja trebalo je dakle iskoristiti nesavršenosti ljudskog slušnog sistema.

Prva znanstvena istraživanja na području ljudskog slušnog sistema, a vezana za audio kodiranje, bazirana su na istraživanjima obavljenim još sedamdesetih godina dvadesetog stoljeća na području kodiranja ljudskog govora. Već tada su uočene velike mogućnosti *neuniformne kvantizacije*: kod kodiranja govora jače je kvantizirano područje visoke energije

(izgovor samoglasnika) jer je visoko-energetski signal u potpunosti skrivao šum uveden kvantizacijom, tako da su za percepciju govora vrlo bitni dijelovi govora sa malenom energijom mogli biti bolje kodirani.

Kao direktna posljedica ovakvog načina razmišljanja o audio kompresiji uveden je i novi način evaluacije kvalitete komprimiranog audio signala. Više nije bilo ispravno kvalitetu kodiranog signala poistovjećivati sa omjerom signala i šuma jer se taj omjer primjenom novih načina kompresije negdje ciljano povećava, da bi se na drugim mjestima dobilo prostora za njegovo minimiziranje. Novi načini procjene kvalitete kodiranog signala su skoro pa isključivo subjektivne prirode. Nedavno su razvijene i prve metode objektivne evaluacije bazirane na simulacijama ljudskog slušnog sistema (Poglavlje 9.3.1).

Osnovna ideja audio kompresije sa gubitcima bila je ideja neuniformne kvantizacije u kombinaciji sa karakterističnostima ljudskog slušnog sistema. Kako kvantizacija u signal unosi šum, bilo je potrebno kvantizacijski šum prilagoditi svojstvima signala na način da šum u kombinaciji sa signalom ne bude čujan. Međutim, prvo je trebalo odgovoriti na mnoga iskrsla pitanja. Koje signale jače kvantizirati? Kako odrediti dozvoljeni prag šuma kvantizacije? Da li kvantizirati u vremenskoj ili frekvencijskoj domeni? Koja frekvencijska područja su najbitnija za ljudsku percepciju složenog audio signala?

1.4 Zaključak

Sva gore postavljena pitanja zajedno su inicirala ponovno aktualiziranje vrlo bitne grane u izučavanju audio kodiranja, nazvane *psihoakustika*. Za potpunije shvaćanje osnovnih psihoakustičkih principa potrebno je najprije upoznati građu i način rada ljudskog slušnog sistema, što je tema idućeg poglavlja.

Poglavlje 2

2 Ljudski slušni sistem

2.1 Uvod

Čovjekov sluh je kroz evoluciju bio od presudnog značaja za njegov opstanak. Spektar zvukova koje je čovjek morao čuti bio je vrlo širok: od šuštanja lišća koje je značilo blizinu neprijatelja ili lovine, pa sve do udara groma ili udaljenog huka reptila što je značilo opasnost i alarm za traženje zaklona. Čuti je značilo preživjeti.

Uho svojim slušnim mogućnostima impresionira. Raspon u intenzitetu zvuka koje čovjek čuje je $1:10^{12}$, dok je frekvencijski raspon čak 10 oktava (20 Hz do 20 kHz).

Rad uha kao organa za slušanje je do danas prilično detaljno razjašnjen, ali treba imati u vidu činjenicu da nije samo uho odgovorno za stvaranje doživljaja slušanja. Uho je prijemnik zvuka koji mehaničku energiju zvučnih valova (longitudinalno titranje molekula zraka) pretvara u električne impulse. Električki impulsi dalje odlaze u mozak, gdje ih centri za sluh dekodiraju – tek se tamo stvara doživljaj zvuka. Mozak je zadnja stanica na slušnom putu, a što se u njemu događa još točno ne znamo. Zato na samom početku treba naglasiti činjenicu da sve karakteristike ljudskog slušnog sistema ne mogu biti razjašnjene samo kroz opis rada samog uha, pa će i nakon ovog izlaganja ostati nejasne. Ipak, za shvaćanje osnovnih principa percepcijskog audio kodiranja, upoznavanje građe i načina rada samog ljudskog uha sasvim je dovoljno.

2.2 Građa uha

Kao što smo rekli, uho je prijemnik zvuka. Međutim, osim što je prijemnik, uho je i frekvencijski analizator zvuka, određuje smjer iz kojeg zvuk dolazi, osjeća glasnoću, visinu i

boju tona. Sve ove karakteristike uha na neki način su povezane sa audio kodiranjem, a neke, kao na primjer frekvencijska selektivnost, i od presudnog su značaja.

Prije pojašnjenja načina na koje uho obavlja spomenute funkcije, razmotrimo samu fizičku građu uha.

U osnovi, uho se dijeli na tri glavna dijela: vanjsko, srednje i unutrašnje uho (Slika 2.1).

2.2.1 Vanjsko uho

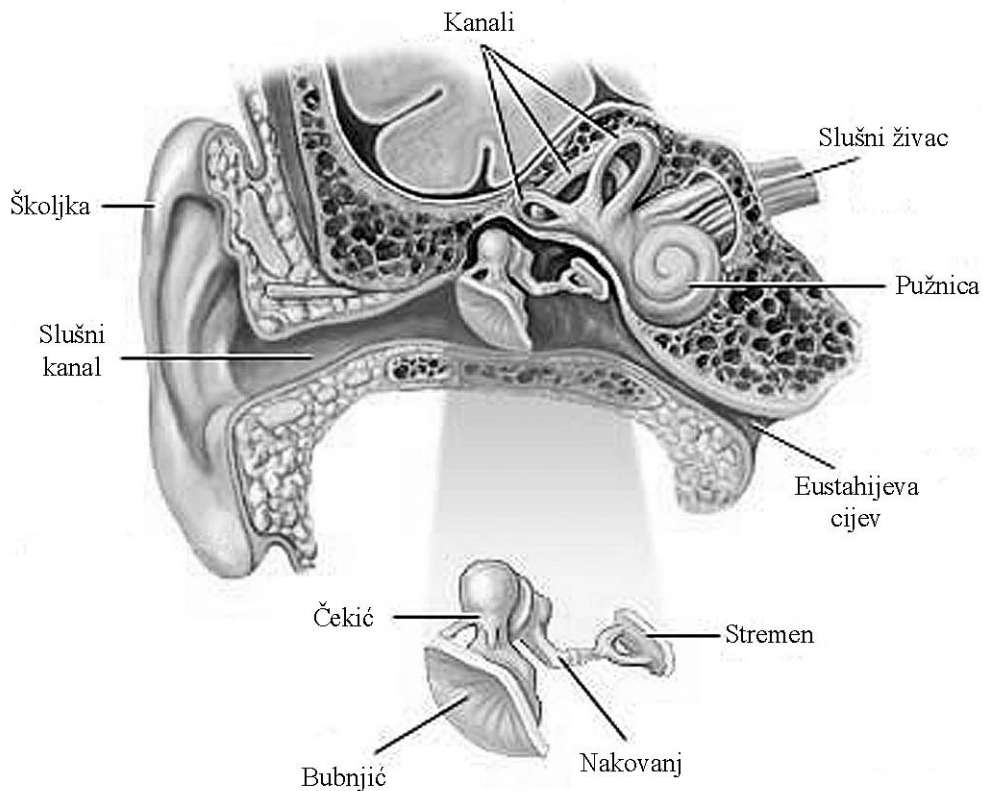
Vanjsko uho se sastoji od školjke, slušnog kanala i bubnjića. Školjka i kanal se ponašaju kao svojevrsan zvučni lijevak koji usmjerava i pojačava zvuk na njegovom putu prema bubnjiću. Kod nekih životinja školjka se može pomicati i tako pojačavati zvuk iz određenog smjera. Školjka ima utjecaj samo na zvuk valne duljine usporedljive sa njom samom (kod čovjeka iznad 3000 Hz). Bubnjić je membrana, vrlo osjetljiva na promjene tlaka – može registrirati tlak od čak $0,00002 \text{ W/m}^2$. Kao graničnik između vanjskog i srednjeg uha bubnjić predstavlja prijemnik zvučnih valova.

2.2.2 Srednje uho

Srednje uho se sastoji od Eustahijeve cijevi i slušnih košćica: čekića, nakovnja i stremena. Eustahijeva cijev služi samo za izjednačavanje tlaka unutar i izvan uha. Važniji dio srednjeg uha su slušne košćice koje čine vrlo efikasan sistem za prijenos zvučne energije sa bubnjića prema unutrašnjem uhu. Direktno sa bubnjićem je povezana prva slušna košćica – čekić. Pomak bubnjića prenosi se na čekić, a sa čekića na nakovanj i stremen. Sve tri košćice čine polužni sistem kojem je omjer prijenosa otprilike 3 naprema 2. Još veći omjer tlaka na bubnjiću prema onome prenesenom prema unutrašnjem uhu postiže se i omjerom površina bubnjića i ovalnog prozorčića na koji je pričvršćen – taj omjer je 1 naprema 20.

U središnjem uhu se nalazi i mehanizam za zaštitu uha od prevelike glasnoće. Dva mišića (tympani i stapedius) efikasno reguliraju glasnoću. Prvi mišić se na velikim glasnoćama refleksno steže, povlačeći vrh stremena u stranu tako smanjujući površinu kojom pločica stremena tlači tekućinu unutrašnjeg uha. Drugi mišić reagira na još većim glasnoćama, zatežući bubnjić i tako smanjujući mogućnost njegova vibriranja.

Slika 2.1: Uho se može podijeliti na tri dijela: vanjsko, srednje i unutrašnje uho. Najbitniji dio za audio kodiranje je unutrašnje uho, gdje se na bazilarnoj membrani vrši frekvencijska (i druga) analiza pristiglog zvučnog signala. Nakon analize, dobivena informacija šalje se preko zvučnih živaca u mozak – u mozgu se zvuk *doživljava*.



2.2.3 Unutrašnje uho

Unutrašnje uho je najvažniji dio uha obzirom na shvaćanje psihoakustičkih principa koji se koriste u audio kodiranju.

Unutrašnje uho je u cijelosti ispunjeno tekućinom u kojoj se nalaze predvorje, kanali i pužnica. Kanali služe za reguliranje ravnoteže pa su za doživljaj slušanja nebitni. Glavni dio unutrašnjeg uha je pužnica. Pužnica ima vrlo složenu građu. Po obliku je cjevasta i savijena, a uzduž nje se protežu tri kanala međusobno odvojena sa dvije membrane: Reisenerovom i bazilarnom. Iznad bazilarne membrane je Cortijev organ koji sadrži mnoštvo osjetnih stanica iz kojih strše dlačice.

2.3 Slušni proces

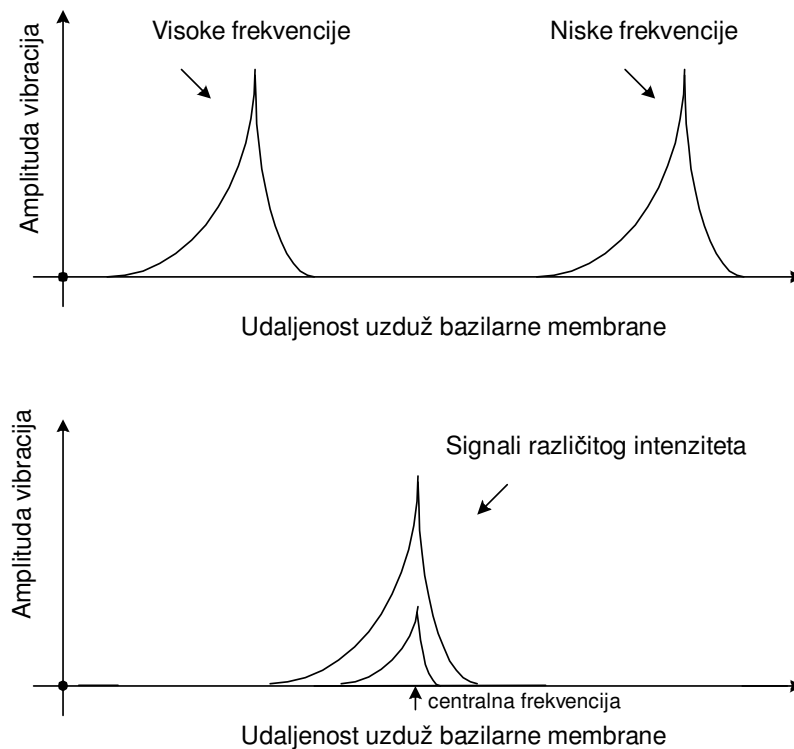
Tok slušnog procesa se odvija na sljedeći način: zvučni valovi dolaze na bubnjić koji se pod valnim tlakom uvija. Ušne koščiце prenose tlak kroz srednje prema unutrašnjem uhu dodatno ga pojačavajući. Preko srednjeg uha mehanička energija se predaje tekućini unutrašnjeg uha. Talasanje tekućine ima za posljedicu titranje bazilarne membrane. Kako je bazilarna membrana gipka, na njoj se stvara putujući val koji se na određeni način prostire u širinu i visinu. Val na bazilarnoj membrani dodiruje dlačice osjetnih stanica na Cortijevu organu. Obzirom na amplitudu i mjesto gdje putujući val doseže svoj maksimum, odnosno na to koje dlačice su i koliko jako podražene, slušni mehanizam stvara sliku frekvencijske karakteristike i jačine signala. Bazilarna membrana je na određeni način mehaničko-neuralni spektralni analizator zvuka. Ako je podražaj dlačica dovoljno jak, osjetne stanice šalju električne signale preko snopa slušnih živaca u mozak. Mozak dalje dekodira dobivene impulse pa se tako stvara doživljaj slušanja.

2.4 Osobine slušnog sistema

2.4.1 Karakteristična frekvencija

Zadržimo se još malo na pojašnjavanju karakteristika bazilarne membrane. Stimuliramo li uho sa tonom niske frekvencije, bazilarna membrana vibrira sve do mjesta gdje formirani putujući val nestaje. Amplituda vibracija ima vrlo oštar vrh. Stimuliramo li uho sa tonom visoke frekvencije, bazilarna membrana vibrira, a putujući val nestaje bliže početku bazilarne membrane nego što je to slučaj sa nisko-frekvencijskim tonom. Mjesto na membrani na kojem se nalazi maksimum amplitude vibracija određuje frekvenciju koja se naziva *karakteristična frekvencija* (Slika 2.2). Pojam karakteristične frekvencije je čvrsto povezan sa većinom psihoakustičkih principa koji se koriste u audio kodiranju.

Slika 2.2: Putujući val na bazilarnoj membrani doseže svoj maksimum to dalje od početka bazilarne membrane što je frekvencija zvuka niža. Dio bazilarne membrane koji oscilira dodiruje osjetne dlačice Cortijevog organa koji dalje prenosi signal prema mozgu. Amplituda vibracija bazilarne membrane proporcionalna je intenzitetu zvučnog signala.



2.4.2 Nelinearnost slušnog sistema

Uho ima jako izraženu nelinearnost odziva bazilarne membrane. Nelinearnost se najbolje ilustrira sljedećim primjerima:

- udvostručenje podražaja na ulazu u uho ne dovodi do dvostruko jače reakcije bazilarne membrane,
- doda li se na jedan ton, drugi različite frekvencije, intenzitet reakcije na prvi ton će se smanjiti,
- sastoji li se zvuk od dva tona, treći ton se može pojaviti u percepciji, a da njegov fizički izvor ne postoji,
- jači ton može maskirati ton slabijeg intenziteta, ako su dovoljno jedan blizu drugoga na bazilarnoj membrani (bliskih su frekvencija).

Nelinearnosti dodatno kompliciraju razumijevanje ljudskog slušnog sistema, i treba ih sa posebnom pažnjom uzeti u obzir kod analize algoritama za percepcijsko audio kodiranje.

2.4.3 Binauralno slušanje

Sve dosad opisane karakteristike uha odnosile su se na monoauralno slušanje – slušanje jednim uhom. Kod binauralnog slušanja, slušanja sa dva uha istovremeno, pojavljuju se dodatni efekti koje treba uzeti u obzir kod audio kodiranja.

Prvo pitanje koje se postavlja je kako binauralno lokaliziramo izvor zvuka? Prema prvoj i najstarijoj teoriji koju je predstavio Lord Rayleigh* 1778. godine, za binauralno lokaliziranje uho koristi sljedeća dva faktora:

- razliku u glasnoći zbog zasjenjenja,
- razliku u fazi jer jedno uho prima ton sa drugačijom fazom nego drugo.

Razlika u glasnoći za lokalizaciju je korisna na frekvencijama iznad 1500 Hz. Na ovim frekvencijama glava je mnogo veća od valnih dužina zvuka, pa se većina energije zvuka reflektira. Ako znamo da je brzina zvuka 343 m/s, razmak između lijevog i desnog uha prosječno 17.5 cm, slijedi da valnu dužinu ekvivalentnu širini glave ima zvuk frekvencije 1960 Hz. Uzmimo za primjer da je izvor zvuka pomaknut za 90° u stranu. Zvuk frekvencije 3 kHz biti će prigušen za 10 dB, zvuk od 6 kHz 20 dB, a zvuk od 10 kHz čak 30 dB, gleda li se percipirani intenzitet lijevog i desnog uha. Na sasvim niskim frekvencijama valne dužine su mnogo veće od dimenzija glava, pa glava u stvari i nije zapreka širenju valova tako da razlike u percepciji intenziteta nema.

Lokaliziranje na osnovu fazne informacije dominira na frekvencijama ispod 1500 Hz. Na višim frekvencijama fazna razlika je veća od 180° pa lokaliziranje nije moguće.

Posljednja istraživanja su pokazala da se lokaliziranje zvuka ne može objasniti samo na osnovu gornje dvije pojave. Naprotiv, utvrđeno je da na osnovu razlike u intenzitetu i fazi signala, uho može odrediti samo koliko je izvor zvuka pomaknut ulijevo ili udesno, dok je uho sasvim nemoćno pri utvrđivanju da li zvuk dolazi odozgo, odozdo, sprijeda ili otraga.

* John William Strut (Lord Rayleigh), 1842 – 1919, engleski znanstvenik, dobitnik Nobelove nagrade za fiziku 1904. godine, autor vrlo značajnog djela "Theory of Sound".

Rayleighjeva teorija nije bila dostatna jer ona nije uzimala u obzir kako ljudsko tijelo utječe na zvučne valove koji se na njemu lome i odbijaju, kako sobna konfiguracija također ima jak utjecaj na mogućnosti lokalizacije, kako pomak ljudske glave doprinosi poboljšanju lokalizacije, itd.

Danas se jednim od glavnih načina opisa lokalizacije smatra HRTF funkcija (*engl.* Head Related Transfer Function). Pomoću nje se opisuje utjecaj glave, torza, ramena i ušne školjke na promjenu zvuka koji iz izvora stiže prema dva uha, a teoretski ona sadrži potpunu informaciju o udaljenosti i smjeru izvora zvuka.

Dokazano je da su razlika u glasnoći i fazi ovisne o frekvenciji zvuka. Jedno od glavnih otkrića je činjenica da za lokalizaciju uho koristi efekt amplitudne modulacije: na niskim frekvencijama lokalizacija se zasniva na faznim razlikama, dok je na frekvencijama višim od 1500 Hz lokalizacija moguća na osnovu razlika u envelopama signala. Lokalizacija se odvija na način da uho razlaže signal na spektralne komponente, a zatim analizira envelope svih komponenti.

Poseban slučaj su impulsni zvukovi, koje čovjek lokalizira na osnovu vremenske razlike kojom zvuk stiže u dva uha. Uho je jako osjetljivo na vremensku razliku: vremenska razlika već od 30 mikrosekundi za uho znači da je izvor pomaknut za 3° u stranu, dok kašnjenje od 600 mikrosekundi znači pomak od čak 90° .

Realan je zaključak da sve gore spomenute karakteristike uha ne isključuju jedna drugu, nego da se one međusobno nadopunjuju pri procesu lokalizacije izvora zvuka. Općenito, uho najbolje lokalizira složene zvukove jer nastaje tzv. *kombinirano lokaliziranje*.

Iz ovih razmatranja je jasno da sasvim niske i sasvim visoke frekvencije najmanje doprinose procesu lokalizacije. Iz svakodnevnog iskustva je poznato da za zvučnik tipa *sub-woofer* nije bitno gdje ga se postavi – niski tonovi se jednako čuju bez obzira na njegovu poziciju, a lokalizacija je nemoguća. Također, zamislimo usamljen, čist ton, jako visoke frekvencije. Bez pomicanja glave lijevo-desno i njegova lokalizacija bila bi vrlo teška (iako moguća, što pokazuju HRTF karakteristike).

2.4.4 Efekt prvenstva

Kao bitna činjenica utvrđeno je da sam početak signala (prva valna fronta) uhu daje posebne informacije o poziciji zvučnog izvora. Testovi sa nizovima impulsa su pokazali da uho samo

iz prvog impulsa uzima informaciju o smjeru iz kojeg zvuk dolazi. Ovaj efekt se naziva efekt prvenstva (engl. *precedence effect*).

2.4.5 Efekt zabave

Efekt *zabave* (engl. *cocktail party effect*) je pojava koja je dobila ime po čestoj situaciji u kojoj više ljudi razgovara, a slušatelj se može koncentrirati na samo jednu osobu. Ova pojava nije moguća ukoliko se ne sluša binauralno, tj. sa oba uha. Pojava se objašnjava principom maskiranja: ako su i maskirajući i maskirani signali sa istog mjesta u prostoru, dolazi do potpunog maskiranja. Slušajući binauralno, bolje se određuje položaj maskirajućeg i maskiranog zvuka pa se na osnovu te informacije o položaju moguće koncentrirati na jedan izvor zvuka.

2.4.6 BMLD efekt

Neka je slabiji ton B maskiran jačim tonom A i neka su tonovi u istom trenutku stigli u oba uha. Ton B više neće biti maskiran tonom A u sljedeća dva slučaja:

- 1) promijenjena je faza jednog od dva signala,
- 2) maskirani ton B je poslan na oba uha, a glavni signal A je poslan na samo jedno uho.

U ovim slučajevima, da bi signal A ponovo maskirao signal B, njegov intenzitet mora biti povećan. Razlika između novog i starog intenziteta naziva se *binauralna razlika nivoa maskiranja* (engl. *Binaural Masking Level Difference – BMLD*). Vrijednost BMLD intenziteta je otprilike 15 dB za niske frekvencije do oko 500 Hz, i 2 do 3 dB za više frekvencije od oko 1500 Hz.

2.5 Zaključak

U ovom poglavlju je dan kratak pregled građe ljudskog slušnog sistema, nakon čega je opisan i sam proces slušanja. Naglašene su neke od osnovnih karakteristika ljudskog slušnog sistema na kojima počiva psihoakustika. Sami principi psihoakustike tema su sljedećeg poglavlja.

Poglavlje 3

3 Osnove psihoakustike

3.1 Uvod

Psihoakustika je područje akustike koje istražuje mogućnosti iskorištavanja pojedinih karakteristika ljudskog slušnog sistema za poboljšanje i unapređenje algoritama percepcijskog audio kodiranja, s osnovnim ciljem dobivanja što veće kompresije uz što manji gubitak kvalitete kodiranog signala.

Glavni psihoakustički principi koji se intenzivno koriste u audio kodiranju i koji će biti pojašnjeni u ovom poglavlju su: apsolutni prag čujnosti, kritični pojasevi maskiranja, simultano maskiranje, širenje maskiranja preko bazilarne membrane i vremensko maskiranje.

3.2 Apsolutni prag čujnosti

Apsolutni prag čujnosti definira najmanji intenzitet zvuka određene frekvencije koju ljudsko uho čuje u bešumnoj okolini.

Kao što je rečeno, najniži zvučni tlak koji prosječno ljudsko uho može registrirati iznosi 0,00002 Pa. Međutim, ovako nizak tlak uho ne osjeća na svim frekvencijama, nego samo na frekvencijama gdje je najosjetljivije (približno 1 kHz). Na drugim frekvencijama prag čujnosti je sve veći što su frekvencije bliže donjem rubu od 20 Hz i gornjem rubu od 20 kHz. Kompletna krivulja koja prikazuje intenzitet praga čujnosti preko cijelog frekvencijskog područja naziva se *krivulja apsolutnog praga čujnosti* ili *krivulja praga čujnosti u tišini*.

Krivulja apsolutnog praga čujnosti je vrlo bitna u psihoakustičkom audio kodiranju – frekvencijske komponente kodiranog signala čiji intenzitet pada ispod krivulje praga čujnosti mogu se zanemariti i izbaciti iz signala bez bojazni da će kvaliteta signala biti degradirana – spomenute frekvencije uho ionako ne percipira. Također, i svaki šum uveden u signal neće

biti čujan ako je njegov intenzitet ispod praga čujnosti. Razumijevanje ove činjenice predstavlja prvi korak prema razumijevanju percepcijskog audio kodiranja uopće.

Krivulja praga čujnosti određuje se eksperimentalno i dosta se razlikuje obzirom na uvjete i način vođenja eksperimenta, a također i na starosnu dob i očuvanost/oštećenost sluha testiranih osoba. Općenito, utvrđeno je da je prag čujnosti najniži kod mlađih osoba, starosti do oko 30 godina, a da se starenjem prag čujnosti podiže prema višim intenzitetima. Također, starenjem se smanjuje i raspon frekvencija koje osoba može registrirati. Tako osobe starije od 60 godina rijetko čuju frekvencije više od 15 kHz, iako su u mladosti čule i zvukove frekvencija 20 kHz.

Formula 3.1 predstavlja nelinearnu funkciju koja dobro aproksimira krivulju praga čujnosti.

$$T(f) = 3.64 \left(\frac{f}{1000} \right)^{-0.8} - 6.5 e^{-0.6 \left(\frac{f}{1000} - 3.3 \right)^2} + 10^{-3} \left(\frac{f}{1000} \right)^4 \quad (3.1)$$

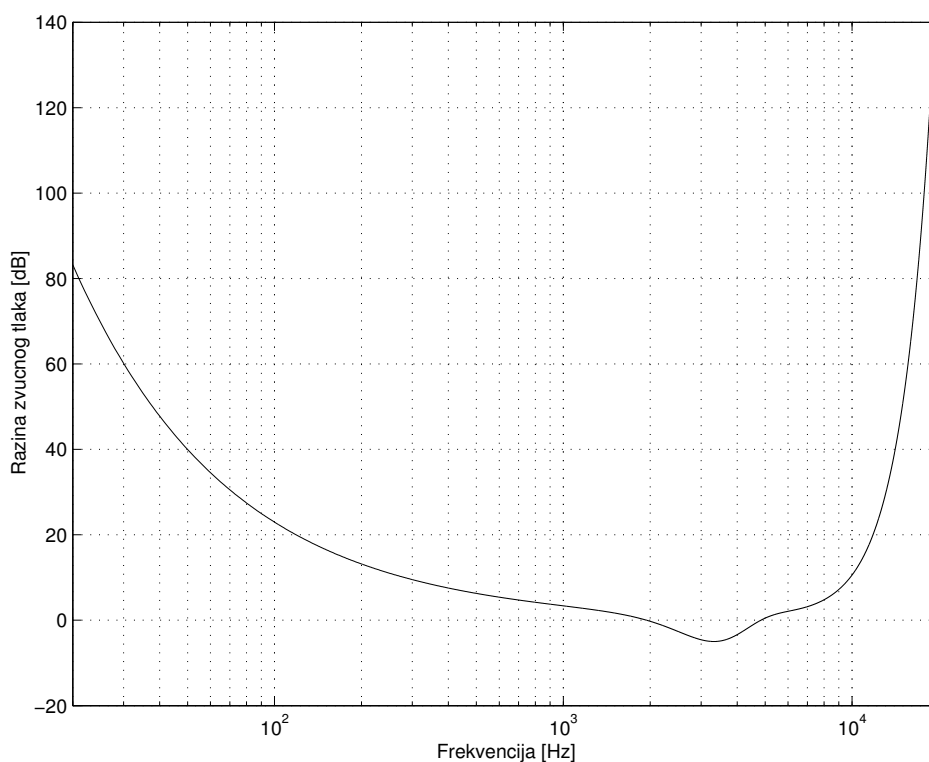
Treba imati na umu da dno krivulje leži na frekvenciji na kojoj je uho najosjetljivije, a da su ostali pragovi čujnosti određeni relativno prema tom nivou. Referenca je zvučni tlak od 0,00002 Pa, što je ekvivalentno 0 dB.

3.3 Kritični pojasevi bazilarne membrane

Ljudsko uho ima ograničenu mogućnost razlučivanja frekvencija zvuka. Ono ne opaža bilo kakvu malenu *promjenu* frekvencije. Na temelju mjerenja je zaključeno da u cijelom čujnom intervalu uho može razlikovati oko 850 stupnjeva tonskih visina. Da bismo razumjeli zašto su ovi navodi točni, vratimo se na trenutak na osobine bazilarne membrane.

Kako je već rečeno, bazilarnu membranu možemo smatrati spektralnim analizatorom, što znači da ona na određeni način u sebi sadrži frekvencijske filtre. Kako je propusno područje tih zamišljenih pojasevo-propusnih filtara vrlo usko, ali ipak određene širine, jasno je da će više bliskih frekvencija izazvati pobudu na istom mjestu na bazilarnoj membrani. U tom slučaju mozak će za dvije bliske, ali različite frekvencije, dobiti istu informaciju pa ih neće percipirati različito.

Slika 3.1: Krivulja praga čujnosti prema Formuli 3.1. Ordinata krivulje je logaritamsko mjerilo zvučnog tlaka relativno prema tlaku od 0.00002 Pa. Na apscisi je logaritamski prikaz frekvencija čujnog područja do otprilike 20 kHz. Uho je najosjetljivije na frekvencijama od 1 kHz do 4 kHz.



Eksperimenti koji su pokazali postojanje tzv. *kritičnih pojaseva* među prvima su izveli Fletcher i Zwicker. Glavni zaključak eksperimenata je da ljudski slušni sistem percipira jednako:

- čisti ton od 1 kHz, i
- složeni uskopojasni signal centriran oko 1 kHz*,

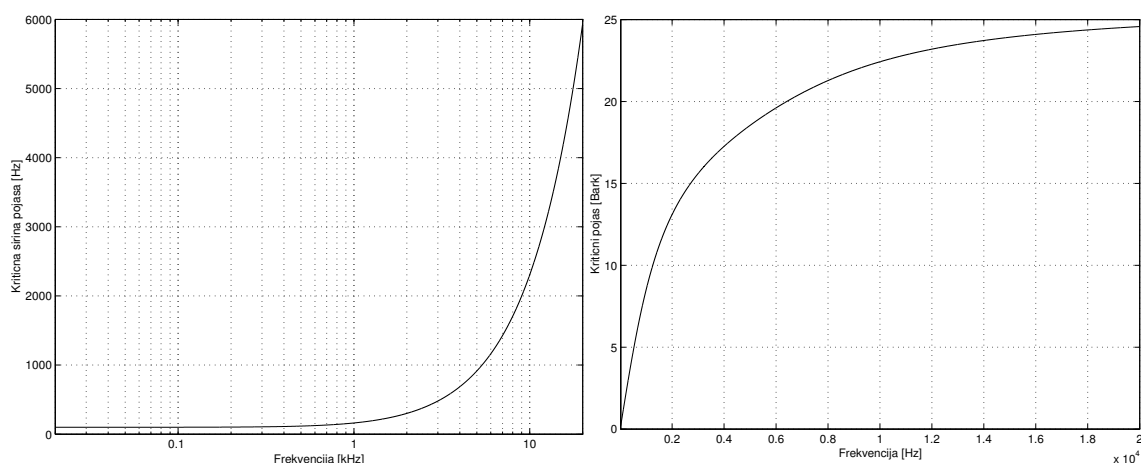
ako su zadovoljena sljedeća dva uvjeta:

- intenzitet dva signala je jednak,
- frekvenijski spektar uskopojasnog signala je unutar jednog kritičnog pojasa.

* Novija istraživanja pokazuju da je potrebno koncentrirati 70% energije unutar terce.

Eksperimenti su također pokazali da je širina ovih frekvencijskih pojaseva (filtara) oko 100 Hz na frekvencijama nižim od 500 Hz, dok se na frekvencijama višim od 500 Hz pokazuje relativno povećanje od oko 20% od jednog do drugog pojasa (Slika 3.2). Širina ovih frekvencijskih pojaseva se u praksi naziva *širina kritičnog pojasa*, a jedinica u čast znanstvenika Barkhausena* se naziva *Bark*.

Slika 3.2: *Lijevo*) Širina kritičnih pojaseva ovisno o frekvenciji. Širina kritičnih pojasa je konstantna do frekvencije od oko 500 Hz, dok se na višim frekvencijama naglo povećava (bazilarna membrana više nije tako dobro frekvencijski selektivna). *Desno*) Preslikavanje frekvencija u odgovarajuće kritične pojaseve. Cijeli čujni spektar se može preslikati u 24 kritična pojasa.



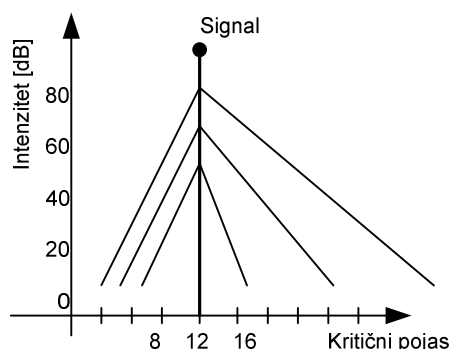
3.4 Simultano maskiranje

Pojava da je zvuk jedne frekvencije nečujan uz prisustvo druge frekvencije, a koja direktno proizlazi iz činjenice postojanja kritičnih pojaseva, naziva se *efekt simultanog maskiranja*. Efekt maskiranja također predstavlja jedan od osnovnih principa psihoakustike koji se koriste u audio kodiranju.

Ovisno o intenzitetu, najjača frekvencija signala koja postoji unutar kritičnog pojasa maskirati će neke od susjednih joj frekvencija. Slika 3.3 pokazuje maskiranje uskopojasnog signala centriranog na 2 kHz.

* Heinrich Georg Barkhausen, 1881 – 1956, njemački fizičar, prvi uveo logaritamsku skalu za mjerenje intenziteta zvuka.

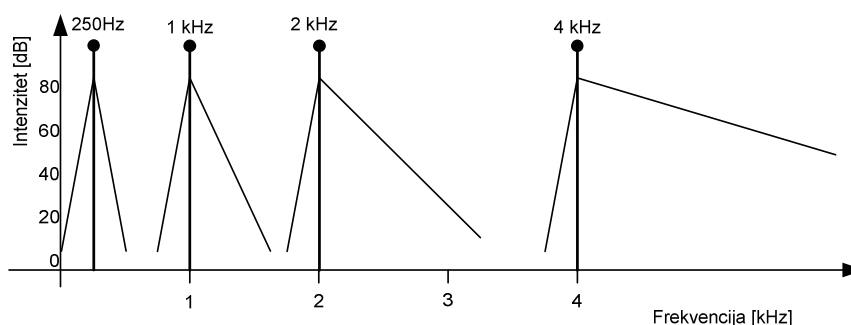
Slika 3.3: Maskiranje tonom od 2 kHz kod različitih intenziteta. Eksperimentalno je potvrđeno da krivulja maskiranja sa lijeve strane ima konstantan nagib od oko 27 dB/Bark, dok se nagib desnog dijela krivulje mijenja sa intenzitetom signala koji maskira, i iznosi otprilike 20 dB/Bark na niskim intenzitetima, a 5 dB/Bark na visokim intenzitetima.



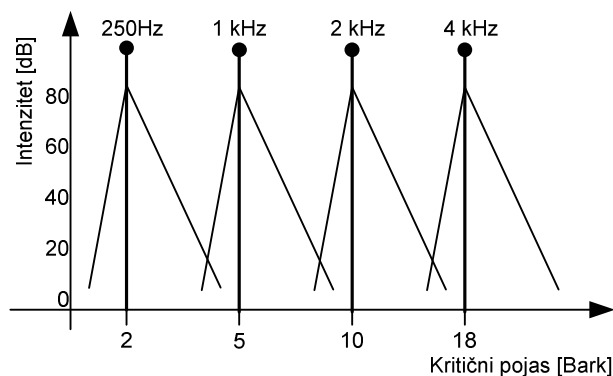
Dodatni eksperimenti su pokazali sljedeće osobine efekta maskiranja:

- na niskim intenzitetima do otprilike 30 dB, uskopojasni signal maskira simetrično i frekvencije niže i frekvencije više od sebe,
- uskopojasni signal većeg intenziteta od otprilike 40 dB pokazuje jednostavan paralelni pomak maskiranja na nižim frekvencijama, dok na višim frekvencijama maskiranje postaje sve jače,
- širokopojasni (bijeli) šum maskira jednolično frekvencije do 500 Hz, dok se na višim frekvencijama maskiranje pojačava u koracima od oko 10 dB po dekadi,
- čišći sinusni signali imaju slabiju moć maskiranja od spektralno složenih signala i šuma.

Slika 3.4: Maskiranje u frekvencijskoj domeni. Širina pojasa maskiranja proporcionalna je širini kritičnih pojasa. Na frekvenciji 250 Hz pojas maskiranja je širok oko 500 Hz dok se na frekvencijama 1 kHz, 2 kHz i 4 kHz pojasevi maskiranja znatno šire.

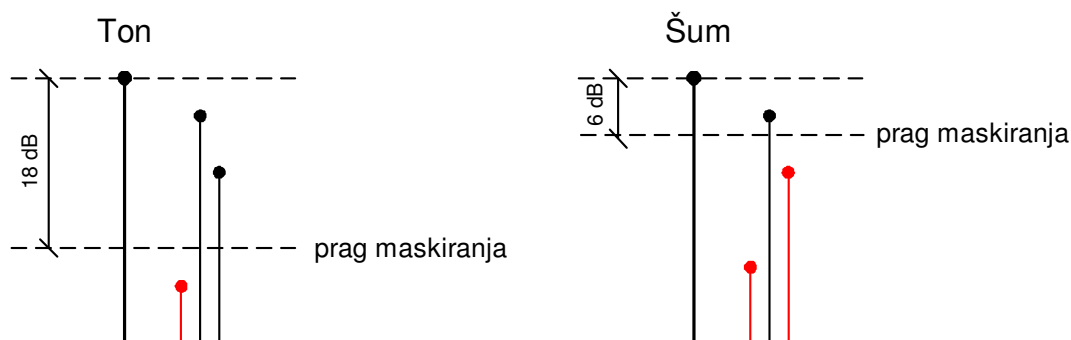


Slika 3.5: Maskiranje u Bark domeni. Širina pojasa maskiranja predstavljena u Bark domeni je konstantna (izuzmemo li rubna područja vrlo niskih i vrlo visokih frekvencija) pa je Bark domena vrlo pogodna za psihoakustičku analizu.



Iako je efekt maskiranja vrlo težak za egzaktno prikazivanje i proračun, jer su audio signali u pravilu vrlo složeni, ipak je za potrebe audio kodiranja napravljena podjela na dvije vrste maskiranja: maskiranje šuma tonom i maskiranje tona šumom. Eksperimentalno je utvrđeno da ton maskira približno 10 dB slabije od šuma, otprilike ispod nivoa udaljenog 15 do 20 dB ispod samog signala, a šum ispod 5 do 8 dB (Slika 3.6).

Slika 3.6: Razlika u maskiranju tonom i šumom. Ton i šum iste energije maskiraju različito.

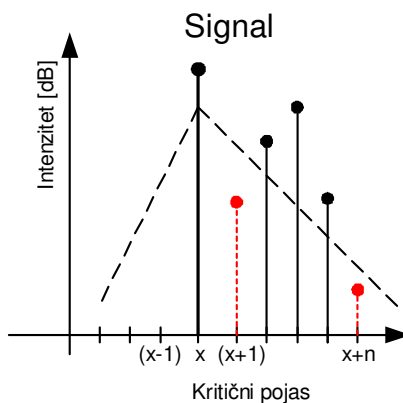


3.5 Širenje maskiranja

Efekt maskiranja nije prisutan samo unutar jednog kritičnog pojasa, nego i između susjednih kritičnih pojaseva. Ovakvo maskiranje se naziva *širenje maskiranja* i može se za potrebe audio kodiranja aproksimirati trokutastom funkcijom sa nagibom od 27 dB sa lijeve i 10 dB sa desne strane (po kritičnom pojasu). Kako su eksperimenti pokazali, maskiranje sa desne

strane krivulje ovisi o intenzitetu pa bi bilo poželjno i tu činjenicu uzeti u obzir kod aproksimacije krivulja maskiranja.

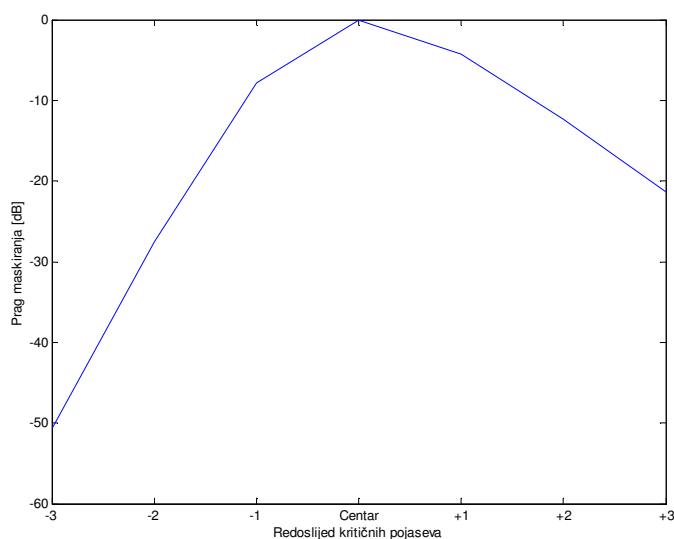
Slika 3.7: Širenje maskiranja preko više kritičnih pojaseva. Jedan kritični pojas može maskirati ne samo susjedni nego, ovisno o intenzitetu, i kritični pojas nekoliko puta udaljeniji.



Krivulja maskiranja dobro aproksimirana Formulom 3.2 je prikazana na Slici 3.8.

$$SF(x) = 15.81 + 7.5(x + 0.474) - 17.5\sqrt{1 + (x + 0.474)^2} \quad [\text{dB}] \quad (3.2)$$

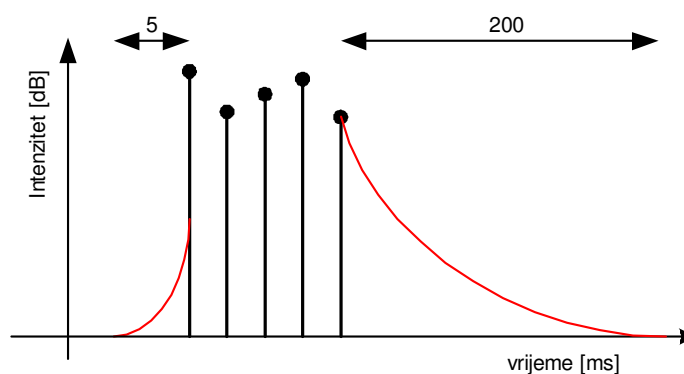
Slika 3.8: Krivulja maskiranja aproksimirana Formulom 3.2. Nedostatak je činjenica da formula ne uzima u obzir ovisnost krivulje maskiranja o intenzitetu maskirnog signala. Ipak u audio kodiranju su ovakve aproksimacije prihvatljive zbog brzine računanja (sve vrijednosti funkcije se izračunaju unaprijed) i jednostavnosti izvedbe.



3.6 Vremensko maskiranje

Kada nestane zvučne pobude u uhu, proći će još neko kratko vrijeme dok sve oscilacije na bazilarnoj membrani ne utihnu. Također, zbog tromosti sistema, trebati će i neko vrijeme nakon pojave pobude da se dobije odziv bazilarne membrane. Ova dva efekta zajedno čine efekt *vremenskog maskiranja*. Maskiranje prije pojave samog signala (engl. *pre-masking*) traje svega nekoliko milisekundi (~5 ms), dok maskiranje poslije završetka pobude (engl. *post-masking*) može, ovisno o intenzitetu pobude trajati i 100 do 200 ms. Efekt je prikazan na Slici 3.9.

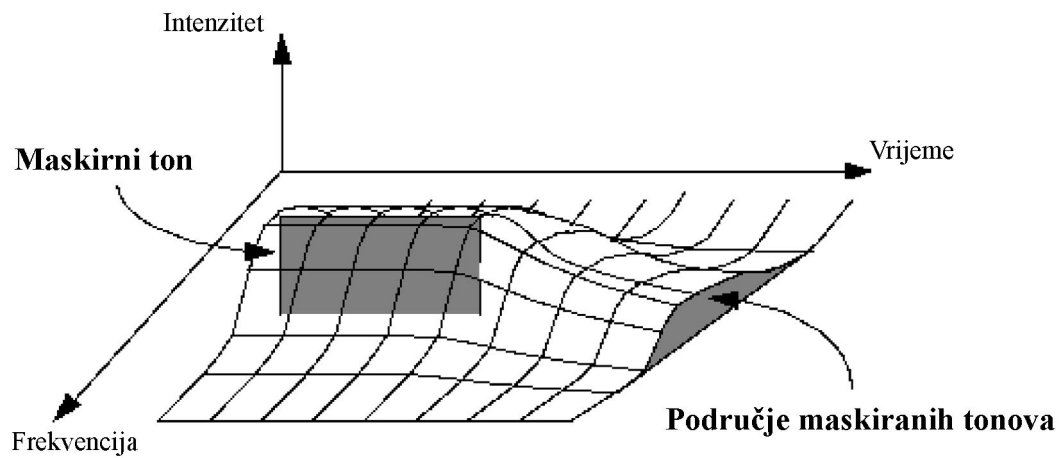
Slika 3.9: Vremensko maskiranje signala direktno ovisi o svojstvima bazilarne membrane. Maskiranje prije pojave signala traje oko 5 ms, dok maskiranje nakon kraja signala traje 100 do 200 ms.



3.7 Zaključak

Iako je uho gotovo savršen zvučni prijemnik, ipak se neke njegove karakteristike mogu iskoristiti u kodiranju zvuka: prag podražljivosti, rezolucija frekvencijskog analizatora, konačno vrijeme pobude i odziva, maskiranje, i dr.

Najbitnija karakteristika ljudskog slušnog sistema za audio kodiranje je maskiranje. Sveukupan efekt maskiranja je vizualno prikazan na Slici 3.10. Vidi se da jedan maskirni ton ima cijelo područje koje maskira i u frekvencijskoj i u vremenskoj domeni. Glavni zadatak audio kodiranja je kako za složeni audio signal precizno odrediti koji njegovi dijelovi pripadaju upravo tom području, a zatim ih tretirati kao irelevantne za percepciju.

Slika 3.10: Zbroj frekvencijskog i vremenskog maskiranja.

U sljedećim poglavljima ćemo vidjeti kako se karakteristike ljudskog slušnog sistema opisane u ovom poglavlju, primjenjuju u percepcijskom audio kodiranju, a nakon toga i u konkretnoj izvedbi audio koda prema danas najsofisticiranijem standardu za audio kodiranje ISO 14496-3 (MPEG-4 Audio).

Poglavlje 4

4 Percepcijsko audio kodiranje

4.1 Uvod

Ovo poglavlje donosi opis percepcijskog audio kodiranja kakvo je danas uobičajeno. Najprije je dan opis općeg* (izvornog, *generičkog*) percepcijskog audio koder. Njegova osnovna struktura je detaljnije pojašnjena. Težište je stavljeno na filtarske slogove, psihoakustički modul i modul za kvantizaciju. Također, dan je kratak pregled entropijskog koder. Nakon opisa ovih neizostavnih dijelova svakog percepcijskog koder objedinjenih u zamišljeni *opći* koder, dan je pregled povijesnog razvoja percepcijskog audio kodiranja, sa kraćim opisom danas standardiziranih i koder koji su u praksi našli veliku primjenu.

4.2 Opći audio koder

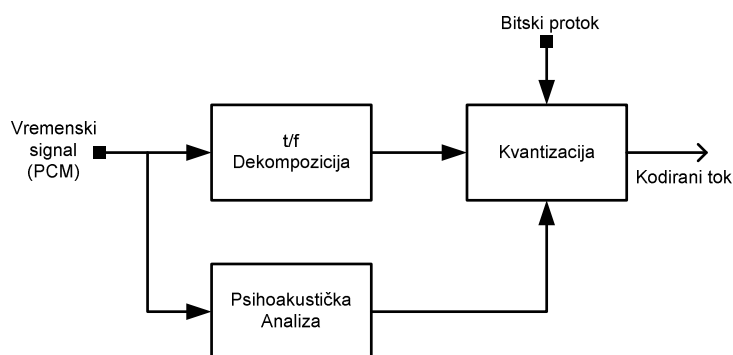
Tokom godina istraživačkog rada na razvoju percepcijskih audio koder, nametnula se donekle standardna arhitektura percepcijskog audio koder koja se ponavlja, sa malim razlikama od koder do koder. Sastavni dijelovi takvog percepcijskog audio koder su:

- vremensko-frekvencijska transformacija,
- psihoakustički modul,
- uobličavanje frekvencijskog spektra,
- kvantizacija,
- kompresija bez gubitaka i
- formiranje kodiranog toka.

* Ovdje ne treba pomiješati *općeniti* (engl. *General Audio*) MPEG-4 koder, čija izvedba je tema samog rada, sa *općim* koderom koji će biti predstavljen u ovom poglavlju.

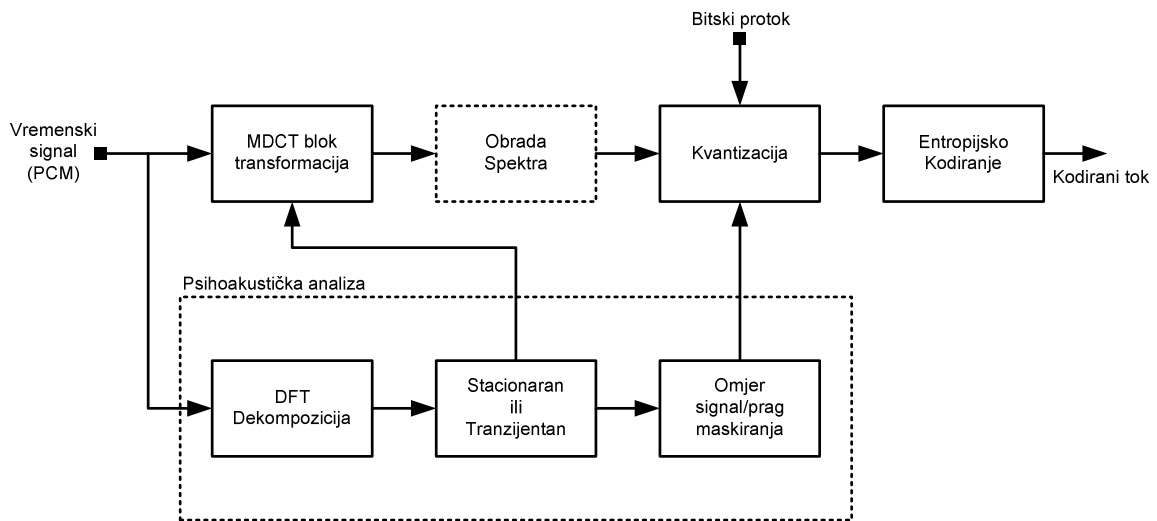
Tri bloka koja predstavljaju *osnovu* svakog percepcijskog audio koda su (Slika 4.1): blok za transformaciju signala iz vremenske u frekvencijsku domenu, blok za psihoakustičku analizu i blok za kvantizaciju. Ova tri bloka su dovoljna da bi se opisale osnove rada percepcijskog audio koda. Ulazni vremenski signal se prvim blokom prebacuje u frekvencijsku domenu. Paralelno se vrši i psihoakustička analiza signala. Na kraju se, na osnovu rezultata psihoakustičke analize, vrši kvantizacija frekvencijskih komponenti.

Slika 4.1: Osnovni sistem audio kodiranja se sastoji od 3 bloka: vremensko-frekvencijska transformacija, psihoakustički modul i kvantizacija. Ovaj model predstavlja osnovnu ideju audio kodiranja i sistem od kojega se počinje.



Ovakav pojednostavljen prikaz psihoakustičkog koda može se proširiti prema Slici 4.2 koja prikazuje *standardnu* arhitekturu percepcijskog audio koda. Dodani su blok za obradu spektra i blok za kompresiju bez gubitaka. Blok za obradu spektra se obično sastoji od više pod-blokova koji svaki na svoj način uobličavaju ulazni signal i tako doprinose postizanju veće kompresije. Blok za kompresiju bez gubitaka sve podatke potrebne za dekodiranje dodatno komprimira poznatim metodama kao što su aritmetičko ili Huffmanovo kodiranje.

Slika 4.2: Standardni sistem audio kodiranja: osim osnovnih, dodatni blokovi su obrada spektra (sadrži više pod-blokova) i blok za kompresiju bez gubitaka (entropijsko kodiranje).



Opišimo rad svakog od spomenutih blokova sa više detalja.

4.2.1 Vremensko-frekvencijska transformacija

Kao što smo vidjeli u Poglavlju 3, osobine ljudskog slušnog sistema lakše je sagledati i uspješno analizirati u frekvencijskoj domeni. Da bismo konstruirali prihvatljiv model audio kodiranja, najprije je potrebno audio signal predstaviti u frekvencijskoj domeni, dakle razložiti ga na frekvencijske komponente.

Idealna vremensko-frekvencijska transformacija imala bi sljedeća svojstva:

- prilagodljiva vremenska i frekvencijska rezolucija,
- idealna rekonstrukcija,
- jako gušenje u nepropusnom pojasu,
- bez *aliasinga*,
- kritično uzorkovanje i
- bez rubnog efekta.

Fina frekvencijska rezolucija je neophodna jer sve karakteristike ljudskog sluha vezane uz maskiranje mogu se kvalitetno analizirati samo u frekvencijskoj domeni. Najpoželjnija frekvencijska analiza bila bi ona identična kritičnim pojasevima ljudskog slušnog sistema, dakle neuniformna, međutim ovaj uvjet nije neophodan. Potrebno i dovoljno je da

frekvencijska rezolucija bude manja od širine najužeg kritičnog pojasa, a to je otprilike 100 Hz. Kako veća frekvencijska rezolucija implicira manju vremensku rezoluciju, treba biti vrlo pažljiv u odabiru i balansiranju ova dva parametra. Ipak, veći značaj treba dati frekvencijskoj rezoluciji, na štetu vremenske.

Uvjet idealne rekonstrukcije zahtjeva da signal rekonstruiran vraćanjem iz frekvencijske u vremensku domenu bude identičan početnom signalu, ukoliko spektar u međuvremenu nije modificiran.

Kritično uzorkovan signal daje jednak broj komponenti nakon transformacije, broju uzoraka signala prije transformacije. Ovaj uvjet je bitan zbog opterećenja procesa kodiranja, pa je poželjno da bude ispunjen. Dakle, broj frekvencijskih komponenti nakon transformacije ulaznog signala mora biti jednak broju uzoraka vremenskog signala na kome je transformacija izvedena.

Rubni efekt dovodi do pojave jakih izobličenja na rubovima transformacijskih blokova nakon rekonstrukcije signala. Ako rekonstrukcija nije sasvim idealna, dolazi do pojave diskontinuiteta na rubovima, a diskontinuiteti su jako čujni. Rubni efekt se eliminira upotrebom tzv. *prozora* (vremenskih otvora) i preklapanjem susjednih blokova.

Frekvencijska dekompozicija signala vrši se modulom koji se u audio kodiranju najčešće naziva *filtarski slog* (engl. *filterbank*). Ugrubo, filtarske slogove možemo podijeliti u podpojasne (engl. *subband*) i transformacijske (engl. *transform*). Podpojasni filtarski slog vremenski signal transformira u nekoliko posebnih frekvencijskih pojaseva čiji broj obično nije velik (npr. 32), dok transformacijski filtarski slog vremenski signal transformira u velik broj frekvencijskih komponenti (npr. 1024).

Također, moguća je i podjela filtarskih slogova na uniformne i neuniformne. Uniformni filtarski slog se sastoji od više pojasno-propusnih filtara jednake širine, koji prekrivaju cijelo željeno frekvencijsko područje. Nakon propuštanja ulaznog signala kroz svaki od podpojasnih filtara, dobivaju se frekvencijski pojasevi jednake širine. Neuniformni filtarski slog se sastoji od pojasno-propusnih filtara različite širine pojasa propuštanja. Omogućuju bolju frekvencijsku rezoluciju pojedinih dijelova signala, tako da se upotrebljavaju u sistemima sa posebnim zahtjevima, recimo kod nestacionarnih signala, tranzijenata itd. Logično je da se podjela čujnog spektra na kritične pojaseve bolje aproksimira neuniformnim filtrima. U praksi se međutim pokazalo da neuniformni filtri imaju veliki nedostatak: frekvencijska rezolucija im se smanjuje sa porastom frekvencije, što kod nekih signala može biti nenadoknativ nedostatak. Jedan od primjera neuniformnog filtarskog sloga je *wavelet* filtarski slog koji se

pokazao neučinkovit kod predstavljanja signala govora jer su za percepciju nekih samoglasnika vrlo bitna srednje-frekvencijska područja gdje wavelet transformacija ima nedovoljnu rezoluciju. Također, iako se širina kritičnih pojaseva povećava sa povećanjem frekvencija, to ne znači da ljudsko uho ima manju selektivnost na višim frekvencijama (što je i dokazano usporednim analizama spektralno intenzivnih signala i čistih sinus-signala) pa i ova činjenica mora biti povezana sa upotrebom neuniformnih filtarskih slogova.

Jedno od najvažnijih svojstava filtarskog sloga je njegova vremensko-frekvencijska rezolucija. Ilustrirajmo važnost ovog parametra na sljedećem primjeru: analizirajmo izrazito stacionaran signal instrumenta *pikolo* (engl. *piccolo*) i izrazito tranzijentan signal instrumenta *kastanjete* (engl. *castanets*).

Pikolo ima jako izražene i stalne harmonike, sam signal je dakle stacionaran. Maskiranje u frekvencijskoj domeni je, zbog istog razloga, koncentrirano na mjestima gdje su te komponente izražene, pa je stoga razumljivo da je kod ovog primjera od presudnog značaja frekvencijska rezolucija ulaznog signala.

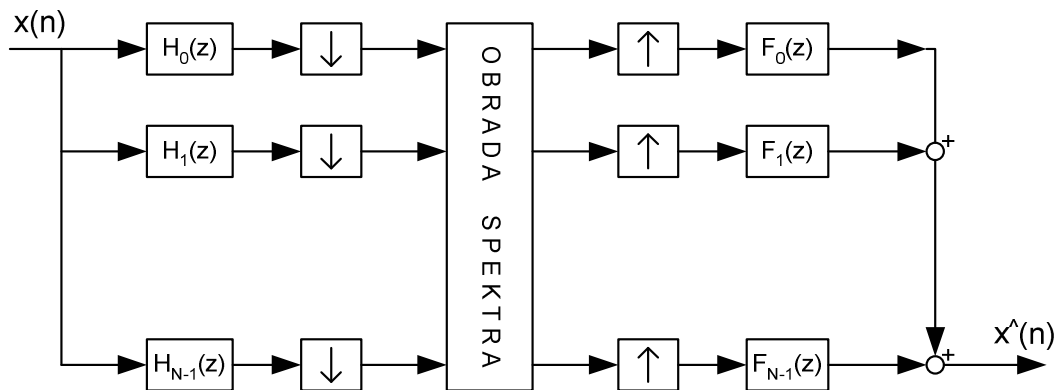
Kastanjete imaju jako izražene tranzijente, pa je maskiranje vrlo vremenski neuravnoteženo, pojavljuje se zajedno sa tranzijentima u nepravilnim vremenskim razmacima. Kako su tranzijenti i frekvencijski različiti odnosno nepredvidivi, bitno ih je lokalizirati u vremenu, prepoznati. Zaključujemo da je kod ovog primjera presudna dobra vremenska rezolucija.

Kako je većina audio signala kombinacija dva prikazana primjera, koji su prilično radikalni predstavnici dviju po zahtjevima prema filtarskom slogu oprečno različitih grupa signala, zaključujemo da bi idealni filtarski slog trebao imati mogućnost prilagođenja dužine bloka podataka ulaznog signala na kojem vrši vremensko-frekvencijsku transformaciju. U pravilu signali su većinom stacionarni, da bi nakon nekog vremena imali tranzijentne promjene prema nekakvom novom stacionarnom stanju. Filtarski slog bi trebao pratiti promjene signala i prema njemu se prilagođavati.

Većina percepcijskih audio koda danas ima mogućnost detekcije tranzijentnih signala, nakon čega se na određeni način filtarskom slogu signalizira prebacivanje u režim veće vremenske rezolucije.

Tipičan filtarski slog koji se upotrebljava u audio kodiranju je sastavljen je od paralelnog niza pojasno propusnih filtara normaliziranih centralnih frekvencija $\frac{(2k+1)\pi}{2N}$. N pojasno propusnih filtara zajedno pokriva cijeli ciljani frekvencijski pojas.

Slika 4.3: Uobičajeni prikaz audio filtarskog sloga sa N kanala. Na ulazu svakog kanala je filter za analizu. Nakon analize vrši se *uzorkovanje-na-dolje* da bi se postigao uvjet maksimalne decimacije. Nakon obrade spektra u audio koderu, pristupa se sintezi vremenskog signala. Najprije se signal *uzorkuje-na-gore*, da bi se nakon toga slao na niz izlaznih filtera za sintezu. Ako se preskoči obrada signala, ulazni i rekonstruirani signal morali bi biti isti kod filtera sa idealnom rekonstrukcijom.



Slika 4.3 prikazuje spomenuti filtarski slog kroz proces analize i sinteze kod kodiranja odnosno dekodiranja. Nakon što se ulazni signal razloži na N frekvencijskih pojaseva (analiza), vrši se decimacija, odnosno broj izlaznih uzoraka se svodi na broj ulaznih (*uzorkovanje-na-dolje*). Zatim se vrši uobličavanje spektra shodno psihoakustičkoj analizi, a na kraju kvantizacija i pakiranje u kodirani tok. Na strani dekodera proces je obrnut: vrši se raspakiravanje kodiranog toka, dekvantizacija, da bi frekvencijski uzorci zatim bili *uzorkovani-na-gore* i poslani na ulaz filtarskog sloga za sintezu. Kombinacijom izlaza pojedinih filtera za sintezu dobiva se rekonstruirani vremenski signal.

Svaki ulazni pojasno propusni FIR filter sa Slike 4.3 definiran je Z transformacijom (vrijedi za k -ti filter):

$$H_k(z) = \sum_{n=-\infty}^{+\infty} h_k(n) z^{-n} \quad (4.1)$$

Svaki izlazni pojasno propusni FIR filter sa Slike 4.3 definiran je Z transformacijom (vrijedi za k -ti filter):

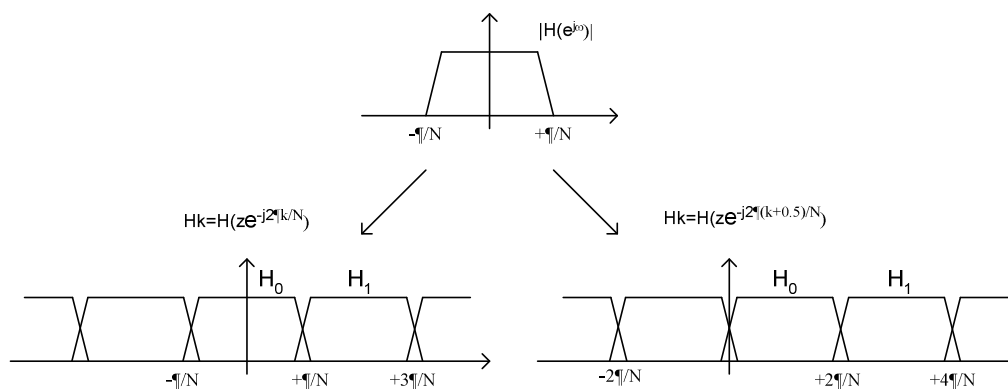
$$F_k(z) = \sum_{n=-\infty}^{+\infty} f_k(n) z^{-n} \quad (4.2)$$

Glavni zahtjev koji se postavlja na dizajn ovakvih filtara je idealna rekonstrukcija, kada vrijedi:

$$x(n) = \hat{x}(n) \quad (4.3)$$

Na Slici 4.4 prikazana su dva načina modulacije prototipa FIR filtra.

Slika 4.4: Dva načina modulacije prototip filtra $H(z)$. Lijevo je prikazan rezultat parne, a desno neparne modulacije. Kod parnog moduliranja, prvi pojas propuštanja je prepolovljen, pa se kod audio kodiranja koristi neparna modulacija.



Danas su u audio kodiranju glavno mjesto zauzela dva filtarska sloga: prvi se naziva PQMF (engl. *Pseudo Quadrature Mirror Filterbank*), a drugi MDCT (engl. *Modified Discrete Cosine Transform*) filtarski slog.

Unutar MPEG-1 i MPEG-2 audio koda koristi se hibridni filtarski slog koji se sastoji od kombinacije PQMF i MDCT filtarskih slogova, dok se u MPEG-4 standardu uglavnom koristi MDCT filtarski slog iako se u određenim slučajevima koristi i PQMF. Detaljnije o ova dva filtarska sloga biti će govora u opisu MPEG-4 audio standarda.

4.2.2 Psihoakustički modul

Svaki percepcijski audio koder je zasnovan na psihoakustičkoj analizi ulaznog signala. Može se slobodno reći da je psihoakustički modul *srce* audio koda. Njegov cilj je određivanje karakteristika ulaznog signala kao što su tonalnost i tranzijentnost, zatim određivanje dijelova ulaznog signala koji su najbitniji za percepciju, a koji su manje bitni ili zanemarivi, određivanje stupnja maskiranja kojim signali djeluju jedni na druge, itd.

Odmah treba napomenuti da psihoakustički model nije i niti ne može biti standardiziran. Njegova izvedba neovisna o standardu je razumljiva: kada bi psihoakustički model bio standardiziran, svaki koder bi postizao iste rezultate obzirom na kvalitetu dekomodirano rekonstruiranog signala, pa bi različite izvedbe bile bespredmetne.

Pogledajmo način na koji psihoakustički modul funkcionira. Kao i svaki blok unutar audio kodera, i psihoakustički modul ima svoj ulaz i izlaz. Međutim, psihoakustički modul, za razliku od ostalih dijelova kodera, ne vrši nikakve promjene na ulaznom signalu, nego ga samo analizira, i na osnovu dobivenih rezultata, usmjerava rad drugih modula unutar kodera.

Ugrubo, ulaz u psihoakustički modul čine:

- trenutni blok vremenskih uzoraka ulaznog signala,
- zadnjih nekoliko blokova vremenskih uzoraka ulaznog signala,

Izlaz iz psihoakustičkog modula čine:

- omjer *signal/prag maskiranja* za svaki kritični pojas,
- tip prozora koji će se upotrebljavati kod filtarskog sloga obzirom na tonalnost signala,
- dužina prozora koja će se upotrijebiti kod analize u filtarskom slogu obzirom na stacionarnost ili tranzijentnost signala.

Iz trenutnog bloka vremenskih uzoraka određuje se dozvoljeni intenzitet maskiranja unutar svakog ciljanog frekvencijskog pojasa ulaznog bloka (omjer *signal/prag maskiranja*). Zadnjih nekoliko blokova su potrebni za određivanje tonaliteta (zbog predviđanja preko više blokova), a u nekim slučajevima i zbog određivanja tranzijentnosti signala.

4.2.2.1 Posebna vremensko-frekvencijska transformacija

Kako je u frekvencijskoj domeni razumljivije i lakše primjenjivati rezultate psihoakustičkih istraživanja svojstava ljudskog slušnog sistema, vremenski signal se prvo prebacuje u frekvencijsku domenu. Iako bi se na ovom mjestu mogla iskoristiti i transformacija u filtarskom slogu, psihoakustički modul u većini današnjih kodera ima svoju zasebnu vremensko-frekvencijsku transformaciju. U psihoakustičkoj analizi, najveći značaj je dan frekvencijskoj rezoluciji signala. Kako je transformacija koja se koristi u filtarskom slogu obično u tom pogledu loša jer su njeni prioriteti drugačiji, a ne daje niti nikakvu informaciju o fazi signala (koja je potrebna pri proračunu tonaliteta signala), njen rezultat se obično ne

upotrebljava u psihoakustičkom modulu. Psihoakustički modul u većini slučajeva upotrebljava diskretnu Fourierovu transformaciju za analizu vremenskog signala.

4.2.2.2 Funkcija širenja maskiranja

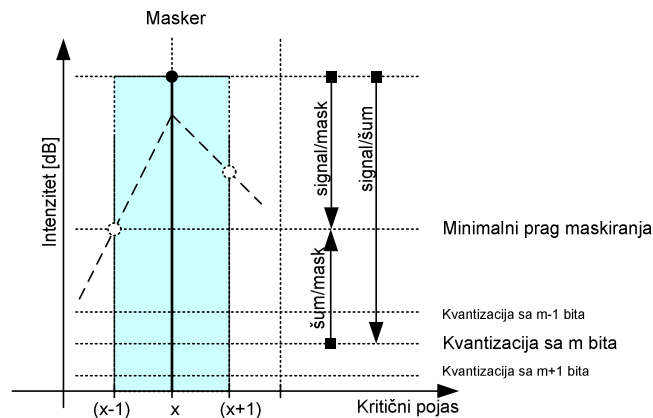
Psihoakustička analiza počinje podjelom čujnog spektra na niz pojaseva koji aproksimiraju kritične pojaseve ljudskog slušnog sistema. Nakon podjele, proračunava se energija unutar svakog pojasa (energija svake frekvencijske komponente proporcionalna kvadratu njenog intenziteta). Nakon proračuna energije, računa se intenzitet maskiranja kojim pojasevi uzajamno djeluju jedni na druge. Pri proračunu ovog fenomena, od presudne važnosti je funkcija *širenja maskiranja* (engl. *spreading function*). Ova funkcija određuje koliko koji pojas utječe na podizanje praga čujnosti susjednih pojaseva, obzirom na efekt maskiranja. Proračun kao svoj rezultat daje krivulju maskiranja koju određuju same energije maskiranja pojedinih kritičnih pojaseva. Na kraju se dobivena krivulja superponira na krivulju apsolutnog praga čujnosti, pa se iz ove dvije krivulje formira konačna krivulja praga čujnosti, što je i najbitniji rezultat psihoakustičke analize.

4.2.2.3 Rezultati psihoakustičke analize

4.2.2.3.1 Omjer *signal/prag maskiranja*

Glavni rezultat psihoakustičke analize je niz omjera *signal/prag maskiranja* koji određuju koliki šum može biti uveden u signal kvantizacijom, a da još uvijek ostane nečujan, odnosno da ostane maskiran samim signalom. Svakom kritičnom pojasu je pridijeljen jedan omjer. Niz omjera se prosljeđuje kvantizacijskom modulu koji svaki pojas kvantizira uzimajući u obzir zadani omjer, na način da šum uveden kvantizacijom ostane ispod praga čujnosti, tj. ispod krivulje maskiranja u spomenutom frekvencijskom (kritičnom) pojasu.

Slika 4.5: Minimalni prag maskiranja unutar jednog kritičnog pojasa je određen minimumom rezultante djelovanja susjednih pojaseva putem funkcije *širenja maskiranja*. Kvantiziramo li komponente signala sa m bita, rekonstrukcijom signala u pojas će biti unesen šum. Cilj percepcijskog audio kodiranja je držanje kvantizacijskog šuma ispod praga maskiranja. Kvantiziramo li signal sa manjim kvantizatorom ($m-1$ bita), unosimo više šuma, koji nakon prelaska praga maskiranja postaje čujan.



4.2.2.3.2 Tonalitet

Prije konačnog proračuna niza omjera *signal/prag maskiranja*, psihoakustički model mora izračunati i tonalitet svakog kritičnog pojasa ulaznog signala, obzirom da tonalni i netonalni signali imaju različita svojstva maskiranja. Postoji nekoliko metoda za proračun tonaliteta, a najpoznatije su: metoda lokalnih maksimuma, metoda mjere spektralne uravnoteženosti (engl. *spectral flatness measure* - *SFM*) i metoda predviđanja.

Metoda lokalnih maksimuma

Nakon DFT transformacije, unutar jednog bloka pronalaze se lokalni maksimumi frekvencijskih komponenti signala. Lokalnim maksimumom se smatra komponenta otprilike 5-10 dB iznad susjednih komponenti. Nakon određivanja lokalnih maksimuma, vrši se proces decimacije, odnosno uklanjanja susjednih maksimuma. Proces decimacije se vrši na sljedeći način: svaka komponenta lokalnog maksimuma koja je ispod apsolutnog praga čujnosti se eliminira, zatim se eliminira svaki lokalni maksimum ukoliko u njegovoj blizini manjoj od pola Barka postoji po intenzitetu jači lokalni maksimum. Nakon procesa decimacije, energije komponenti koje nisu svrstane u tonalne zbrajaju se međusobno i tako tvore jednu netonalnu komponentu signala po svakom kritičnom pojasu. Za frekvenciju proračunatih netonalnih komponenti se uzima frekvencija najbliža geometrijskoj sredini pripadajućeg kritičnog pojasa.

Metoda mjere spektralne uravnnjenosti

Metoda mjere spektralne uravnnjenosti proračunava omjer geometrijske i aritmetičke sredine frekvencijskih komponenti signala unutar kritičnog pojasa:

$$\alpha_p = \frac{\left[\prod_{k=x}^{x+d} f_k^2 \right]^{\frac{1}{d}}}{\frac{1}{d} \sum_{k=x}^{x+d} f_k^2} \quad (4.4)$$

gdje je p oznaka kritičnog pojasa, x je početna frekvencijska komponenta unutar kritičnog pojasa koji se sastoji od d komponenti. Johnson [51] [52] je predložio da referenca koja će određivati sasvim tonalan signal bude -60 dB, pa se konačni tonaliteta dobiva iz sljedeće formule:

$$\alpha_{p[dB]} = \min \left(\frac{10 \log_{10} \alpha_p}{-60dB}, 1 \right) \quad (4.5)$$

Svojstvo ovog proračuna je da svaka vrijednost tonaliteta pada u interval $[0,1]$. Vrijednosti bliže nuli označavaju netonalne signale, dok tonalni signali imaju vrijednosti bliže jedinici.

Predviđanje

Određivanje tonaliteta predviđanjem zahtijeva informaciju o fazi i amplitudi signala ne samo u tekućem bloku za koji se proračun obavlja, nego i za nekoliko prethodnih blokova, najčešće dva. Pri proračunu se upotrebljava jednostavno predviđanje faze i amplitude signala, nakon čega se uspoređi predviđena i stvarna vrijednost. Predvidljiviji signal je tonalniji dok je nepredvidiv signal šumniji.

4.2.2.3.3 Tranzijentnost

Treći zadatak psihoakustičke analize je svrstavanje signala u jednu od dvije karakteristične grupe obzirom na njegovu tranzijentnost: signal može biti ili stacionaran ili tranzijentan. Ovaj podatak je bitan obzirom da se kod tranzijentnih signala vrši smanjenje dužine bloka za analizu zbog pojave efekta *pred-jeke*.

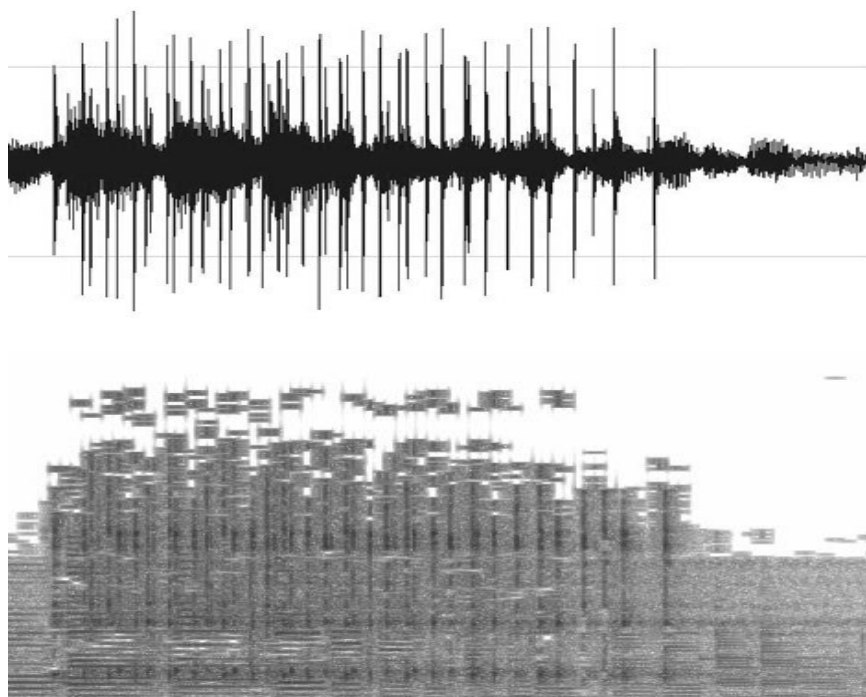
Detekcija tranzijenata analizom distribucije energije

Tranzijent se može detektirati analizom raspodjele energije signala koristeći svojstvo tranzijentnog dijela signala da ima veliku pozitivnu promjenu energije u vrlo malom vremenu. Algoritam može povećanje energije signala povezati sa istodobnom pojavom visokih frekvencija u signalu (jer je tranzijent u pravilu visoko-frekvencijski), pa tek u slučaju da su oba slučaja ispunjena, algoritam može signalizirati pojavu tranzijenta. Jedna od varijacija koju je vrijedno spomenuti je uzimanje u obzir samo energije visokih frekvencija, otprilike iznad 6 kHz, ili/i pridavanje veće težine energijama viših frekvencija.

Detekcija tranzijenta analizom envelope signala

Detekcija envelope signala se vrši u vremenskoj domeni – detektira se nagli porast envelope signala te se tada signalizira pojava tranzijenta. Prednost ove metode je jednostavna izvedba, a također i preciznost lociranja tranzijenta koja je u idealnom slučaju trenutna. Ova metoda zahtijeva implementaciju algoritma za izračunavanje envelope signala (npr. slijedilo maksimuma, engl. *peek follower*), a treba voditi računa i o veličini bloka uzoraka nad kojima se vrši analiza obzirom na opterećenost računanjem (2-5 ms). Problematična je i detekcija kraja tranzijentnog perioda.

Slika 4.6: Sonogram signala pokazuje više tranzijenata u nizu. Tranzijentni dijelovi imaju vrlo velik i nagao upliv visokih frekvencija. Nakon tranzijenata signal se *smiruje* i postaje stacionaran.



Detekcija tranzijenta analizom spektra

Ova metoda detekcije tranzijenta je donekle slična metodi analize distribucije energije, međutim ovdje se analiziraju razlike u spektru između susjednih blokova. Analiza se vrši duž cijelog spektra tako da se prisutnost tranzijenta može signalizirati iako nije lokalizirana u visoko-frekvencijskom području.

Detekcija tranzijenata proračunom percepcijske entropije

Izračun percepcijske entropije signala, čiji cilj je određivanja moguće mjere kompresije audio signala preko procjene relevantnosti komponenti audio signala za njegovu percepciju, koristi se i kao sredstvo za detekciju tranzijenata u signalu. Entropija signala kao *mjera nepravilnosti* signala detektira pojavu tranzijenta zbog sljedećih svojstava tranzijentnog signala:

- nagla pojava,
- unošenje znatne dodatne energije u signal,
- dolaska pretežito iz novog izvora zvuka pa tako izaziva sasvim novi spektralni sastav signala (onemogućava predviđanje),
- unošenje mnoštva novih, većinom visoko-frekvencijskih komponenti koje nisu bile prisutne u blokovima prije,
- unutar kratkog vremenskog intervala, pojavljuje se mnoštvo kaotičnih komponenti koje brzo nestaju.

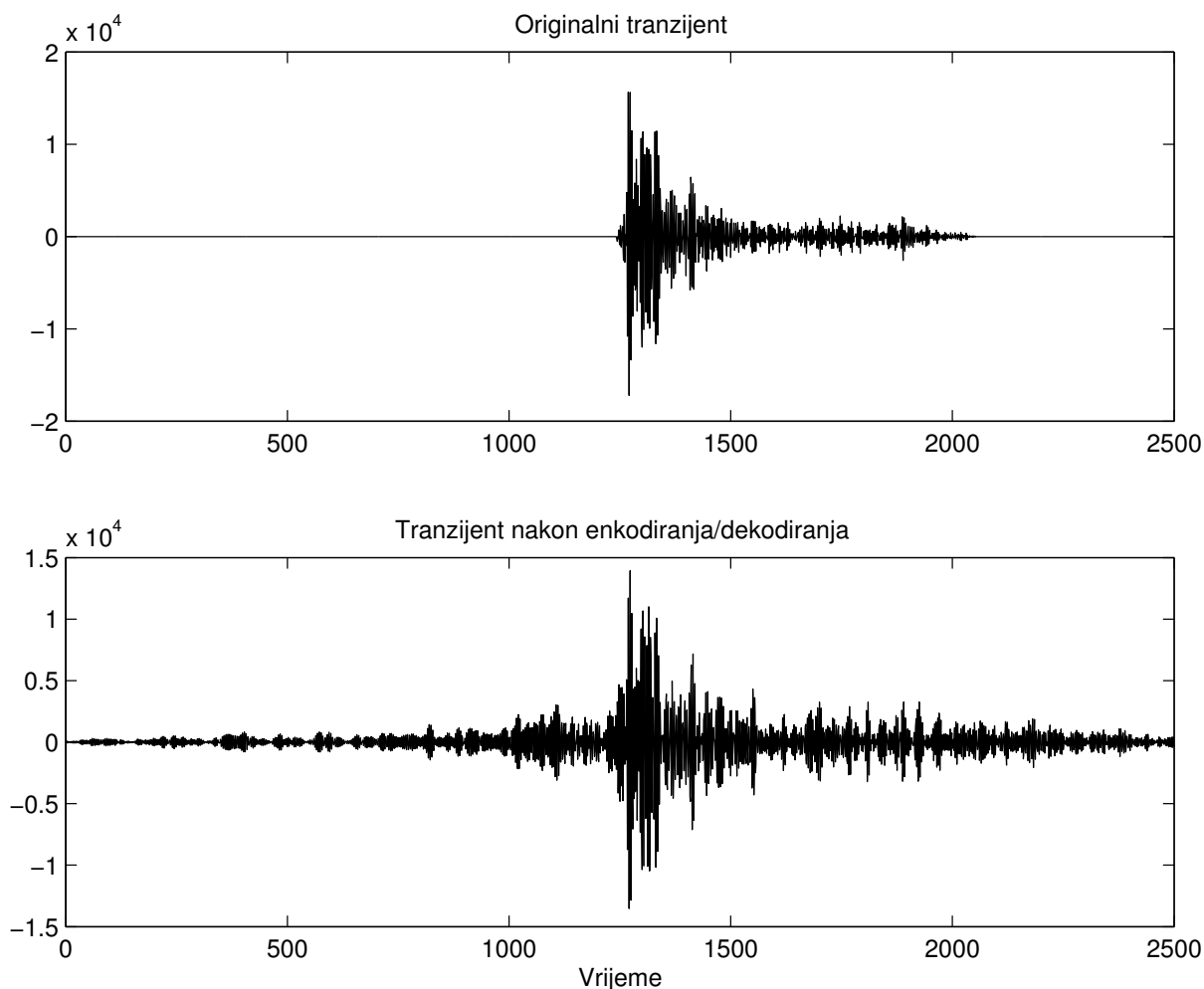
Zbog svih ovih razloga, tranzijentan signal skokovito povećava entropiju pa se to njegovo svojstvo može iskoristiti pri detekciji tranzijenta. Nakon izračuna percepcijske entropije, eksperimentalno se određuje granica kod koje se signal proglašava stacionarnim ili tranzijentnim.

4.2.2.4 Efekt *pred-jeke*

Efekt *pred-jeke* je efekt širenja kvantizacijskom šuma preko cijelog bloka signala u vremenskoj domeni. Zamislimo vremenski signal razložen na frekvencijske komponente. Unošenje šuma u jednu frekvencijsku komponentu odrazit će se na cijeli vremenski blok signala, nakon što se signal vrati iz frekvencijske u vremensku domenu. Kod stacionarnih signala spomenuti *razliveni* šum će biti nečujan jer će biti maskiran signalom. Kod tranzijentnih signala međutim, vremenski blok signala nije ujednačen. Prvi dio bloka je bez signala, dok se u drugom dijelu bloka naglo pojavljuje tranzijent. Šum koji se proširi u

vremenskoj domeni nakon dekodiranja, neće moći biti maskiran signalom, pa će biti jako čujan. Opisana pojava se naziva efekt *pred-jeke* i prikazana je na Slici 4.7.

Slika 4.7: Kodiranje izrazito tranzijentnih signala dovodi do pojave efekta *pred-jeke* odnosno do efekta širenja kvantizacijskog šuma u vremenskoj domeni. U prvoj polovini bloka, ne postoji signal koji bi maskirao šum, pa je nakon dekvantizacije i rekonstrukcije šum jako čujan.



Postoji nekoliko načina eliminacije efekta *pred-jeke*, od kojih su danas najpouzdanije metoda upotrebe kratkih blokova i TNS (engl. *Temporal Noise Shaping – TNS*) metoda. Obje metode će biti prezentirane u idućim poglavljima.

4.2.3 Uobličavanje frekvencijskog spektra

Iako svaki percepcijski audio koder ima ugrađen blok koji se može nazvati blokom za uobličavanje spektra, vrlo je teško definirati njegovu standardnu ili ustaljenu arhitekturu. Ovaj

blok se jako razlikuje od kodera do kodera, a sigurna je samo činjenica da se on uvijek pojavljuje. Zato na ovom mjestu neće biti detaljno opisan, nego će ovaj blok biti predstavljen na način na koji se on pojavljuje u izvedbi općenitog MPEG-4 kodera koja je opisana u idućim poglavljima.

4.2.4 Kvantizacija

Kvantizacija se kod percepcijskih audio kodera vrši uzimajući u obzir rezultate psihoakustičke analize. Komponente se kvantiziraju poštujući zadani omjer *signal/prag maskiranja* za svaku komponentu, tako da svaka komponenta u idealnom slučaju ima svoj faktor kvantizacije. U realnim slučajevima komponente se grupiraju, pa unutar određenih frekvencijskih pojaseva imaju isti faktor kvantizacije. Grupiranje komponenti je na neki način kompromis između sasvim konzervativne uniformne kvantizacije, i kvantizacije koja bi svakoj komponenti signala pridjeljivala poseban kvantizator, što bi pogubno utjecalo na potrošnju bita. Kompromisom se zadržava razuman odnos prema uobličavanju kvantizacijskog šuma obzirom na rezultate psihoakustičke analize sa jedne strane, dok se sa druge strane ipak kodirani tok previše ne opterećuje sa prijenosom kvantizatora.

Postoji nekoliko vrsta kvantizacije koje su korištene ili se još koriste u audio kodiranju, a mogu se podijeliti u nekoliko skupina kao što su skalarna (uniformna, ne-uniformna, diferencijalna, adaptivna), *block companding*, vektorska, parametarska (zamjena šuma, zamjena koeficijenata), i dr.

Neuniformna skalarna kvantizacija

Većina kodera u početku razvoja percepcijskog audio kodiranja je imala izvedenu uniformnu kvantizaciju. Nove spoznaje u psihoakustici dovele su do zaključka da neuniformna kvantizacija puno više odgovara novim principima percepcijskog audio kodiranja. Ona je postala presudna za poboljšanje kvalitete kodiranog audio signala. Tako se danas koriste isključivo koderi sa neuniformnom kvantizacijom. Ova vrsta kvantizacije ima već u sebi ugrađeno određeno uobličavanje šuma tako što se slabije komponente kvantiziraju točnije, dok se jače komponente kvantiziraju manje točno. Iako se obično neuniformna kvantizacija upotrebljava da bi reducirala srednju kvadratnu grešku, u audio kodiranju se neuniformna kvantizacija upotrebljava upravo radi automatskog uobličavanja šuma.

Block Companding kvantizacija

Block companding metoda je uspješno korištena u MPEG-1 standardu, a sastoji se od normaliziranja vrijednosti unutar određenog frekvencijskog pojasa na neku maksimalnu vrijednost. Faktor normalizacije se naziva faktor skaliranja (engl. *scalefactor*) ili kvantizator. Nakon normalizacije vrši se daljnja kvantizacija po pojasevima, uzimajući u obzir broj bita koji je na raspolaganju za svaki pojas.

Vektorska kvantizacija

Vektorska kvantizacija, koja podrazumijeva kvantiziranje grupa frekvencijskih komponenti jednim faktorom kvantizacije, danas se sve više upotrebljava u audio kodiranju. Jedna od novijih metoda je Twin-VQ metoda uključena u MPEG-4 standardu, a koristi se za audio kodiranje na vrlo niskim bitskim protocima.

4.2.5 Kompresija bez gubitaka

Kvantizirane vrijednosti frekvencijskih komponenti audio signala zajedno sa dodatnim informacijama o kvantizatorima, obično se u kodiranom toku zapisuju na dva načina: direktno ili se na njima prethodno vrši dodatno kodiranje – entropijska kompresija bez gubitaka. Većina audio kodača danas upotrebljava neku vrstu dodatne kompresije bez gubitaka.

Poznatiji načini entropijske kompresije bez gubitaka su *run-length* kodiranje, Huffmanovo i aritmetičko kodiranje. Danas se u audio kodiranju najviše koristi Huffmanova metoda, dok se aritmetičko kodiranje, iako je vrlo efikasno, često ne koristi jer je patentno zaštićeno od strane tvrtke *IBM*. MPEG serija kodača ima standardiziran modul koji koristi Huffmanovo kodiranje. Nova verzija MPEG-4 standarda uvodi BSAC metodu koja koristi aritmetičko kodiranje, a upotrebljava se za izvođenje skalabilnosti u malim koracima (Poglavlje 5.3.1.5.3).

4.2.6 Kodirani tok

Tipični kodirani tok koji se kao rezultat audio kodiranja dobije na izlazu audio kodača, sastoji se od niza blokova od kojih se svaki sastoji od sljedećih osnovnih komponenti:

- zaglavlja,
- entropijski kodiranih kvantiziranih frekvencijskih komponenti,
- entropijski kodiranih kvantizatora,
- pomoćnih informacija.

Veličina svakog bloka je određena sa tri komponente: frekvencijom kojom je uzorkovan ulazni signal, ciljnim bitskim protokom i dužinom bloka koji se kodira. Proračunava se prema sljedećoj formuli:

$$\begin{aligned} \text{bita_po_bloku} &= \frac{\text{ciljni_bitski_protok} \times \text{dužina_bloka}}{\text{frekvencija_uzorkovanja}} \\ &= \frac{[\text{bita/sec}] \times [\text{uzoraka/blok}]}{[\text{uzoraka/sec}]} = [\text{bita/blok}] \end{aligned} \quad (4.6)$$

Zadatak koder je da unutar zadanog broja bita *stisne* što više informacija relevantnih za što bolju reprodukciju dekodiranog signala. Ukoliko se neki blok sa zadovoljavajućom kvalitetom kodira sa manje bita, onda se nepotrošeni bitovi pamte i upotrebljavaju u zahtjevnijim blokovima.

Zaglavlje svakog bloka sadrži osnovne informacije o signalu općenito, a i o samom bloku čiji je dio. Neki koderi imaju posebno podijeljeno zaglavlje na fiksni i promjenjivi dio. Fiksni dio je isti unutar svakog bloka, a razlog zbog kojeg se ponavlja je omogućavanje sinkronizacije daljinskog prijenosa kodiranog toka čak i nakon prekida prijenosa. Dekoder mora nakon prekida nastaviti sa dekodiranjem, a fiksni dio zaglavlja mu osigurava sve informacije neophodne za taj nastavak. Promjenjivi dio zaglavlja se odnosi na tekući blok, a u njemu se pretežito nalaze informacije koje se mijenjaju od bloka do bloka, kao što su npr. precizna dužina bloka u bitima, i slično.

4.3 Povijesni razvoj audio kodiranja

Povijesni razvoja percepcijskih audio koder je može se podijeliti na dva zasebna ali ipak vrlo povezana dijela. Prvi dio čine audio koderi koji su razvijeni istraživačkim radom na sveučilištima i u tvrtkama, dok drugi dio čine standardizirani i široko prihvaćeni audio koderi. Očito je da su ove dvije grupe koder je vrlo povezane, dapače, jedni su proistekli iz drugih. Detaljniji pregled povijesti razvoja percepcijskih audio koder je dan je u [81].

4.3.1 Rezultati istraživačkog rada

4.3.1.1 OCF koder

K. Brandenburg sa Fraunhofer instituta razvio je OCF (engl. *Optimum Coding in the Frequency Domain*) koder još 1987. Glavni principi su: DCT transformacija nad 512 uzoraka, neuniformna kvantizacija i entropijsko kodiranje. Ovaj koder je imao prvo poboljšanje uvođenjem detekcije tranzijenata i MDCT transformacije (OCF-2), što je i prva upotreba MDCT u audio kodiranju, te drugo poboljšanje (OCF-3) uvođenjem diferencijalnog kodiranja i poboljšanjem psihoakustičkog modela. OCF je postizao transparentnu kvalitetu signala kod 132 kb/s/ch. OCF je vrlo bitan koder jer su se mnogi audio koderi poslije naslanjali na njegove algoritme, npr. ASPEC, MPEG-1 i dr.

4.3.1.2 PXFM koder

J. Johnson iz AT&T Bell laboratorija razvio je 1989. prvi koder koji je procjenjivao nivo šuma koji bi se mogao tolerirati unutar svakog kritičnog pojasa. Koder nazvan PXFM (engl. *Perceptual Transform Coder*) upotrebljavao je DFT za frekvencijsku analizu (2048 uzoraka). Procjenjivao je šum u svakom pojasu posebno i kvantizirao signale unutar pojaseva prema dobivenom šumu. Koder je uspio postići CD kvalitetu kod 192 kb/s.

4.3.1.3 Brandenburg – Johnson hibridni koder

Primarni cilj koder je bio je povećanje frekvencijske rezolucije i bolja aproksimacija kritičnih pojasa ljudskog slušnog sistema, pa je zato upotrijebljen QMF filtarski slog sa četiri pojasa koja su nakon toga još jednom transformirana sa MDCT transformacijom. Kvantizacija i kodiranje je preuzeto iz prethodna dva koder. Detekcija tonalnih i netonalnih komponenti je ovdje prvi put uvedena na bazi predvidivosti faze i amplitude. Slična struktura je kasnije usvojena u MPEG-1 i MPEG-2 (nivo 3) audio koderima.

4.3.1.4 CNET koder

Razvijen u tvrtki CNET 1989. godine, baziran na DFT transformaciji, ovaj koder je prvi uveo predviđanje unatrag u frekvencijskoj domeni, tako koristeći redundanciju stacionarnih signala za dobivanje manjeg bitskog protoka. Kasnije je ova metoda usvojena u MPEG-2 AAC koderu.

4.3.1.5 ASPEC koder

Ideje i rezultati dobiveni eksploatacijom gore spomenutih pet koder su na neki način sistematizirani i ujedinjeni u prijedlog standardizacije audio koder pod nazivom ASPEC (engl. *Adaptive Spectral Entropy Coding*). Ovaj koder je dodao i modul za Huffmanovo entropijsko kodiranje, a imao je kontrolu efekta pred-jeke dinamičkom promjenom dužine bloka za analizu. Ciljni bitski protoci su bili unutar intervala 64 – 192 kb/s/ch. ASPEC je bio baza za standardizirani MPEG-1 audio koder.

4.3.1.6 DPAC koder

DPAC (engl. *Differential Perceptual Audio Coder*) su razvili 1995. godine Paraskevas i Mourjopoulos sa Patras sveučilišta u Grčkoj, koristeći u koderu dugoročnu (engl. *long-term*) analizu signala i eliminaciju redundancija. Uveli su podjelu blokova na *referentne*, koji su imali značajne razlike u odnosu na svoje prethodnike, i *obične* koji su bili slični svojim prethodnicima. Referentni blokovi su kodirani na standardni način uzimajući u obzir psihoakustičke principe i neuniformnu kvantizaciju, dok su obični blokovi kodirani relativno prema svojim prethodnim referentnim blokovima. Nakon evaluacije ovog koder zaključeno je da je upotrebljiv samo za netransparentno kodiranje na niskim bitskim protocima ispod 90 kb/s/ch.

4.3.1.7 MASCAM koder

MASCAM je nastao kao zajednički projekt više tvrtki na čelu sa IRT, Philips i CNET. Koristio je QMF filtarski slog koji je približno aproksimirao širine kritičnih pojasa ljudskog slušnog sistema. Koder je imao 24 neuniformna pojasa, a svaki pojas je procesiran u blokovima od 2 ms. Svaki blok unutar svakog pojasa je imao svoj kvantizator određen prema osnovnim principima maskiranja u frekvencijskoj domeni. MASCAM je davao dobre rezultate kod signala sa frekvencijskim opsegom od 15 kHz na bitskim protocima od 80 do 100 kb/s/ch.

4.3.1.8 MUSICAM koder

Ovaj koder je zasnovan pretežno na MASCAM koderu i na radu istraživačkog tima tvrtke Philips. Za razliku od MASCAM koder, ovaj koder je imao 32 pojasa iste širine (*višefazni* filtarski slog), žrtvujući tako istovjetnost sa ljudskim kritičnim pojasevima, ali zato znatno umanjujući složenost filtarskog sloga. Poboljšan je psihoakustički model koji je koristio posebnu FFT frekvencijsku analizu veće razlučivosti. Blokovi dužine 8 ms su procesirani i

kvantizirani sa 6-bitnim kvantizatorima. Uzorci iz svakog pojasa su kvantizirani obzirom na omjer *signal/prag maskiranja* dobiven iz psihoakustičkog modela. MASCAM je također bio jedan od koderi koji su činili bazu za MPEG-1 audio koder.

4.3.2 Standardizirani audio koderi

4.3.2.1 ISO MPEG-1

ISO/MPEG-1 (engl. *International Standards Organization/Moving Pictures Expert Group*) standard je usvojen 1992. god. nakon četiri godine intenzivne suradnje, konzultiranja i rada najpriznatijih stručnjaka u području audio kodiranja.

MPEG-1 koder vrši kompresiju 16 bitnog PCM audio signala uzorkovanog sa 32, 44.1 ili 48 kHz. Podržava kodiranje jednog ili dva kanala na sljedeća četiri načina:

- mono,
- dual-mono (stereo kao dva nezavisna mono kanala),
- stereo (dva kanala koja dijele informacije, ali bez *združeni-stereo* funkcije),
- *združeni-stereo* (uzima u obzir korelacije između dva kanala).

Ciljni bitski protok je u području od 32 do 224 kb/s po kanalu, ali on nije standardiziran.

MPEG-1 standard je podijeljen u tri nivoa, svaki različitih ciljnih bitskih protoka i različitih skupova algoritama s kojima se cilj postiže (Tablica 4.1). Tri nivoa se razlikuju i po složenosti – prvi je najjednostavniji, a treći najsloženiji.

Nivo 1

Nivo jedan kodira audio u blokovima od po 384 uzorka, grupirajući po 12 uzoraka u 32 frekvencijska pojasa pomoću pseudo QMF filtarskog sloga. Unutar svakog pojasa pronalazi se najveći uzorak po amplitudi i kvantizira na 6 bita, određuje se broj bita koji su na raspolaganju, zatim se bitovi raspoređuju po pojasevima tako da se minimizira omjer *signal/prag maskiranja*. Na kraju se uzorci linearno kvantiziraju i zajedno sa popratnim informacijama šalju u modul za entropijsko kodiranje.

Nivo 2

Nivo dva kodira audio u blokovima od po 1152 uzorka, grupirajući 3 puta po 12 uzoraka u 32 frekvencijska pojasa. Unutar pojasa svaka grupa od 12 uzoraka ima svoj kvantizator, međutim

kvantizatori susjednih pojaseva se mogu grupirati obzirom na međusobne razlike čime se postiže ušteda u broju kvantizatora. Ostatak algoritma je sličan nivou jedan.

Nivo 3

Nivo tri MPEG-1 audio koda je najsofisticiraniji, a u mnogo čemu se oslanja na iskustva stečena u OCF i MUSICAM koderima. Glavna promjena u poređenju sa nivoima jedan i dva je dodatak MDCT filtarskog sloga koji, za razliku od višefaznog QMF filtarskog sloga, ima svojstvo idealne rekonstrukcije. Dakle, nakon višefaznog filtarskog sloga koji dijeli ulazni signal u 32 pojasa, vrši se daljnja podjela svakog pojasa MDCT transformacijom na 6 ili 18 pod-frekvencija čime se dobiva bolja frekvencijska rezolucija. Transformacija se obavlja na dvije dužine blokova odabrane u skladu sa svojstvima ulaznog signala: ako je ulazni signal tranzijentan upotrebljava se kratki blok od 6 uzoraka, a ukoliko je signal stacionaran, onda se upotrebljava blok od 18 uzoraka. Također, postoji i 50% prekrivanje između susjednih blokova.

Tablica 4.1: Usporedba tri nivoa MPEG-1 audio koda obzirom na optimalni bitski protokol, omjer kompresije i tipičnu primjenu.

	MPEG-1 Nivo 1	MPEG-1 Nivo 2	MPEG-1 Nivo 3
Optimalni bitski protokol po kanalu (kb/s)	192	128	64
Tipični omjer kompresije	1:4	1:6 – 1:8	1:10 – 1:12
Primjena	DCC (engl. <i>Digital Compact Cassette</i>)	Video CD, digitalno radio emitiranje	Lokalno, ADSL tok

4.3.2.2 ISO MPEG-2

MPEG-2 standard se sastoji od dva audio standarda koje se najčešće dijele po načelu kompatibilnosti unazad. Prvi dio pod nazivom MPEG-2 BC (engl. *Backward Compatible*) jeste kompatibilan unatrag sa MPEG-1 standardom, ali proširuje njegove mogućnosti. Tako, MPEG-2 BC dekodirer može dekodirati MPEG-1 kodirani tok, dok obratno nije moguće. Drugi dio pod nazivom MPEG-2 AAC (engl. *Advanced Audio Coding*) je sasvim novi standard koji nije kompatibilan sa MPEG-1 i MPEG-2 BC standardima.

MPEG-2 BC

ISO/MPEG-2 BC standard je proširio mogućnosti MPEG-1 standarda, ali na način da je ostao s njim kompatibilan. MPEG-2 BC je proširio niz frekvencija uzorkovanja koje podržava, dodajući podršku za 16, 22.5 i 24 kHz. MPEG-2 BC je također podržao i tzv. 3/2 format kanala sa lijevim, desnim, centralnim, lijevim *surround* i desnim *surround* kanalom. Uvedena su još neka manje bitna poboljšanja.

MPEG-2 AAC

MPEG-2 AAC koder predstavlja jedan od danas najsavršenijih percepcijskih audio kodera. Nakon uspjeha MPEG-1 i MPEG-2 BC kodera, pojavili su se mnogi zahtjevi koje nadogradnja postojećih MPEG kodera nije mogla ispuniti, pa se pristupilo osmišljavanju sasvim novog standarda koji bi naslijedio dobre osobine svojih prethodnika, ali bi ispoštovao nove zahtjeve i nova saznanja proistekla eksploatacijom MPEG-1 i MPEG-2 kodera. Nakon tri godine rada, 1995. godine objavljen je MPEG-2 AAC standard.

MPEG-2 AAC koder objašnjen je detaljno u sljedećim poglavljima jer je on u potpunosti sadržan unutar MPEG-4 General Audio kodera, čija izvedba je i tema ovog rada.

4.3.2.3 ISO MPEG-4

MPEG-4 je najsvježiji audio standard, objavljen 1998. godine. Može se slobodno reći da je MPEG-4 standard na neki način revolucionaran jer je proširio granice audio kodiranja: sadrži algoritme za kodiranje najrazličitijih tipova audio signala, uključujući govor, složeni audio, sintetički audio, MIDI (engl. *Musical Instruments Digital Interface*), pretvaranje teksta u govor, itd. Pošto je izvedba dijela MPEG-4 standarda tema ovog rada, opisu MPEG-4 standarda je posvećeno posebno poglavlje.

4.3.2.4 PASC koder

Phillips Digital Compact Cassete (DCC) je primjer upotrebe standarda MPEG-1 (doduše u pojednostavljenom obliku) pod nazivom PASC (engl. *Precision Adaptive Subband Coding*). Bitski protok je 384 kb/s. Ovaj koder ne upotrebljava posebnu DFT transformaciju za psihoakustički model, tako značajno dobivajući na brzini i jednostavnosti.

4.3.2.5 ATRAC koder

ATRAC (engl. *Adaptive Transform Acoustic Coding*) je razvijen u tvrtki Sony za potrebe audio kodiranja u Mini Disk formatu. CD kvaliteta je postignuta uz bitski protok od 146 kb/s po kanalu. ATRAC koristi filtarski slog sličan MPEG-1 (nivo 3) hibridnom filtarskom slogu

koji se sastoji od dva stupnja, višefaznog QMF filtra i MDCT transformacije. Također, ATRAC sadrži i blok za detekciju tranzijenata. Poslije MDCT analize, uzorci se grupiraju u 52 neuniformna pojasa (kritični pojasevi), prema kojima se vrši daljnja obrada. Treba napomenuti da se nakon detekcije tonalnih komponenti vrši dodjela bita tako da tonalni pojasevi dobivaju više, a netonalni manje bita i to više nižim frekvencijama, a manje višim, što je prilično konzervativan, ali zato jednostavan princip.

Osim što se koristi za Sony *MiniDisk* tehnologiju, ATRAC se koristi i kod SDSS (engl. *Sony Dynamic Digital Sound*) formata za kino zvuk. SDSS koristi 8 nezavisnih ATRAC kanala za lijevi, lijevi centralni, centralni, desni centralni, desni, *subwoofer*, lijevi *surround* i desni *surround* kanal.

4.3.2.6 PAC koder

AT&T PAC (engl. *Perceptual Audio Coder*) koder je također, kao i MPEG standardi, proizašao direktno iz iskustva stečenih testiranjem PXFM i ASPEC koderima. PAC je višekanalni koder, teoretski podržava do 16 kanala. Filtarski slog koristi isključivo MDCT transformaciju koja se vrši nad 2048 uzoraka ulaznog signala kod stacionarnih, ili nad 256 uzoraka kod tranzijentnih signala. Kvantizacija se vrši nad 49 ili 14 pojaseva (kod signala 44.1 kHz). PAC koder je usavršen u EPAC i MPAC nasljednike, koji su postigli vrlo povoljne omjere bitskog protoka i kvalitete signala, pa su uvršteni u najuže razmatranje pri odabiru raznih standarda (npr. *USA Digital Radio*).

4.3.2.7 AC-2 i AC-3 koderi

Razvijen u tvrtki Dolby 1990. godine, AC-2 koder je jednokanalni koder koji postiže transparentnu kvalitetu zvuka između 128 i 192 kb/s. Postoje četiri varijante AC-2 koderima, međutim sve dijele zajedničku osnovnu arhitekturu: ulazni signal se prebacuje u frekvencijsku domenu upotrebom MDCT transformacije, uveden je novi Kaiser-Bessel prozor koji ima vrlo dobro gušenje u nepropusnom pojasu, frekvencijske komponente su grupirane u pojaseve koji odražavaju kritične pojaseve ljudskog slušnog sistema, vrši se kvantizacija i na kraju pakiranje u kodirani tok. MDCT transformacija se obavlja na dvije dužine blokova od 512 i 128 uzoraka, uzimajući tako u obzir tranzijentne signale. Dva od četiri AC-2 algoritma su dizajnirana za malo kašnjenje i malu složenost, dok druga dva imaju veće kašnjenje i namijenjena su za sisteme sa visokom vjernosti reprodukcije (engl. *high fidelity – hi-fi*).

Nakon početnog uspjeha jednokanalnog AC-2 koderima, razvijen je u mnogočemu poboljšan AC-3 koder, sa ciljem podržavanja višekanalnih sistema. Ovaj koder podržava 5.1 raspored

kanala, pa je ubrzo postao *de facto* standard za kino ozvučenje pod imenom Dolby Digital ili Dolby Surround*. AC-3 koder se također upotrebljava kod DVD-a, satelitskih sistema, kablovske televizije, a danas mnogi *hi-fi* sustavi sadrže ugrađene AC-3 dekodere.

4.4 Zaključak

Ovo poglavlje je ugrubo podijeljeno na dva glavna dijela. U prvom dijelu je opisan opći audio koder. Osnovni cilj opisa općeg kodera je predstavljanje osnovnih principa rada percepcijskih audio kodera danas. Glavni ukratko opisani dijelovi su: filtarski slog, psihoakustički modul, modul za kvantizaciju i kompresiju bez gubitaka, i na kraju kodirani tok. Pojašnjen je rad baš ovih dijelova kodera jer su prisutni u gotovo svim danas poznatim izvedbama audio kodera koji se vežu uz načela psihoakustike.

U drugom dijelu poglavlja je dan kratak povijesni pregled razvoja audio kodiranja zbog lakšeg uvođenja u shvaćanje rada jednog visoko sofisticiranog kodera kao što je MPEG-4 General Audio koder, koji će biti opisan u sljedećim poglavljima. Drugi cilj je bio stjecanje osjećaja o putu koji je znanstvena zajednica prešla do ovog trenutka i gdje bi mogla krenuti u budućnosti, obzirom na impresivan razvoj u zadnjih desetak godina.

U nadolazećim poglavljima slijedi analiza MPEG-4 audio standarda, a nakon toga i predstavljanje konkretne izvedbe jednog njegovog dijela.

* Prvi film ozvučen AC-3 zvukom bio je *Star Trek* 1991. godine.

Poglavlje 5

5 Međunarodni standard *ISO 14496* (MPEG-4)

5.1 Uvod

Standard ISO 14496 (engl. *International Standards Organization*), poznatiji pod nazivom MPEG-4, posljednji je iz serije MPEG standarda (prva verzija je izašla 1998., a druga 2001. godine). MPEG-4 je vrlo opširan multimedijalni standard koji obuhvaća video i audio, i njihovu međusobnu povezanost i sinkronizaciju (Slika 5.1). Mogućnosti kreiranja multimedijalnih audio/video sadržaja koje pruža MPEG-4 gotovo da su nesagledive. MPEG-4 omogućuje kombiniranje teksta, slike, 2D i 3D animacija, sintetičkog zvuka, govora, složenog audio signala, itd. Iako je nastao kao logičan nastavak MPEG-1 i MPEG-2 multimedijalnih standarda, MPEG-4 je ipak na određeni način revolucionaran. Osim tradicionalnog predstavljanja 2D i 3D audio/video sadržaja, on omogućuje i tzv. *miješanje medija*. Sada se grafika i tekst ne moraju kodirati kao pikseli, nego imaju svoje posebno definirane metode, kako za predstavljanje tako i za združivanje sa tradicionalnim audio i video sadržajima, tako kreirajući jedinstven multimedijalni doživljaj. MPEG-4 također omogućava i dosad izostavljanu interaktivnost: kako je jedna multimedijalna scena predstavljena sa nizom posebnih objekata, moguće je interaktivno djelovati na izdvojene objekte unutar scene, scena je postala fleksibilna. Tako je postalo suvišno višestruko predstavljanje istih multimedijalnih sadržaja u različitim formatima, a uštede su ogromne.

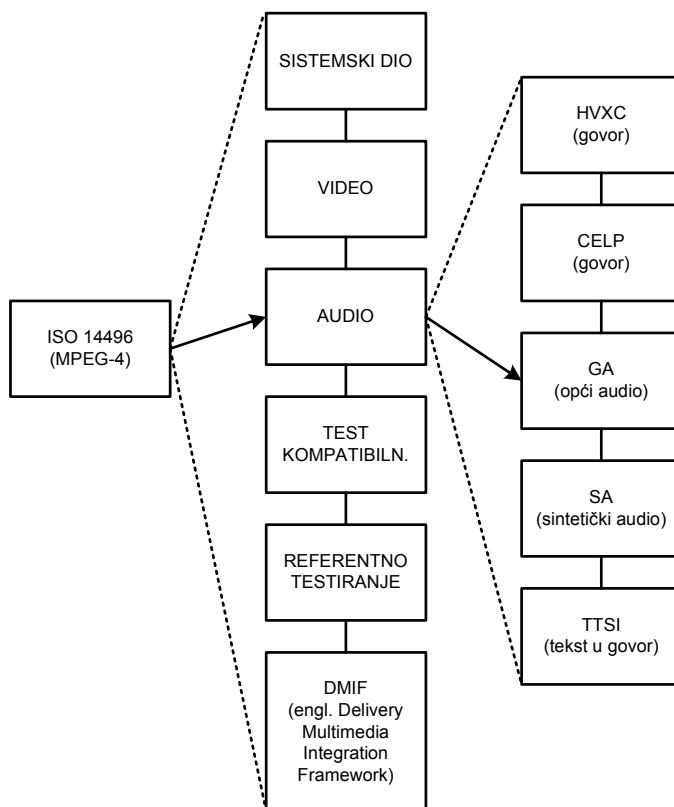
Uspjeh MPEG-4 standarda već je gotovo zagarantiran. *De facto* je usvojen kao standard za multimedijalne sadržaje niske propusnosti, za 3G mobilne uređaje, digitalnu televiziju, itd. Vodeći davatelji audio-vizualnih sadržaja već su usvojili MPEG-4 (tvrtke Apple, Real

Networks). Trenutno najpopularniji DivX video koder je zasnovan na MPEG-4 standardu. Također i proizvođači hardwarea usvajaju MPEG-4: tvrtka Sony nudi svoje najnovije PDA (engl. *Personal Digital Assistant*) uređaje i DVC (engl. *Digital Video Camera*) kamere sa mogućnostima kreiranja MPEG-4 sadržaja, Philips je objavio prvi MPEG-4 audio CD uređaj, *DVD Forum* upravo ispituje MPEG-4 kao format za sljedeću generaciju uređaja za pohranjivanje.

5.2 MPEG-4 Audio

Treći dio MPEG-4 standarda odnosi se na kodiranje audio signala. Unutar ovog dijela definirano je kodiranje širokog spektra audio signala uključujući kodiranje govora, širokopojasnog audio signala, sintetičkog audio signala, pretvaranje teksta u govor, i dr.

Slika 5.1: Organizacija MPEG-4 standarda. *General Audio* dio standarda se odnosi na kodiranje širokopojasnog, složenog audio signala.



U najgrubljoj podjeli MPEG-4 audio možemo podijeliti na kodiranje prirodnog audio signala i sintetičkog audio signala.

5.2.1 Prirodni audio

Prirodni audio se odnosi na govor i širokopojasni složeni audio signal.

Za kodiranje govora upotrebljavaju se dvije vrste algoritama, CELP i HVXC. HVXC se upotrebljava za frekvencije uzorkovanja od 8 kHz, gdje je ciljni bitski protok između 2 i 4 kb/s. CELP se upotrebljava za tzv. širokopojasni govorni signal frekvencije uzorkovanja od 16 kHz, za ciljni bitski protok od 6 do 24 kb/s.

Složeni audio signal se kodira općenitim* (engl. *General Audio – GA*) percepcijskim koderom. MPEG-4 GA obuhvaća kodiranje prirodnog audio signala na bitskim protocima od niskih 6 kb/s do visokih nekoliko stotina kb/s po jednom kanalu audio signala. Za bitske protoke iznad 16 kb/s koristi se AAC kodiranje direktno naslijeđeno iz MPEG-2 AAC dijela, ali sa određenim poboljšanjima. Za niže protoke od 6 kb/s do 16 kb/s predviđen je poseban Twin VQ koder. U novom izdanju standarda [45] uvedena je i BSAC (engl. *Bit Sliced Arithmetic Coding*) kompresija bez gubitaka, koja omogućava fino podešavanje bitskog protoka u koracima od 1 kb/s, uz zadržavanje ostalih dijelova koda istih kao i kod standardnog AAC koda. MPEG-4 GA je u potpunosti kompatibilan sa MPEG-2 NBC AAC koderom, s tim da donosi i određena poboljšanja. Treba napomenuti da se kodirani tok koji je generiran MPEG-2 NBC AAC koderom može dekodirati MPEG-4 GA dekoderom, dok je obrnuto moguće samo u slučaju da MPEG-4 GA koder ne koristi nove funkcije po kojima se razlikuje od MPEG-2 NBC AAC koda.

5.2.2 Sintetički audio

Kod prirodnog audia, već postojeći signal se kodira, prenosi do primatelja pa dekodira. Ovaj tip kodiranja je već vrlo dobro poznat, dok se međutim novi tip audio signala, pod nazivom *sintetički audio*, prvi put pojavljuje unutar MPEG-4 standarda. Sintetički audio signal se *opisuje* (a ne kodira), nakon čega se prenosi do primatelja gdje se *sintetizira* (a ne dekodira).

Jedan od dijelova MPEG-4 standarda koji se odnosi na sintetički audio je sučelje za pretvaranje teksta u govor (engl. *Text To Speech Interface – TTSI*). TTSI omogućava sintetiziranje govora, bez njegovog prethodnog snimanja.

Ovaj dio MPEG-4 audio standarda također donosi standardiziran opisni jezik čijim se tumačenjem sintetizira audio na strani dekoda. Opisani su *instrumenti* čijim tumačenjem se

* Općeniti MPEG-4 audio koder iz naslova ovog rada odnosi se upravo na General Audio dio MPEG-4 standarda.

generira određeni audio signal prema zadanom algoritmu. Niz komandi opisanih SASL (engl. *Structured Audio Score Language*) jezikom poziva, u određenim vremenskim intervalima, instrumente opisane SAOL (engl. *Structured Audio Orchestra Language*) jezikom, tako stvarajući složeni audio signal kreiran od više različitih instrumenata.

Definiran je i način za uključivanje audio uzoraka, koji se upotrebljavaju u sintezi zvuka sa MIDI *Downloaded Sounds Level 2* standardom. Inače, cijeli MPEG-4 strukturirani audio je kompatibilan sa MIDI standardom, pa se MIDI može koristiti kao zamjena ili dodatak SASL jeziku.

5.2.3 Profili, audio objekti i moduli

MPEG-4 audio se sastoji od različitih algoritama kojima se opisuju načini djelovanja na audio signale sa ciljem postizanja određene transformacije tih signala. Ovi algoritmi većinom ne daju korisne rezultate ako su primijenjeni sami, nego moraju djelovati isključivo kao grupa algoritama, pa tek tako kombinirani imaju mogućnost suvisle promjene ciljanog signala. Niz takvih algoritama grupiran u jednu logičku cjelinu koja daje određeni rezultat naziva se *modul*. Svaki od modula vrši na ulaznom signalu unaprijed određenu i očekivanu promjenu.

Međutim, da bi se od *sirovog* ulaznog audio signala dobio standardom definiran izlazni signal – kodirani tok – kodirani audio, potrebno je na signal djelovati sa više modula. Niz kaskadiranih modula koji grupirani daju koristan rezultat na svome izlazu, u MPEG-4 standardu naziva se *audio objekt*.

Kako bi se poboljšala interoperabilnost i razumijevanje među implementatorima pojedinih dijelova standarda, a i zbog činjenice da se u određenoj audio aplikaciji koriste samo određeni audio objekti, MPEG-4 standard je podijeljen na *profile*. Profili nisu ništa drugo do skupovi audio objekata koji su na neki način logički grupirani, uglavnom obzirom na ciljane primjene.

5.2.3.1 Profili

MPEG-4 standard za sada definira osam profila. Četiri profila su definirana u prvoj verziji standarda, dok su u drugoj dodana još četiri. Spomenuti profili su: *Speech Audio* profil, *Synthesis Audio* profil, *Scalable Audio* profil, *Main Audio* profil, *High Quality Audio* profil, *Low Delay Audio* profil, *Natural Audio* profil i *Mobile Audio Internetworking* profil.

Speech Audio profil objedinjuje audio objekte koji se odnose na kodiranje govora (HVXC i CELP). U ovaj profil je uključen i TTSI, audio objekt koji opisuje fonemski opis govora za prijenos na vrlo niskim bitskim protocima od 200 do 1200 bit/s.

Synthesis Audio profil omogućava generiranje zvuka i govora na vrlo niskim bitskim protocima, a uključuje TTSI audio objekt i Main Synthetic audio objekt koji uključuje sve module za generiranje strukturiranog sintetičkog audia.

Scalable Audio profil je nadograđeni *Speech Audio* profil koji osim audio objekata za kodiranje govora, sadrži sve dodatne audio objekte za skalabilno kodiranje i govora (CELP, HVXC, TTSI) i složenog audio signala (Low Complexity, LTP, Twin VQ), i njihovu kombinaciju (AAC Scalable) za prijenos putem Interneta ili digitalnog radija.

Main Audio jedan je od najsveobuhvatnijih profila, a obuhvaća niz audio objekata iz sva tri prethodna profila, uključujući audio objekte za kodiranje govora, složenog audio signala, sintetičkog audia i skalabilno kodiranje.

High Quality Audio profil sadrži samo dio audio objekata koji se odnose na visoko-kvalitetno kodiranje govora (CELP) i složenog audio signala (Low Complexity). Uključuje i skalabilno kodiranje (AAC Scalable).

Low Delay profil se prvenstveno odnosi na kodiranje u realnom vremenu sa malim kašnjenjem, pa sadržava AAC Low Delay audio objekt. Također, u ovaj profil su uključeni i CELP i HVXC audio objekti za kodiranje govora, i TTSI audio objekt za fonemski opis govora.

Natural Audio profil sadrži sve audio objekte koji se odnose na kodiranje prirodnog audio signala, uključujući govor (CELP, HVXC, TTSI) i složeni audio (AAC: Main, Low Complexity, Low Delay).

Mobile Audio Internetworking profil sadrži audio objekte za kvalitetno kodiranje složenog audio signala (Low Complexity), ali i na niskim bitskim protocima (BSAC, Twin VQ) i bez kašnjenja (Low Delay). Namjera je integriranje složenog audio kodiranja u aplikacije koje već koriste kvalitetno kodiranje govora izvedeno na neki drugi način.

Zadnja četiri profila sadrže i proširene verzije svih audio objekata koje sadrže (engl. *Error Resilient – ER*), a proširenje se odnosi na otpornost na greške. ER audio objekti naknadno su doneseni u drugoj verziji MPEG-4 standarda [45].

5.2.3.2 Audio objekti i moduli

MPEG-4 audio objekti se mogu svrstati u tri glavne skupine:

- 1) audio objekti koji definiraju kodiranje govora,
- 2) audio objekti koji definiraju kodiranje složenog, širokopojasnog audio signala, i
- 3) audio objekti koji definiraju kodiranje sintetičkog audia.

5.2.3.2.1 Govor

Za kodiranje govora definirana su tri audio objekta od kojih svaki sadrži i jedan istoimeni modul.

CELP

CELP (engl. *Code Exciter Linear Prediction*) audio objekt sadrži CELP modul unutar kojeg je definirano kodiranje jednog mono govornog kanala uzorkovanog sa 16 kHz ili 8 kHz na bitskim protocima od 4 kb/s do 24 kb/s. Također je definirana skalabilnost bitskog protoka i skalabilnost širine frekvencijskog pojasa.

HVXC

HVXC (engl. *Harmonic Vector Excitation Coding*) audio objekt sadrži HVXC modul unutar kojeg je definirano kodiranje jednog mono govornog kanala uzorkovanog sa 8 kHz na stalnim bitskim protocima od 2 kb/s do 4 kb/s ili na promjenljivom bitskom protoku ispod 2 kb/s. Skalabilnost je također podržana.

TTSI

TTSI (engl. *Text To Speech Interface*) audio objekt sadrži TTSI modul koji definira fonemski opis govora, a ciljani bitski protok je jako nizak, od 200 b/s do 1200 b/s.

5.2.3.2.2 Strukturirani audio

Strukturirani audio se sastoji od četiri audio objekta koji, ovisno o složenosti, sadrže različite kombinacije ukupno tri modula koja definiraju kodiranje strukturiranog audia.

Main Synthetic

Main Synthetic audio objekt sadrži sva tri modula MPEG-4 standarda koja omogućavaju generiranje i manipulaciju strukturiranog audia: SA (engl. *Structured Audio*), SASBF (engl. *Structured Audio Sample Bank Format*) i MIDI. SA modul sadrži opis SAOL jezika koji služi za opis algoritama za generiranje zvuka, i SASL jezika, koji opisuje način na koji se

upotrebljavaju algoritmi za generiranje zvuka definirani SAOL jezikom. SASBF definira načine prijenosa audia definiranog *wavetable* sintezom. MIDI je modul koji se poziva na MIDI standard. MIDI predstavlja alternativu SASL jeziku, ali se oba jezika mogu koristiti paralelno.

Wavetable Synthesis

Wavetable Synthesis audio objekt sadrži dva modula strukturiranog audia: SASBF i MIDI. Predstavlja audio objekt namijenjen stvaranju sintetičkog audia na pojednostavljen način. Zvuk je kreiran jednostavnom manipulacijom audio uzoraka, kao npr. ponavljanjem (engl. *looping*) ili mijenjanjem envelope (engl. *enveloping*).

General MIDI

General MIDI audio objekt sadrži samo MIDI modul, a uključen je zbog lakše upotrebe već postojećih MIDI sadržaja.

Algorithmic Synthesis and Audio FX

Algorithmic Synthesis and Audio FX audio objekt sadrži samo SA modul, dakle odnosi se na SAOL i SASL sintetičke jezike.

Više o kodiranju sintetičkih signala pročitati u [95], [96], [97], [98] i [99].

5.2.3.2.3 Složeni audio

Kodiranje složenog audio signala predstavlja centralni i najsloženiji dio MPEG-4 audio standarda, a naziva se *General Audio*. Kodiranje složenog audia je definirano sa osam glavnih audio objekata koji se sastoje ukupno od preko dvadeset različitih modula. Slika 5.2 prikazuje blok shemu MPEG-4 General Audio koda.

AAC Main

AAC Main audio objekt obuhvaća module s kojima se postiže visoko kvalitetno kodiranje složenog audia na bitskim protocima od 16 kb/s/ch do preko 64 kb/s/ch. Uključuje: MDCT filtarski slog sa promjenom dužine bloka i tipa prozora, sve module za uobličavanje spektra (TNS, PNS, MS, IS), predviđanje u frekvencijskoj domeni (MPEG-2 tip), standardnu AAC kvantizaciju i Huffmanovo kodiranje.

AAC LC

AAC LC (engl. *Low Complexity*) audio objekt se razlikuje od AAC Main objekta samo po činjenici da ne uključuje modul za predviđanje. Kako je modul za predviđanje vrlo računski složen, a donosi dobitak u bitskom protoku od tek 1%, ovdje nije uključen. LC objekt je namijenjen realizacijama koje imaju kritične resurse, pa im izostavljanje modula za predviđanje značajno pojednostavljuje izvedbu, sa vrlo malim utjecajem na konačnu kvalitetu.

AAC LTP

AAC LTP (engl. *Long Term Prediction*) audio objekt se također neznatno razlikuje od AAC Main objekta: umjesto standardnog modula za predviđanje, on sadrži modul za dugoročno predviđanje (engl. *Long Term Prediction – LTP*), koji je mnogo jednostavniji za realizaciju od standardnog, a daje slične rezultate.

AAC SSR

AAC SSR (engl. *Scalable Sampling Rate*) audio objekt ima poseban način suzbijanja efekta pred-jeke sa modificiranim filtarskim slogom. Filtarski slog nije standardnog MDCT tipa nego je hibridne izvedbe (PQMF filtarski slog sa MDCT) koja preko Gain modula omogućava kontrolu efekta pred-jeke.

AAC Scalable

AAC Scalable audio objekt ima definiranu posebnu sintaksu koja podržava skalabilnost bitskog protoka i skalabilnost frekvencijskog raspona. Moguće su razne kombinacije ovog objekta sa Twin VQ i CELP objektima u kreiranju skalabilnih kodiranih tokova. Ovaj objekt jedini sadrži SIAQ (engl. *Scalable Inverse AAC Quantization*) modul, FSS (engl. *Frequency Selective Switch*) modul i UFT (engl. *Upsampling Filter Tool*) modul. Podržani su mono i stereo kanali.

Twin VQ

Twin VQ audio objekt je namijenjen kodiranju složenog audio signala na niskim bitskim protocima, ispod 16 kb/s. Njegova glavna karakteristika je poseban Twin VQ modul za kvantizaciju koji se koristi umjesto tradicionalne AAC kvantizacije i Huffmanovog kodiranja.

BSAC

BSAC (engl. *Bit Sliced Arithmetic Coding*) audio objekt omogućava skalabilnost kodiranog toka u malim koracima od po 1 kb/s po kanalu. Upotrebljava se u kombinaciji sa standardnim

AAC modulima, osim što zamjenjuje Huffmanovo kodiranje definirano u standardnom AAC kvantizacijskom modulu.

AAC LD

AAC LD (engl. *Low Delay*) audio objekt definira kodiranje visoko-kvalitetnog audia, ali sa vrlo malim kašnjenjem koje je prihvatljivo u primjenama u realnom vremenu. Ovaj audio objekt osim standardnih AAC modula definira novu, manju dužinu bloka i novi oblik prozora unutar filtarskog sloga za suzbijanje efekta pred-jeke.

5.3 MPEG-4 General Audio

General Audio (GA) je ključna komponenta MPEG-4 audio standarda. GA pokriva pojas bitskih protoka od 16 kb/s/ch naviše. Unutar ovog raspona dobiva se različita kvaliteta signala počevši od prihvatljive *AM radio* kvalitete, pa sve do transparente *CD audio* kvalitete.

Tablica 5.1: Audio objekti i moduli koji se odnose na kodiranje složenog, širokopojasnog audio signala.

Moduli	Audio Objekti	gain kontrola	prebacivanje dužine blokova	oblik prozora - standardan	oblik prozora – AAC LD	filtarski slog MDCT	filtarski slog hibridni	TNS	LTP	IS	coupling	predviđanje	PNS	MS	SIAQ	FSS	upsampling filtar	kvantizacija i kodiranje AAC	kvantizacija i kodiranje Twin VQ	kvantizacija i kodiranje BSAC	AAC ER alat	ER Payload syntax	EP alat
AAC Main		+	+		+			+		+	+	+	+	+				+					
AAC LC		+	+		+			+		+	+		+	+				+					
AAC SSR	+	+	+				+	+		+	+		+	+				+					
AAC LTP		+	+		+			+	+	+	+		+	+				+					
AAC Scal.		+	+		+			+	+	+			+	+	+	+	+	+					
Twin VQ		+	+		+			+	+					+					+				
ERBS AC		+	+		+			+		+			+	+						+		+	+
ER A AC LD				+	+			+	+	+			+	+				+			+	+	+

MPEG-4 General Audio se odnosi isključivo na visoko-kvalitetno kodiranje složenog, širokopojasnog audio signala, koje je definirano preko 8 audio objekata opisanih u Poglavlju

5.2.3.2.3. U Tablici 5.1 je prikazana povezanost ovih audio objekata sa MPEG-4 audio modulima.

5.3.1 General Audio moduli

MPEG-4 GA module je moguće funkcionalno grupirati na sljedeći način:

- moduli koji definiraju vremensko-frekvencijsku transformaciju,
- moduli za obradu spektra,
- moduli za predviđanje,
- moduli za kvantizaciju,
- moduli koji omogućavaju skalabilnost kodiranog toka i
- moduli za korekciju greške.

Analizirajmo svaki od nabrojanih modula s više detalja.

5.3.1.1 Vremensko-frekvencijska transformacija

Unutar MPEG-4 standarda koriste se tri vrste filtarskih slogova. Većinom se koristi MDCT filtarski slog, dok se samo kod audio objekta AAC SSR koristi hibridni filtarski slog koji se sastoji od PQF i MDCT filtarskih slogova. Kod Low Delay audio objekta MDCT filtarski slog je modificiran obzirom na dužinu bloka i oblik prozora.

5.3.1.1.1 MDCT filtarski slog

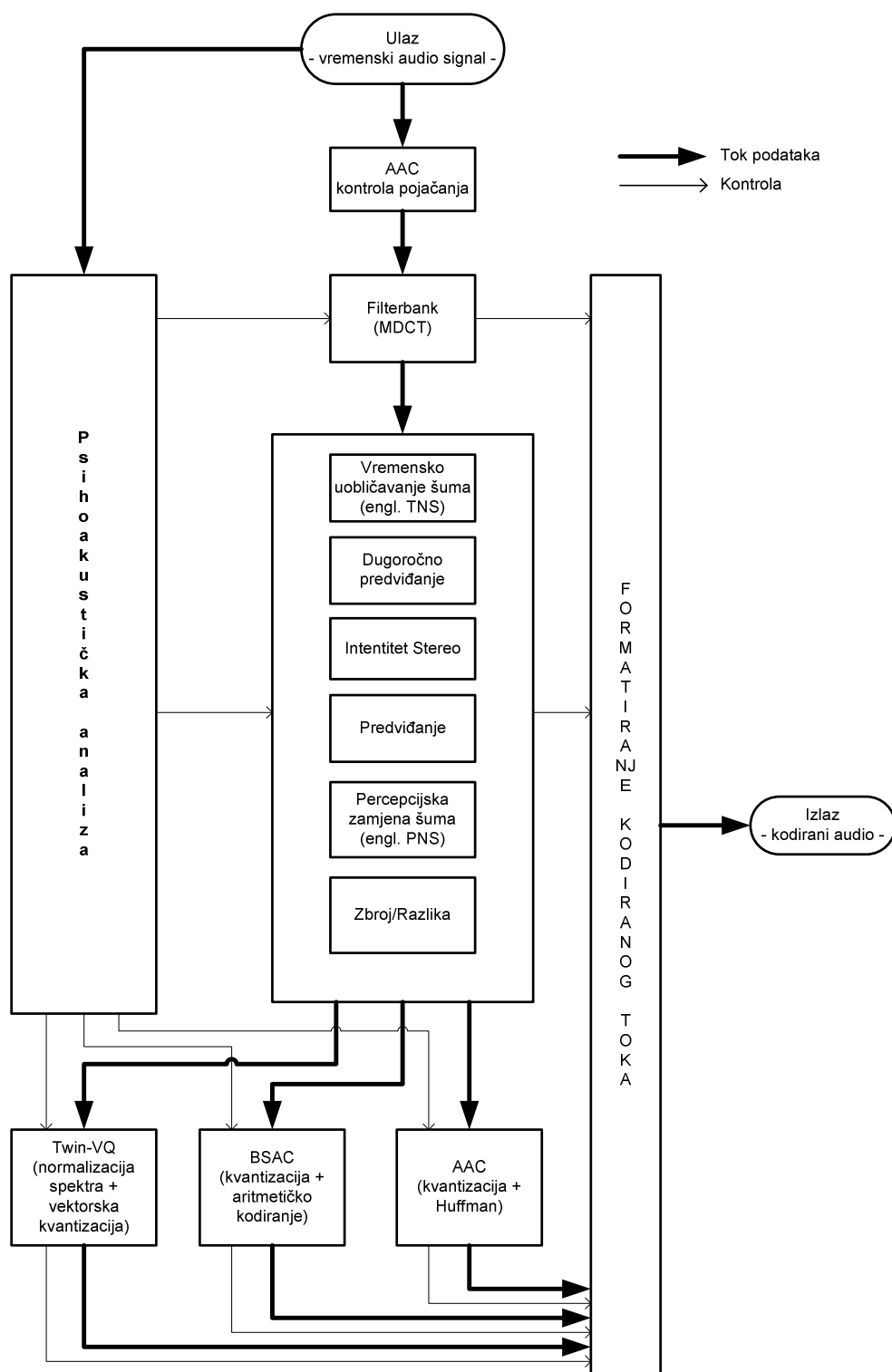
MDCT (modificirana diskretna kosinusna transformacija) filtarski slog [87], [88] danas se najintenzivnije koristi u audio kodiranju. Glavne karakteristike ovog filtarskog sloga su sljedeće:

- idealna rekonstrukcija,
- kritično uzorkovanje,
- eliminacija rubnog efekta,
- brzi algoritmi za proračun.

MDCT filtarski slog je zadan sa impulsnim odzivima moduliranog prototip filtra:

$$h_k(n) = f_k(N-1-n) = w(n) \cos \left[\frac{\pi}{M} \left(k + \frac{1}{2} \right) \left(n + \frac{1}{2} + \frac{M}{2} \right) \right], \quad 0 \leq k \leq (M-1) \quad (5.1)$$

Slika 5.2: Arhitektura MPEG-4 General Audio kodera. Blok dijagram sadrži sve dostupne module definirane u GA dijelu standarda. Tipične konfiguracije kodera u stvarnom svijetu podset su ove konfiguracije, obično usmjerene na specifičnu ciljanu primjenu.



Proračun se vrši kao jedna blok transformacija nad 2048 (standardni blok) ili 256 (kratki blok) uzoraka sljedećom formulom:

$$X(k) = \sum_{n=0}^{2M-1} x(n)h_k(n) \quad (5.2)$$

gdje su:

- $x(n)$ vremenski uzorci,
- $h_k(n)$ prototip FIR filter funkcija (Formula 5.1),
- $w(n)$ vremenski otvor (*prozor* funkcija),
- $X(k)$ dobiveni frekvencijski koeficijenti.

Rezultat transformacije nad $2M$ vremenskih uzoraka je M frekvencijskih komponenti. Da bi bio ispunjen uvjet kritičnog uzorkovanja, sa svakim sljedećim blokom vrši se polovično preklapanje sa prošlim blokom, tako da svaki vremenski uzorak prođe kroz transformaciju dva puta. Na ovaj način se postiže i poništavanje aliasinga. Naime, jedno od glavnih svojstava MDCT transformacije, koja se još naziva i TDAC transformacija (engl. *Time Domain Aliasing Cancellation*) je da, ako se upotrijebi polovično preklapanje susjednih blokova, aliasing uveden rekonstrukcijom jednog bloka, biti će poništen aliasingom uvedenim sljedećim blokom pomaknutim za pola dužine. Na ovaj način, iako blokovi sami po sebi nemaju svojstvo idealne rekonstrukcije, proces preklapanja i zbrajanja u potpunosti poništava aliasing. Također, svakih $2M$ uzoraka dobiva se $2M$ frekvencijskih komponentata, pa je na ovaj način ispunjen i uvjet kritičnog uzorkovanja.

Upotrebom vremenskog otvora i polovičnim preklapanjem izbjegava se još jedna štetna pojava koja se zove rubni efekt - javlja se na rubovima transformacijskih blokova nakon neidealne rekonstrukcije, izazvane kvantizacijom, koja proizvodi diskontinuitete na rubovima koji su u audio kodiranju jako čujni, za razliku od npr. video kodiranja gdje rubni efekt nije problematičan. Vremenski otvor reducira energiju na rubovima blokova pa se ovaj efekt značajno smanjuje.

Inverzna MDCT transformacija se vrši prema formuli:

$$x(n) = \sum_{k=0}^{M-1} \left[X(k)h_k(n) + X_{prev}(k)h_k(n+M) \right] \quad (5.3)$$

iz koje se vidi da je za rekonstrukciju potrebno koristiti i rezultate transformacije prethodnog bloka zbog polovičnog preklapanja.

Da bi filtarski slog ispunio uvjet idealne rekonstrukcije, potrebno je još izabrati funkciju vremenskog prozora $w(n)$ sa sljedećim karakteristikama:

$$w(n) = w(2M - 1 - n) \quad (5.4)$$

i

$$w^2(n) + w^2(n+M) = konst. \quad (5.5)$$

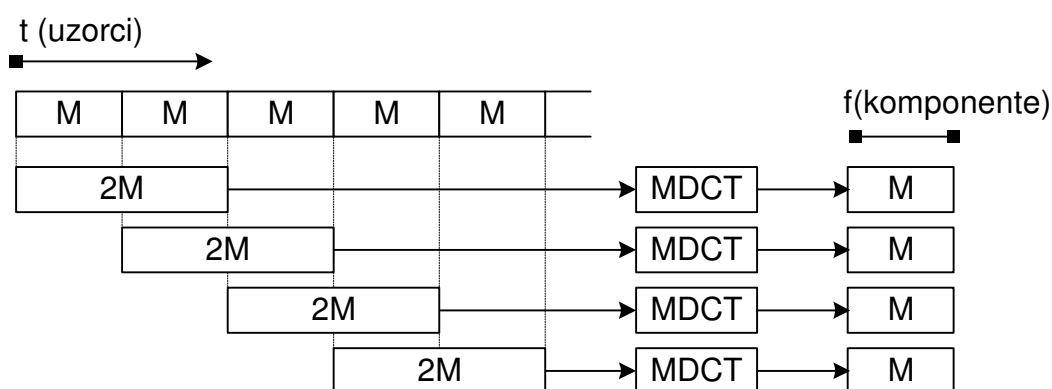
Unutar MPEG-4 standarda zadane su dvije $w(n)$ funkcije: Sinusna i KDB (engl. *Kaiser Bessel Derived*) funkcija.

Uz ispunjene uvjete definirane Formulama 5.4 i 5.5, MDCT filtarski slog ispunjava najvažniji uvjet idealne rekonstrukcije, poništava se aliasing, a smanjuje se i rubni efekt. Sve ove osobine, uz postojanje brzih algoritama za proračun baziranih na FFT transformaciji, čine ovaj filtarski slog vrlo pogodnim za upotrebu u audio kodiranju.

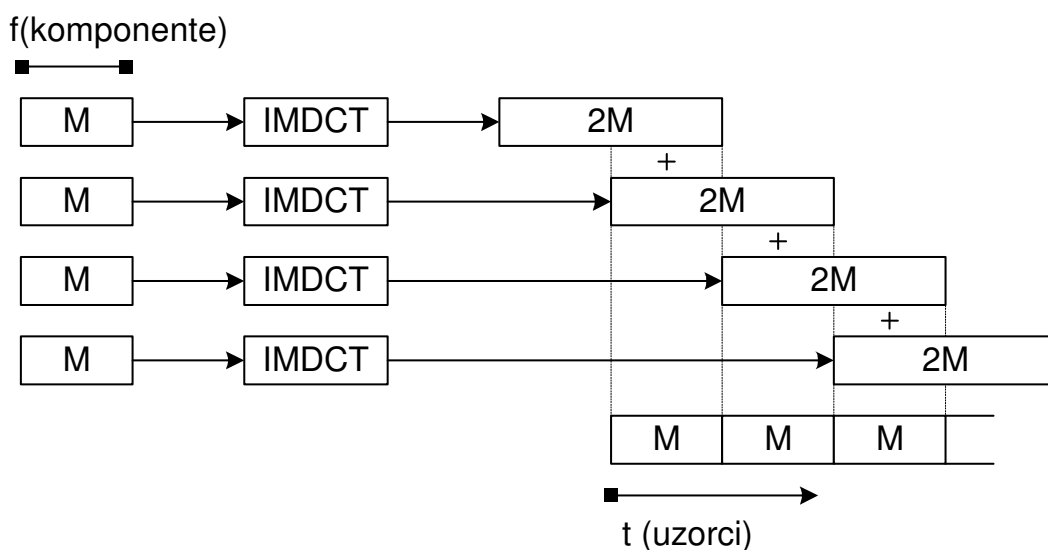
Dvije dužine blokova

MDCT transformacija se u MPEG-4 standardu vrši na jednoj od dvije vrste blokova: standardnom koji ima 2048 uzoraka, i kratkom koji ima 256 uzoraka. Razlog je izvedba opcije izmjene dužine blokova u koderu, čiji cilj je smanjenje efekta pred-jeke. Smanjenjem vremenske rezolucije filtra, efekt pred-jeke se ograničava na manji vremenski interval.

Slika 5.3: a) MDCT transformacija je kritično uzorkovana – za svakih M novih ulaznih uzoraka ona proizvede M novih izlaznih frekvencijskih komponenti. Kritično uzorkovanje se postiže polovičnim prekrivanjem: iako se transformacija vrši nad dvostrukom dužinom niza, uvijek je prva polovica uzeta iz prethodnog bloka. Kod MPEG-4 General Audio filterbanka definirane su dvije dužine blokova: $M=2048$ (standardni blok) i $M=256$ (kratki blok). **b)** Inverzna MDCT transformacija je također kritično uzorkovana – za svakih M novih ulaznih frekvencija ona proizvede M izlaznih vremenskih uzoraka. Iako IMDCT proizvede dvostruko više izlaznih uzoraka nego ulaznih, konačan vremenski signal se dobije tek zbrajanjem lijeve polovice izlaznog bloka sa desnom polovicom iz prošlog bloka.



a)



b)

Kao što je objašnjeno u Poglavlju 4.2.2.4, efekt pred-jeke je pojava širenje kvantizacijskog šuma unesenog u frekvencijskoj domeni kroz cijeli vremenski interval na kojem se vrši frekvencijska analiza. Ovaj problem nije od velikog značaja kod stacionarnih signala,

međutim dolazi do jakog izražaja kod tranzijentnih signala. Kvantizacijski šum će se ravnomjerno raspodijeliti kroz cijeli vremenski interval, ali će na mjestima gdje je signal slabog intenziteta biti više izražen jer neće biti maskiran rekonstruiranim signalom.

Zbog pojave efekta pred-jeke, MPEG-4 koder ima definiranu mogućnost promjene dužine blokova nad kojima se vrši analiza. Kada modul za detekciju tranzijenta signalizira pojavu tranzijenta, prelazi se sa bloka dužine 2048 na blok dužine 256 uzoraka. Nakon što završi tranzijentna pojava analiza se nastavlja na blokovima standardne dužine od 2048 uzoraka. Kratki blokovi se mogu upotrebljavati samo u grupama od 8, da bi se održala ista dužina bloka za analizu od 2048 uzoraka (8×256).

Treba napomenuti da osim standardnog i kratkog bloka postoje i prijelazni blokovi, *dugi-na-kratki* i *kratki-na-dugi*, koji osiguravaju idealnu rekonstrukciju signala koja ne bi bila ostvariva kada bi nakon standardnog bloka odmah bio primijenjen kratki blok. Prijelazni blokovi nisu poželjni jer imaju vrlo slaba svojstva vremenske i frekvencijske lokalizacije, pa treba izbjegavati česta prebacivanja između standardnih i kratkih blokova.

Treba napomenuti da kod MPEG-4 GA kodera postoji posebna dužina bloka određena za upotrebu samo kada se AAC upotrebljava u kombinaciji sa CELP govornim koderom. Kako CELP vrši kodiranje blokova signala trajanja 10 ms, dužina bloka AAC kodera je skraćena sa 1024 na 960 uzoraka, da bi se dužine blokova vremenski izjednačile.

Dva tipa funkcije vremenskog prozora

MPEG-4 standard definira dvije funkcije vremenskog prozora koje su na raspolaganju u vremensko-frekvencijskom modulu. Prva funkcija je poznata *sinus* funkcija, a druga je *Kaiser-Bessel Derived* funkcija. Obje funkcije zadovoljavaju uvjete idealne rekonstrukcije i linearne faze, ali imaju drugačije osobine spektralne analize. Glavni cilj funkcija je da svaka na svoj način optimiraju frekvencijsku selektivnost filtarskog sloga.

Sinusni prozor je pogodniji kad se zahtjeva uža selektivnost unutar propusnog pojasa kao npr. kod signala koji imaju gust spektar, gdje su jake spektralne komponente manje razmaknute. KBD funkcija nauštrb selektivnosti u propusnom pojasu, donosi bolje prigušenje u nepropusnom pojasu. To znači da će za čisti ton koji se pojavi na sredini MDCT bloka, KBD funkcija grupirati više energije u jedan koeficijent nego sinus funkcija. Ova pojava je korisna za signale kod kojih se tonovi nalaze na određenoj većoj frekvencijskoj udaljenosti.

5.3.1.1.2 Filtarski slog sa malim kašnjenjem

Kodiranje sa malim kašnjenjem (engl. *Low Delay – LD*) uvedeno je u MPEG-4 standard zbog potrebe za malim kašnjenjem u sustavima realnog vremena poput mobilnih komunikacija, telefonije ili radio emitiranja. Postojeći MPEG-4 audio ima kašnjenje od više stotina milisekundi, što je u spomenutim sustavima neupotrebljivo.

LD koder je vrlo sličan standardnom General Audio koderu, ali ima nekoliko značajnih izmjena, od kojih su najznačajnije vezane uz filtarski slog.

Na kašnjenje u kodiranju najviše utječu:

- dužina bloka koji se kodira,
- filtarski slog,
- prebacivanje između standardnih i kratkih blokova (detekcija tranzijenta zahtijeva *pogled unaprijed*),
- *bit-rezervoar*.

Da bi se postiglo smanjenje kašnjenja (na ne više od 10 do 20 ms), na osnovu General Audio koder konstruiran je LD koder [4] sa sljedećim osobinama:

- dužina bloka je smanjena sa 1024 na 512 uzoraka,
- izbačeno je mijenjanje dužine blokova, a za eliminiranje efekta pred-jeke uveden je novi oblik prozora,
- *bit-rezervoar* se ne upotrebljava ili je njegova upotreba ograničena.

Glavna promjena se odnosi na dužinu bloka koju obrađuje filtarski slog. Ona je sada upola manja i iznosi 512 vremenskih uzoraka. Također je promijenjen i oblik prozora, pa se u LD koderu koristi *sinus* prozor, ali je umjesto KBD prozora uveden tzv. prozor sa niskim prekrivanjem (engl. *low-overlap*). Ovaj prozor ima maleno prekrivanje sa susjednim prozorima, pa na taj način suzbija efekt pred-jeke. Za netranzijentne signale upotrebljava se *sinus* prozor.

5.3.1.1.3 Hibridni filtarski slog

Samo kod audio objekta AAC SSR, umjesto standardnog koristi se hibridni filtarski slog koji se sastoji od PQF (engl. *Polyphase Quadrature Filter*) filtra i MDCT filtarskog sloga. Razlog je eliminacija efekta pred-jeke koja se ostvaruje na sljedeći način: signal se najprije dijeli PQF

filtrom u 4 pojasa jednake širine, izlaz iz svakog od 4 bloka se pojačava ili prigušuje sa Gain blokom, nakon čega se vrši MDCT transformacija nad četvrtinom standardne dužine.

PQF filter je definiran sljedećom formulom:

$$h_i(n) = \frac{1}{4} \cos\left(\frac{(2i+1) \times (2n+5)}{16}\right) Q(n), \quad 0 \leq n \leq 95, 0 \leq i \leq 3 \quad (5.6)$$

gdje je

$$Q(n) = Q(95 - n), \quad 48 \leq n \leq 95 \quad (5.7)$$

5.3.1.2 Obrada spektra

MPEG-4 GA moduli za obradu spektra (*alati*) će biti detaljno objašnjeni u Poglavlju 8.

5.3.1.3 Predviđanje

MPEG-2 predviđanje u frekvencijskoj domeni

Predviđanje u frekvencijskoj domeni doprinosi eliminaciji ponavljanja izraženo stacionarnih signala. Podržano je samo u Main audio objektu, a naslijeđeno je iz MPEG-2 standarda. Predviđanje unutar kratkih blokova nije podržano. Predviđanje se realizira preko latične strukture drugog stupnja sa prilagodbom unatrag (engl. *backward adaptive*), a računa se posebno za svaku frekvencijsku komponentu. Prebacivanje između upotrebe predviđenih i originalnih vrijednosti vrši se na bazi pojaseva kvantizatora i to ovisno o postignutom dobitku u bitskom protoku.

Algoritam je dosta kompliciran i osjetljiv na zaokruživanja, pa je teško upotrebljiv u izvedbama u aritmetici nepomičnog zarez. Također, upotreba predviđanja čini kodirani tok više osjetljivim na pogreške u prijenosu.

Dugoročno predviđanje

Kao alternativa MPEG-2 predviđanju koje je vrlo složeno, u MPEG-4 standardu je uveden novi algoritam za predviđanje nazvan *dugoročno predviđanje* (engl. *Long Term Prediction – LTP*) predviđanje. LTP uklanja ponavljanje signala unutar susjednih blokova. Posebno je efikasan za signale sa jasno izraženim tonalnim komponentama. Složenost izvedbe LTP algoritma je upola manja od složenosti izvedbe MPEG-2 predviđanja. Također je i manje

osjetljiv na greške zaokruživanja i greške u prijenosu jer je LTP predviđanje realizirano kao predviđanje sa prilagodbom unaprijed (engl. *forward adaptive*).

5.3.1.4 Kvantizacije

Kvantizacija spektralnih vrijednosti je glavni izvor smanjenja bitskog protoka u audio kodiranju. Kvantizacijom se, na osnovu informacija iz psihoakustičkog modula, dodjeljuju bitovi svakom kvantizacijskom pojasu. Ukoliko je pojas percepcijski značajan, on se kodira sa većom kvalitetom, pa mu je potrebno više bita. Pojasevi koji nisu toliko relevantni *jače* se kvantiziraju.

Ključni je tip kvantizacije koji se upotrebljava. MPEG-4 GA koder, osim standardnog AAC kvantizacijsko-kodirajućeg modula, ima standardiziran i TwinVQ kvantizacijsko-kodirajući modul i BSAC kvantizacijsko-kodirajući modul. TwinVQ i BSAC moduli predstavljaju alternativu AAC kvantizacijskom modulu.

5.3.1.4.1 AAC kvantizacija

Funkcija kvantizacije koja se upotrebljava unutar MPEG-4 GA je slična onoj u MPEG-1 i MPEG-2 standardima. Predstavljena je sa nelinearnom funkcijom x karakteristike. Glavna prednost ovakve funkcije je automatsko uobličavanje šuma: veći signali su *automatski* jače kvantizirani nego slabiji. Veličina kvantizatora se može podešavati u koracima od 1.5 dB.

Funkcija kvantizacije definirana je formulom:

$$f_{kv} = \text{int} \left[\left(f \times 2^{-\frac{1}{4}(\text{kvantizator}-100)} \right)^{\frac{3}{4}} + 0,4054 \right] \quad (5.8)$$

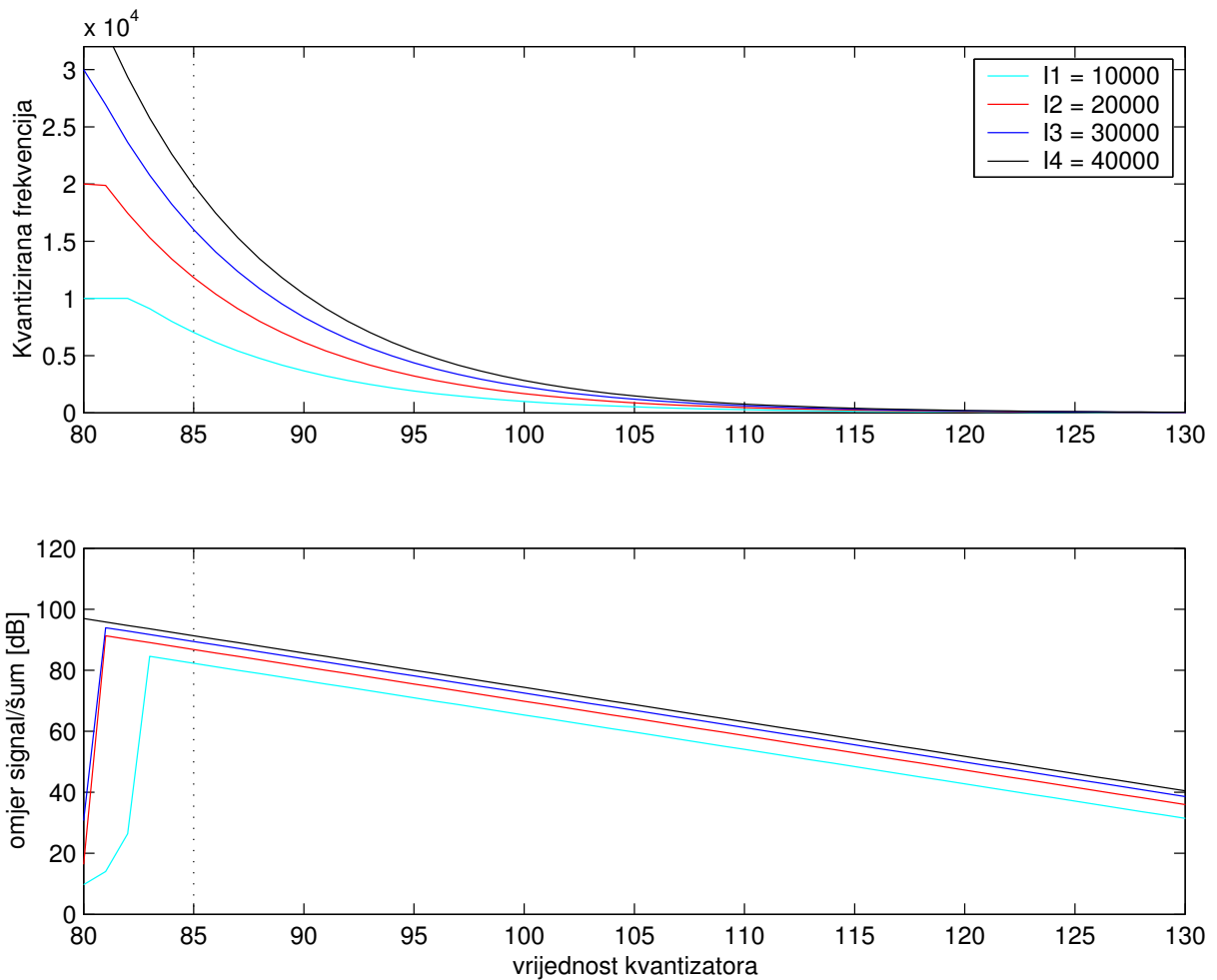
gdje je *int* funkcija *najbliži cijeli broj*. Dekvantizacija se vrši inverznom formulom:

$$f_{dekv} = f_{kv}^{\frac{4}{3}} \times 2^{\frac{1}{4}(\text{kvantizator}-100)} \quad (5.9)$$

Greška kvantizacije je razlika:

$$G = f - f_{dekv} \quad (5.10)$$

Slika 5.4: Neuniformna kvantizacija prema Formuli 5.8. Vrijednost kvantizatora ograničena je standardom na interval $[0,256]$. Na slici je prikazan omjer *signal/šum* za četiri različita intenziteta iste frekvencijske komponente. Uočiti nelinearnu promjenu omjera *signal/šum* sa porastom intenziteta frekvencije: kod kvantizatora 85 i intenziteta 40000, omjer *signal/šum* je 92 dB. Kod istog kvantizatora i 4 puta manjeg intenziteta od 10000, omjer *signal/šum* je 82 dB. Dakle, omjer nije 4 puta manji.



Odnos energija originalnog signala i uvedenog šuma nakon kvantizacije proračunava se na sljedeći način:

$$\frac{Signal}{Šum} = 10 \times \log \left(\frac{f^2}{(f - f_{dekv})^2} \right) = 20 \times \log \left(\frac{f}{(f - f_{dekv})} \right) \quad [dB] \quad (5.11)$$

Ovakvim načinom kvantizacije veće vrijednosti se kvantiziraju s manjom točnošću, dok se manje vrijednosti kvantiziraju sa većom točnošću pa se tako automatski na određeni način vrši uobličavanje šuma: jače komponente imaju jači šum, ali ga one i jače maskiraju.

Pošto su iznosi kvantizatora neophodni za dekodiranje, i oni se prenose u kodiranom toku. Kako promjene susjednih kvantizatora nisu velike, kvantizatori se prvo diferencijalno kodiraju.

Da bi se nakon kvantizacije dodatno smanjila redundancija kodiranih frekvencijskih komponenti i kvantizatora, upotrebljava se dodatna kompresija – Huffmanova entropijska kompresija. Na osnovu maksimalne vrijednosti izdvojene između kodiranih komponenti određenog bloka podataka, selektira se jedna od jedanaest standardom definiranih Huffmanovih kodnih stranica. Kodne stranice su definirane sa tri faktora: maksimalnom vrijednošću koju mogu kodirati, brojem komponenti koje se združeno kodiraju (dvije ili četiri) i da li podržavaju vrijednosti sa predznakom ili se predznak zapisuje u kodirani tok posebno. Jedna od kodnih stranica je nulta. Ona signalizira dekoderu da su sve frekvencijske komponente unutar pripadajućeg pojasa nula. Također i svi kvantizatori su nula pa se na ovaj način postiže dodatna ušteda. Poseban slučaj je i kodna stranica 11 kojom se kodiraju vrijednosti veće od 16.

Brojevi selektiranih tablica se također prenose u kodiranom toku pa se tako donekle povećavaju prenesene sporedne informacije. Da bi se ove sporedne informacije smanjile na minimum, moguće je dodatno grupirati kvantizirane pojaseve sa istim Huffmanovim tablicama, odnosno tablice odabrati na način da se minimiziraju popratne informacije.

Više o Huffmanovom kodiranu u Poglavlju 6.4.9.

5.3.1.4.2 Twin VQ kvantizacija

Twin VQ je nova metoda kvantizacije uvedena u MPEG-4 audio standard sa ciljem poboljšanje kodiranja složenog audio signala na niskim bitskim protocima od 6 do 16 kb/s gdje daje bolje rezultate od standardne AAC kvantizacije sa Huffmanovim kodiranjem. Tradicionalna metoda kodiranja kvantizatora i spektra zamjenjuje se vektorskom kvantizacijom primijenjenom na normalizirani spektar. Ostatak koda je nepromijenjen.

TwinVQ kvantizacija se odvija u dva koraka: prvo se spektralne komponente normaliziraju prema ciljanom rasponu, a zatim se kvantiziraju težinskom vektorskom kvantizacijom. Normalizacija se provodi preko LPC predviđanja, izdvajanja periodičkih komponenti,

spektralnog predviđanja preko Bark pojaseva i predviđanja snage. Parametri koji su povezani sa dobivenim normaliziranim spektrom prenose se u kodiranom toku kao sporedne informacije. Nakon normalizacije, spektralne komponente se preraspodjeljuju i dijele u pod-vektore. Preko svakog pod-vektora kontrolira se šum u pojedinim pojasevima pa se tako poštuju principi percepcijskog audio kodiranja.

5.3.1.4.3 BSAC kvantizacija

BSAC kvantizacija koristi se aritmetičkim kodiranjem da bi se kodirani tok načinio skalabilnim. Zbog ovog razloga, druga verzija standarda donosi BSAC arhitekturu koja omogućuje skalabilnost u koracima od po samo 1 kb/s po kanalu. Upotrebljava se jedan osnovni novo, i više dodatnih nivoa koji doprinose sa 1 kb/s po kanalu. Ovisno o mogućnostima dekodera, dekodira se što više dodatnih nivoa.

5.3.1.5 Skalabilnost

Skalabilnost omogućava prijemniku da izabere iznos bitskog protoka kojim će primati kodirani signal. Na ovaj način jedan pošiljalatelj može slati kodirani audio prema više prijemnika, a da svaki prijemnik, ovisno o svojim mogućnostima, bira bitski protok kojim će primati kodirani audio. Dakle, jedan zajednički kodirani tok se može dekodirati na više načina, umjesto da se za svaki bitski protok signal ponovo kodira i šalje. Skalabilan kodirani tok ima strukturu složenu u više nivoa. Jedan nivo je obično osnovni, a nakon njega se nalaze dodatni nivoi za pojačavanje koje dekoderi mogu i ne moraju uzimati u obzir kod dekodiranja. Naravno, što više nivoa se dekodira, to kvalitetniji će dekodirani audio biti.

Postoje različite rezolucije nivoa koji poboljšavaju kvalitetu audio signala. Dodatni nivoi mogu povećati bitski protok za samo 1 kb/s ili za nekoliko desetaka kb/s, pa se prema ovome skalabilnost dijeli na skalabilnost u velikim koracima (engl. *big step*) i skalabilnost u malim koracima (engl. *small step*).

AAC skalabilnost u velikim koracima

Iako dopušta maksimalno sedam nivoa poboljšanja unutar AAC Scalable audio objekta, skalabilan AAC kodirani tok se obično sastoji od osnovnog nivoa od 24 kb/s, sa mogućnošću podizanja kvalitete sa dva nivoa od po 16 kb/s. Na ovaj način su moguće sljedeće kombinacije po izboru dekodera: 24 kb/s mono, 40 kb/s stereo i 56 kb/s stereo. Zbog dodatnih informacija koje svaki dodatni nivo zahtijeva, skalabilnost sa većom rezolucijom nije isplativa.

AAC skalabilnost možemo ilustrirati na sljedećem primjeru: kombiniramo li CELP koder za govor kao osnovni nivo i GA koder za složeni audio signal kao nivo za poboljšanje, dobiti ćemo bolji koder nego ako bismo upotrijebili samo GA koder i za govor i za složeni audio jer bi GA koder lošije kodirao govor nego CELP.

Skalabilnost uključuje, osim skalabilnosti bitskog protoka, i skalabilnost frekvencijskog raspona. Prva skalabilnost direktno smanjuje izobličenja signala, dok druga povećanjem frekvencijskog raspona na više frekvencije upotpunjuje signal tako poboljšavajući njegovu ukupnu kvalitetu.

MPEG-4 prirodni audio omogućava različite kombinacije osnovnog i nivoa za poboljšavanje. Osim dostupnih kombinacija govornih koda CELP i HVXC, unutar General Audio dijela standarda moguće su sljedeće kombinacije:

- razne kombinacije isključivo AAC nivoa,
- uskopojasni CELP govorni koder kao osnovni nivo, AAC kao poboljšanje,
- Twin VQ kao osnovni nivo, AAC kao poboljšanje,
- AAC kao osnovni, BSAC kao poboljšanje.

Različite kombinacije AAC koda su preporučljive za bitski protok iznad 20 kb/s. Kod npr. digitalnog radio emitiranja gdje je potrebna dobra kvaliteta govora na niskom bitskom protoku, preporučuje se CELP kao osnovni nivo, a AAC kao nivo za poboljšanje. U slučaju da složeni audio signal treba kodirati prihvatljivom kvalitetom na vrlo niskom bitskom protoku od oko 16 kb/s kao kod npr. Internet *streaminga*, preporučuje se Twin VQ kao osnovni koder i AAC kao dodatni.

Frequency Selective Switch

Frequency Selective Switch (FSS) modul pomaže kod skalabilnog kodiranja dodatnog nivoa na osnovu osnovnog nivoa, jer u mnogim slučajevima razlika između izlaznog signala iz osnovnog nivoa i originalnog signala kao ulaz u dodatni nivo nije dovoljna. FSS vrši odabir ulaza u dodatni nivo između razlike ova dva signala i originalnog signala.

Upsampling Filter Tool

UFT (engl. *Upsampling Filter Tool*) se upotrebljava za prilagođenje frekvencija uzorkovanja koda osnovnog nivoa i koda dodatnog nivoa jer su one u mnogim slučajevima različite.

UFT upotrebljava MDCT filtarski slog. Prvo se uzimaju uzorci vremenskog signala kodera osnovnog nivoa, ubacuje se određeni broj nula među ove uzorke pa se takav signal šalje na ulaz MDCT. Na ovaj način se dobiva signal veće frekvencije uzorkovanja.

Skalabilnost u malim koracima – BSAC

Skalabilnost u malim koracima dopušta kontrolu bitskog protoka na strani dekodera u koracima od 1 kb/s po kanalu. Prva verzija MPEG-4 standarda je imala definiranu grubu skalabilnost u koracima od tipično 16 kb/s. Skalabilnost u malim koracima je izvedena preko BSAC metode u drugoj verziji MPEG-4 standarda. BSAC se upotrebljava kao zamjena klasičnom dosadašnjem Huffmanovom kodiranju spektra i kvantizatora.

BSAC kodiranje se vrši na način da se kvantizirane spektralne komponente prvo grupiraju u posebne frekvencijske pojaseve. Bitovi svake grupe se kodiraju prema njihovom značaju. Tako se prvo kodiraju *značajniji* (msb) bitovi, i tako redom. Ovako raskomadani dijelovi se zatim kodiraju *aritmetičkim* koderom. Na ovaj način se sa svakim sljedećim nivoom povećava preciznost kvantiziranih vrijednosti, a također se povećava i frekvencijski raspon signala sa dekodiranjem spektralnih grupa koje se odnose na više frekvencije.

5.3.1.6 Korekcije i prevencije grešaka

MPEG-4 verzija 2 donosi značajna poboljšanja glede otpornosti kodiranog toka na greške u kodiranju i prijenosu kodiranog audio signala. Alati za kontrolu grešaka se mogu ugrubo podijeliti u dvije grupe. Prva grupa sadrži algoritme za povećanje otpornosti na greške, dok druga grupa sadrži općenite alate za zaštitu od grešaka.

Prva grupa koja sadrži alate koji smanjuju eskalaciju i propagaciju greške u kodiranom toku, sastoji se od sljedeća tri alata:

- virtualne Huffmanove kodne stranice koje omogućavaju lakše lociranje grešaka,
- reverzibilno *variable-length* kodiranje zamjenjuje diferencijalno kodiranje kvantizatora, tako povećavajući mogućnost lociranja greške,
- preraspodjela Huffmanovih kodnih riječi omogućava stavljanje određenih Huffmanovih riječi na poznata, obilježena mjesta, tako onemogućavajući propagaciju greške preko ovih riječi.

Druga grupa alata rangira određene dijelove kodiranog toka po njihovoj osjetljivosti na greške, a zatim najosjetljivijim dijelovima pridjeljuje značajniju zaštitu nego drugima. Za

zaštitu se upotrebljavaju FEC (engl. *Forward Error Correction*) ili CRC (engl. *Cyclic Redundancy Check*).

5.4 Zaključak

U ovom poglavlju je dan pregled MPEG-4 audio standarda. Težište je stavljeno na dio koji se odnosi na kodiranje složenog, širokopojasnog, prirodnog audio signala, a naziva se MPEG-4 General Audio.

Najprije je dan pregled organizacije MPEG-4 standarda obzirom na podjelu prema vrstama audio signala na koje se standard odnosi, a to su, u najgrubljoj podjeli, prirodni i sintetički audio. Dalje je predstavljena podjela prirodnog audio signala na govor i složeni audio signal, da bi na kraju bili detaljnije opisani dijelovi standarda koji se odnose samo na kodiranje složenog audio signala. Također su opisani i međusobni odnosi *profila*, *audio objekata* i *modula* koji su definirani u standardu. Na kraju su detaljnije opisani moduli koji su dio MPEG-4 General Audio dijela.

Predstavljeni pregled MPEG-4 standarda jasno ukazuje koliko je MPEG-4 složen standard. Odnosi se na razne vrste audio signala, razne bitske protoke, razne kvalitete kodiranog signala. Sama činjenica da od donošenja samog standarda pa do danas (prošlo je 5 godina), autoru još nije poznata niti jedna izvedba koja podržava sve *audio objekte* bilo kojeg od definiranih *profila*, dovoljno govori sama za sebe. MPEG-4 je vrlo složen standard i njegova aktualnost neće lako izbljediti. MPEG-4 će, po prognozama znanstvenika koji su učestvovali u njegovom stvaranju, biti aktualan do 2010. godine, a mnogi predviđaju i do 2015. godine.

U sljedećim poglavljima će biti predstavljena programska izvedba MPEG-4 koda pod nazivom **RAAC** (*RASIP Advanced Audio Coding*). U Poglavlju 6 biti će predstavljena izvedba osnovnih dijelova koda, a to su filterski slog, modul za kvantizaciju i Huffmanovu kompresiju sa pakiranjem u kodirani tok. U Poglavlju 7 će biti opisan psihoakustički dio, a u Poglavlju 8 algoritmi za obradu spektra, koji su najsofisticiraniji dio koda. Nakon ovih poglavlja, realizirani koder će biti evaluiran i ocijenjen.

Poglavlje 6

6 RAAC izvedba MPEG-4 općenitog audio koda

6.1 Uvod

U ovom poglavlju će biti započet detaljniji opis programske izvedbe MPEG-4 GA koda^{*} u vidu RAAC (*RASIP Advanced Audio Coding*) koda. Opis RAAC koda se proteže kroz sljedeća tri poglavlja.

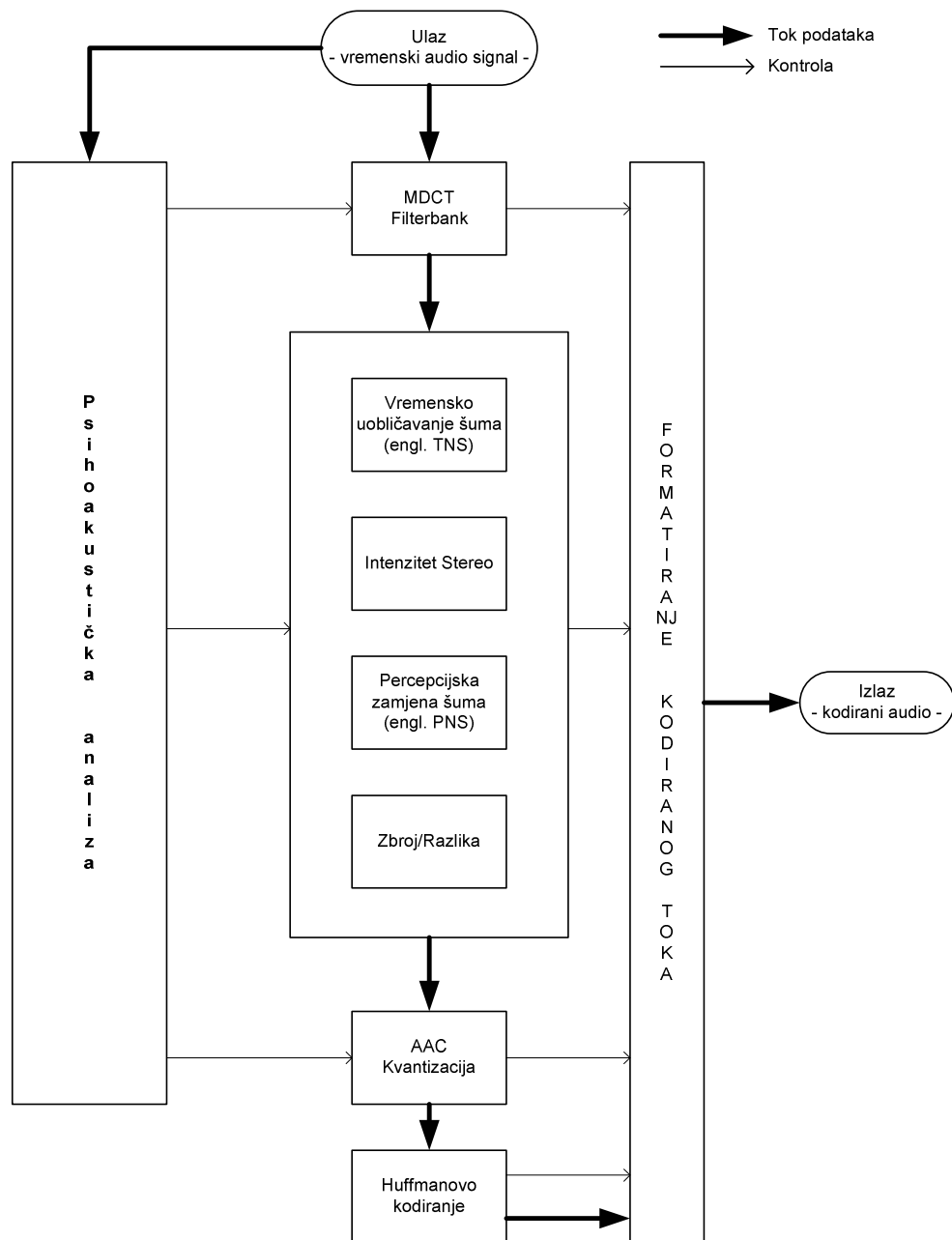
Cilj RAAC koda je izvedba *Low Complexity* (LC) audio objekta. LC audio objekt je dio 4 profila: *Main*, *Scalable*, *High Quality* i *Natural Audio*. LC audio objekt predstavlja centralni dio GA dijela MPEG-4 standarda jer su u njemu sadržani gotovo svi najbitniji moduli. Izvedba LC audio objekta predstavlja najvažniji dio izvedbe svakog GA koda, nakon koje se koder vrlo jednostavno nadograđuje sa nekim od ostalih audio objekata kao što su *Scalable* ili *Main* audio objekti, koji u stvari dodaju LC audio objektu jedan modul više.

Low Complexity audio objekt (prema Tablici 5.1) sadrži sljedeće module:

- filtarski slog (uključuje promjenu dužine bloka i oblika prozora),
- obrada spektra,
- kvantizacija (klasična AAC),
- entropijsko kodiranje (Huffman).

^{*} Riječ *koder* ovdje ima značenje istovjetno značenju riječi *enkoder* (onaj koji *kodira*), suprotno riječi *dekoder* (onaj koji *dekodira*).

Slika 6.1: RAAC programska izvedba MPEG-4 GA koda je limitirana na *Low Complexity* audio objekt. Cilj RAAC koda je kodiranje širokopojsnog, složenog audio signala na bitskim protocima od 32 kb/s/ch na više.



U ovom poglavlju će biti riječi o osnovnim dijelovima i funkcijama RAAC koda, a to su: filtarski slog, kvantizacija, entropijsko kodiranje i organizacija kodiranog toka. Psihoakustički modul koji je realiziran u RAAC koderu biti će posebno predstavljen u Poglavlju 7, dok će blokovi za obradu spektra biti objašnjeni u Poglavlju 8.

6.2 Proces kodiranja

Osnovne karakteristike RAAC koda su:

- kodiranje najviše dva audio kanala,
- raspon bitskih protoka od 32 kb/s/ch do 96 kb/s/ch,
- frekvencije uzorkovanja ulaznih signala od 8 kHz do 96 kHz.

Proces kodiranja možemo podijeliti u dvije faze:

- 1) inicijalizacija koda i
- 2) kodiranje blok po blok podataka.

Prva faza inicijalizacije koda uključuje:

- analizu ulaznog audio signala (informacije o frekvenciji uzorkovanja, broju kanala i sl.)
- inicijalizacija MDCT filtarskog sloga (proračun KBD i Sinus prozora, početna dužina i oblik prozora),
- inicijalizacija psihoakustičkog modula,
- definiranje broja i širina pojaseva kvantizatora obzirom na frekvenciju uzorkovanja (standardizirano),
- izbor zaglavlja (ADTS ili ADIF),
- proračun prosječne dužine bloka obzirom na zadani bitski protokol.

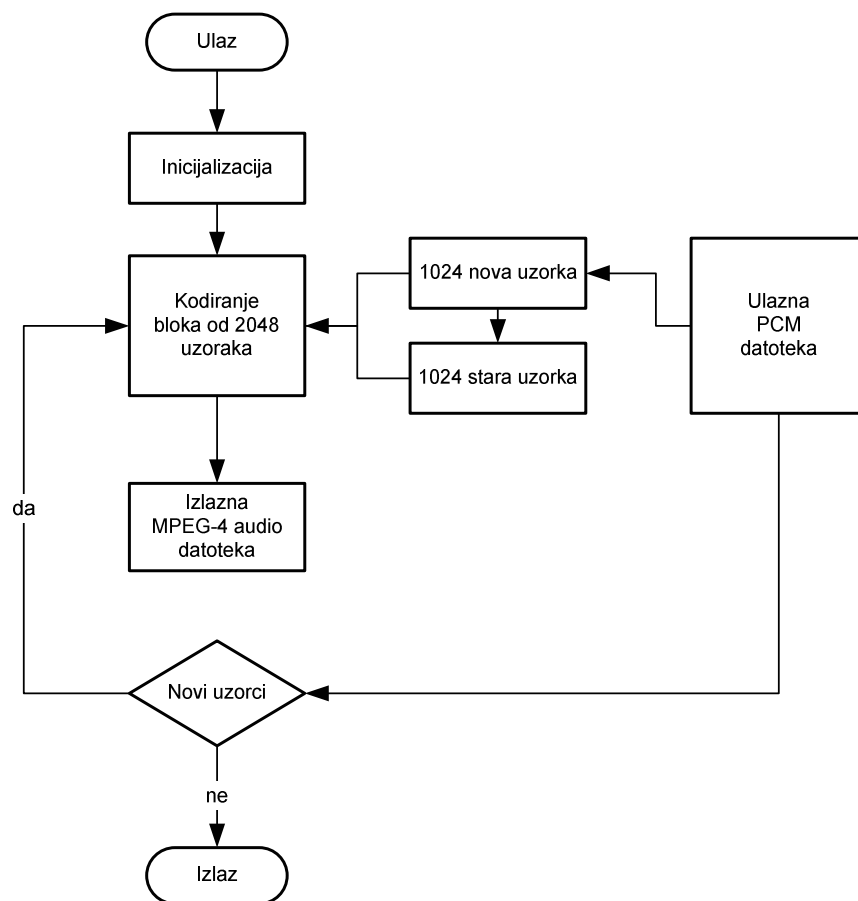
Druga faza kodiranja jednog bloka podataka uključuje:

- formiranje bloka od 2048 uzoraka preklapanjem,
- psihoakustičku analizu,
- određivanje oblika prozora i dužine bloka,
- MDCT transformaciju,
- limitiranje frekvencijskog raspona,
- grupiranje kratkih prozora,

- obrada spektra,
- kvantizacija i Huffmanovo kodiranje,
- ažuriranje *bit-rezervoara* i *bitova za popunjavanje*,
- zapis u kodirani tok.

Blok po blok se kodira sve dok se svi ulazni uzorci ne obrade. Na kraju kodiranog toka ne zapisuju se dodatni podatci.

Slika 6.2: Pojednostavljen proces kodiranja blok po blok. Nakon inicijalizacije, kodira se blok za blokom od 2048 vremenskih uzoraka audio signala. Nakon kodiranja svih uzoraka kodirani tok je formiran, a na njegov kraj se ne upisuju dodatni podatci.



6.3 Inicijalizacija koder

6.3.1 Ulazne veličine

U najjednostavnijom slučaju, RAAC koderu treba zadati samo ciljani bitski protok. Ipak, za potpuno iskorištavanje mogućnosti koder, potrebno je zadati i druge parametre.

Najprije je potrebno definirati graničnu (engl. *cut-off*) frekvenciju. Na ovaj način je moguće uštediti na bitskom protoku jer se frekvencije više od granične frekvencije postavljaju na nulu pa se za njihovo kodiranje troši znatno manje bita.

Osim granične frekvencije, RAAC koderu je potrebno zadati i frekvencijsko područje djelovanja pojedinih modula. Tako je za TNS, PNS i IS module potrebno zadati polazne frekvencije signala od kojih će biti primijenjeni. Preporučeni iznosi ovih frekvencija su predloženi pri detaljnom opisu modula u Poglavlju 8.

Na kraju, RAAC koderu je potrebno zadati i tip zaglavlja koje će se koristiti u kodiranom toku.

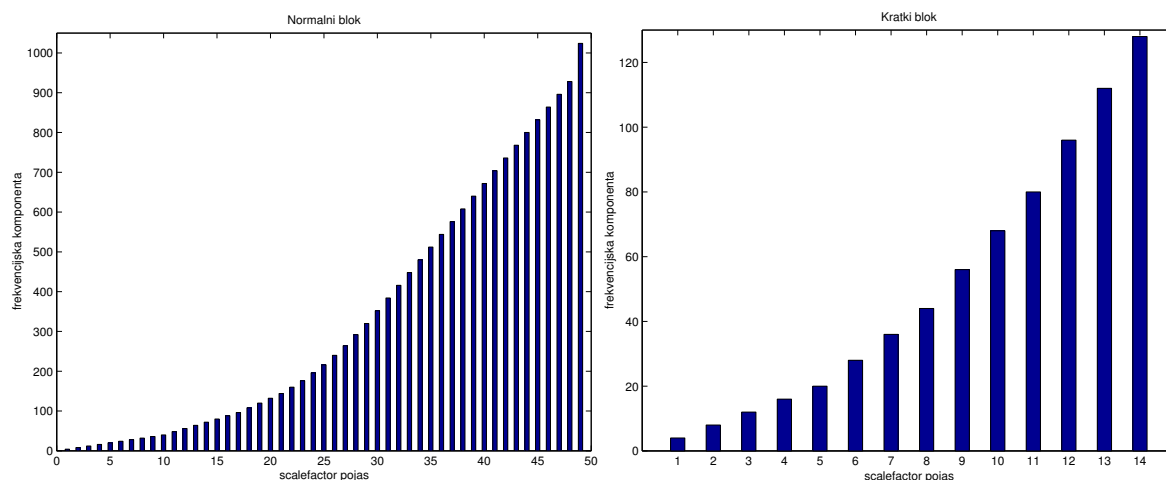
6.3.2 Analiza ulaznog signala

Jedine informacije o ulaznom signalu potrebne koderu su broj kanala i frekvencija uzorkovanja.

Na osnovu frekvencije uzorkovanja koder definira standardiziranu podjelu frekvencijskih komponenti prema pojasevima kvantizatora. Standardom su podržane sljedeće frekvencije uzorkovanja: 8 kHz, 11.025 kHz, 12 kHz, 16 kHz, 22.050 kHz, 24 kHz, 32 kHz, 44.1 kHz, 48 kHz, 64 kHz, 88.2 kHz i 96 kHz. Svaka od ovih frekvencija ima standardiziranu podjelu prema pojasevima kvantizatora. Primjer je dan na Slici 6.3.

Na osnovu broja kanala koder uključuje ili isključuje upotrebu pojedinih modula: npr. za jednokanalni mono signal isključeni su IS i MS moduli jer oni iskorištavaju sličnost između dva kanala.

Slika 6.3: Podjela signala prema pojasevima kvantizatora je standardizirana. Za frekvenciju uzorkovanja 44.1 kHz standard definira 49 pojaseva kvantizatora za blok od 1024 frekvencije, dok za kratki blok od 128 frekvencija broj pojaseva kvantizatora iznosi 14.



6.3.3 Prosječna veličina bloka

Na osnovu ulaznih informacija definira se prosječan broj bita koji je na raspolaganju za kodiranje svakog bloka od 1024 uzorka. Broj bita po bloku se računa prema Formuli 4.6.

Npr., neka je ulazni stereo signal uzorkovan frekvencijom 44.1 kHz, a zadani bitski protok 128000 bita po sekundi audio signala. Slijedi proračun prosječne veličine bloka:

$$\frac{128000[\text{bita} / \text{sec}] \times 1024[\text{uzorka} / \text{blok}]}{44100[\text{uzoraka} / \text{sec}]} = 2972[\text{bita} / \text{blok}] \quad (6.1)$$

Za jedan blok od 1024 uzorka ulaznog signala u prosjeku je raspoloživo 2972 bita.

Maksimalni bitski protok

Maksimalni bitski protok ovisi samo o frekvenciji uzorkovanja signala koji se želi kodirati. Računa se prema maksimalnoj veličini ulaznog spremnika za dekodiranje koji je limitiran i iznosi 6144 bita po bloku. Formula za maksimalni bitski protok je:

$$\max_bitrate = \frac{6144 \frac{bit}{blok}}{1024 \times \frac{uzorak}{blok}} \times f_{sampl.} \quad (6.2)$$

Npr. za frekvenciju uzorkovanja od 44.1 kHz, maksimalni bitski protok je 264 kb/s po kanalu. Obzirom da se već kod bitskog protoka od 80 kb/s po kanalu postiže *transparentna* kvaliteta kodiranog audio signala, maksimalni bitski protok ni u kom smislu nije ograničavajući faktor.

6.4 Kodiranje bloka od 1024 uzorka

6.4.1 Formiranje ulaznog bloka

Prema standardu, koder kodira blokove veličine 2048 vremenskih uzoraka. Ulazni blok od 2048 uzoraka se formira od 1024 uzorka iz zadnjeg kodiranog bloka, zajedno sa 1024 nova uzorka. Ovako spojeni uzorci čine jedna ulazni blok iz kojeg se kodiranjem dobiju 1024 frekvencijske komponente zapisane u kodirani tok. Svaka nova 1024 uzorka čuvaju se za kodiranje sljedećeg bloka.

6.4.2 Psihoakustička analiza

Prije početka kodiranja ulaznog bloka podataka, mora se izvršiti njegova psihoakustička analiza jer su rezultati psihoakustičke analize neophodni za usmjeravanje rada drugih blokova koda. Rezultati koje daje RAAC psihoakustička analiza su:

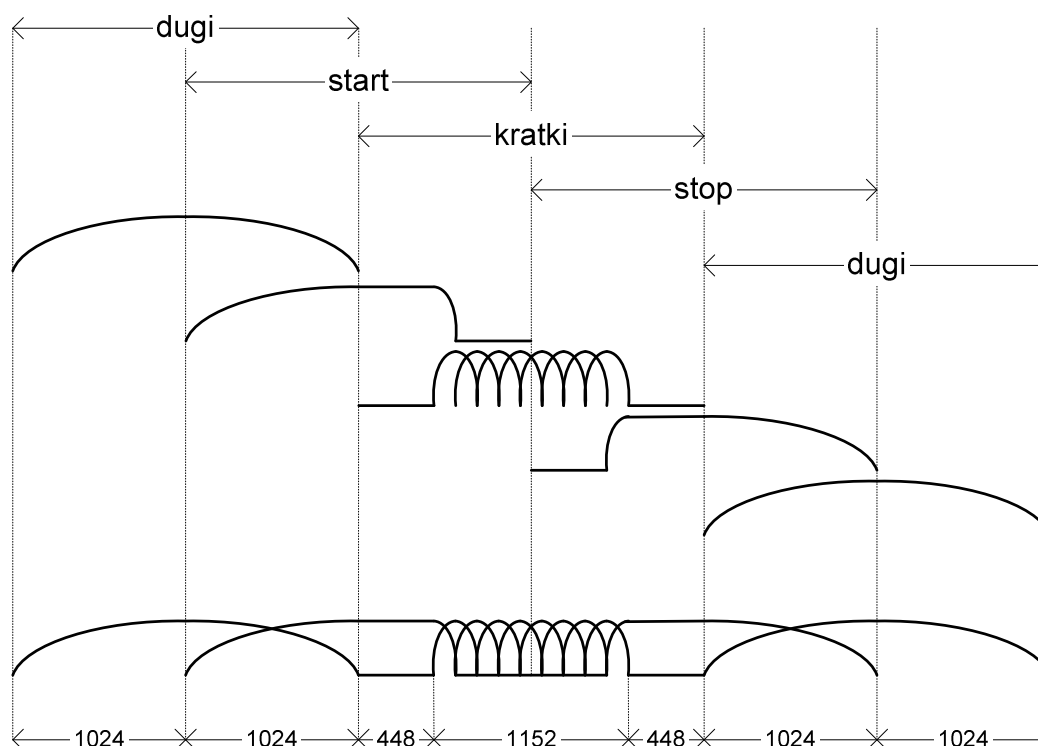
- 1) omjer *signal/prag maskiranja* za svaki pojas kvantizatora,
- 2) tip bloka obzirom na tranzijentnost,
- 3) tonalitet svake komponente signala,
- 4) iznos entropije.

Pošto je izvedbi modula za psihoakustičku analizu, kao jednom od bitnijih dijelova svakog, pa i RAAC koda, posvećeno cijelo Poglavlje 7, na ovome mjestu psihoakustička analiza neće biti detaljnije obrazlagana.

6.4.3 Izbor dužine bloka

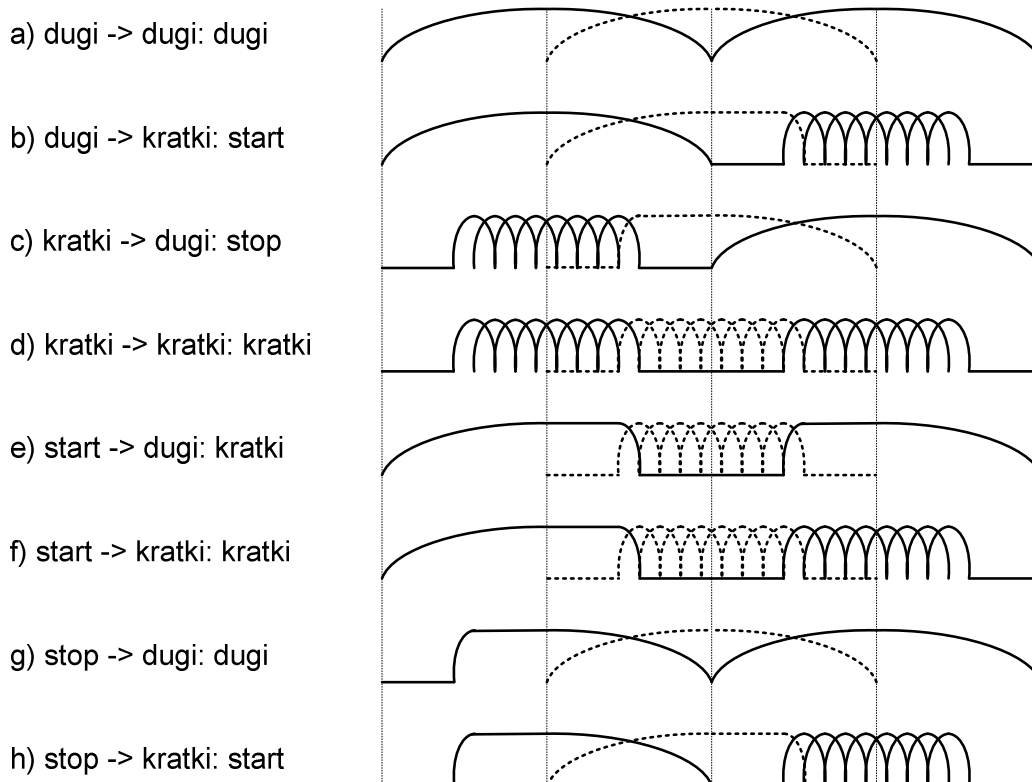
Na osnovu informacije o tranzijentnosti iz psihoakustičkog modula, RAAC koder bira dužinu trenutnog bloka za kodiranje između standardnog bloka od 2048 uzoraka i kratkog bloka koji se sastoji od 256 uzoraka (osam blokova zaredom).

Slika 6.4: Prijelaz sa standardnog (dugog) na kratki blok i sa kratkog na standardni. Između je neophodan prijelazni blok, u prvom slučaju start, a u drugom slučaju stop blok.



Treba napomenuti da se psihoakustička analiza računa za jedan blok unaprijed. Razlog je obavezno ubacivanje *prijelaznih* blokova između standardnih i kratkih blokova sa ciljem zadržavanja idealne rekonstrukcije signala kod filtarskog sloga. Tako standard, osim standardnog i kratkog bloka, definira i prijelazne *start* i *stop* blokove. Naime, da bi koder prešao sa standardnog na kratki blok potrebno je upotrijebiti *start* prijelazni blok, a da bi prešao sa kratkog nazad na standardni ubacuje se *stop* prijelazni blok (Slika 6.4). Zbog umetanja ovih *start-stop* blokova, psihoakustički modul mora na vrijeme – jedan blok unaprijed – signalizirati koderu da želi prebacivanje na kratki blok, tako da koder stigne ubaciti prijelazni blok.

Slika 6.5: Na osnovu prošlog bloka i budućeg bloka, određuje se oblik trenutnog bloka. Psihoakustički modul mora dati informaciju o bloku unaprijed.



Algoritam na osnovu kojeg RAAC koder određuje tranzijentnost bloka biti će opisan u Poglavlju 7 koje se bavi detaljnim predstavljanjem psihoakustičkog modula unutar kojeg se i računa tranzijentnost bloka.

Nakon što je psihoakustički modul definirao tranzijentnost sljedećeg bloka, koder odlučuje o obliku trenutnog bloka prema Slici 6.5.

6.4.4 Izbor oblika prozora

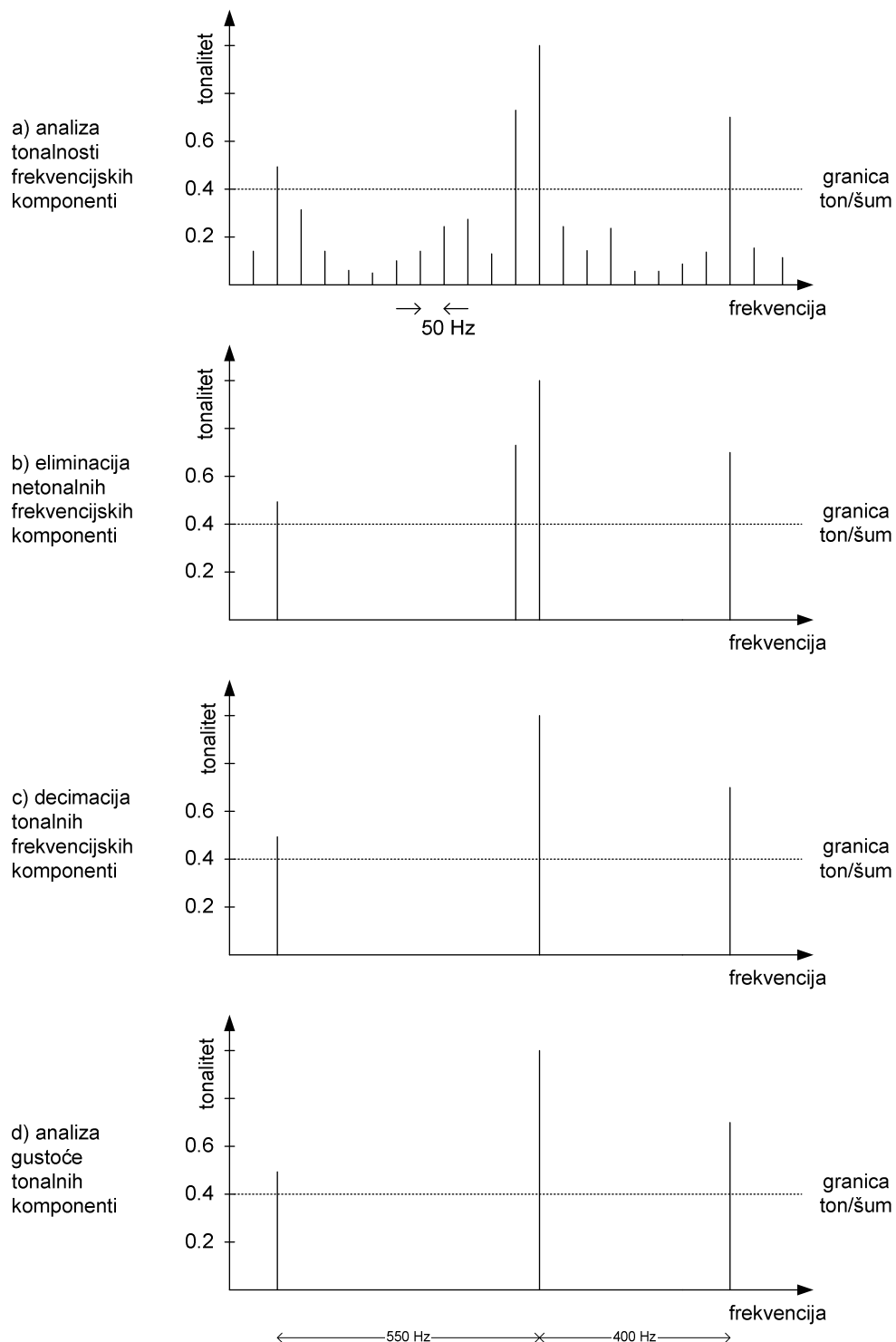
Nakon odluke o dužini bloka između standardnog, kratkog ili prijelaznog, koder određuje oblik prozora. Standardom su zadana dva oblika prozora: sinusni i KBD prozor. Oblik prozora treba prilagoditi analiziranom dijelu signala, točnije trenutnom bloku od 2048 uzoraka.

Na osnovu analize svojstava dva oblika prozora iz Poglavlju 5.3.1.1.1, odlučeno je da na oblik prozora kod RAAC koda presudno utječe spektralni sastav bloka, odnosno raspored tonalnih komponenti unutar bloka. Tonalnost svake komponente se proračunava unutar psihoakustičkog modula (Poglavlje 7), a odluka o obliku prozora se donosi na sljedeći način:

- 1) definira se prag koeficijenta tonalnosti iznad kojeg će frekvencijske komponente biti proglašavane tonalnim,
- 2) izdvajaju se tonalne komponente i vrši se njihova decimacija u rasponu od 70 Hz,
- 3) analizira se međusobni razmak preostalih tonalnih komponenti,
- 4) ukoliko su jake tonalne komponente razmaknute manje od 250 Hz, bira se sinusni prozor jer je potrebna bolja frekvencijska selektivnost,
- 5) u suprotnom slučaju, kada su tonalne frekvencije razmaknute za više od 250 Hz, bira se KBD prozor zbog boljeg prigušenja u nepropusnom pojasu, i jačeg koncentriranja energije frekvencija u jednu komponentu.

Prag kod kojeg se komponente označavaju tonalnim je određen eksperimentalno i iznosi od 0.2 do 0.4. Proces je ilustriran na Slici 6.6. Efikasnost algoritma pogledati u Poglavlju 9.

Slika 6.6: Algoritama za odabir oblika prozora za filterski slog. Nakon odbacivanja netonalnih frekvencijskih komponenti vrši se decimacija tonalnih komponenti unutar 70 Hz. Na kraju se analizira međusobna udaljenost preostalih tonalnih komponenti. Udaljenost manja od 250 Hz signalizira gust spektar pa se bira *sinus* prozor. Uz rjeđi spektar upotrebljava se KBD prozor. Granica na kojoj se komponenta proglašava tonalnom ili netonalnom je određena eksperimentalno (vidi algoritam za proračun tonaliteta u Poglavlju 7.3) i iznosi 0.4.



6.4.5 *Prozorovanje i MDCT transformacija*

Nakon što je definiran oblik trenutnog bloka obzirom na njegovu tranzijentnost i nakon što je izabrana funkcija prozora obzirom na spektralni sastav bloka, vrši se MDCT transformacija.

Prije same MDCT transformacije potrebno je izvršiti *prozorovanje* – množenje bloka od 2048 uzoraka sa vrijednostima vremenske *prozor* funkcije (za objašnjenje *prozora* pogledati Poglavlje 5.3.1.1.1).

Kako se blok podataka sastoji od 1024 podatka iz prošlog bloka i 1024 podatka iz novog bloka, zbog postizanja idealne rekonstrukcije, potrebno je svaku polovicu pomnožiti sa odgovarajućom prozor funkcijom: uzorci iz prošlog bloka se množe sa prozor funkcijom koja je bila upotrijebljena u prošlom bloku, dok se novi uzorci množe sa novo-izabranom prozor funkcijom (Slika 6.7).

Nakon *prozorovanja*, vremenski blok uzoraka je spreman za MDCT transformaciju. Ovisno o tipu prozora (standardni ili kratki), vrši se MDCT transformacija nad 2048 uzorka odjedanput ili se vrši osam uzastopnih kratkih MDCT transformacija, svaka nad 256 uzoraka (Slika 6.8). U svakom slučaju, na izlazu MDCT modula dobije se 2048 frekvencijskih komponenti, ali s obzirom da je:

$$f(n) = f(2047 - n), \quad 0 \leq n \leq 1023 \quad (6.3)$$

MDCT transformacija daje 1024 relevantne frekvencijske komponente.

MDCT transformacija je definirana sa:

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos \left[\frac{\pi}{N} (2k+1)(n+n_0) \right], \quad 0 \leq k \leq (N/2-1) \quad (6.4)$$

$$n_0 = \frac{1}{2} \left(\frac{N}{2} + 1 \right)$$

Realizacija MDCT transformacije u RAAC koderu je izvedena preko algoritma brze Fourierove transformacije (engl. *Fast Fourier Transform – FFT*) nad $N/4$ točaka. Realizacija je zasnovana na sljedećoj dekompoziciji:

$$\begin{aligned}
X(k) &= \mathbb{R} \left\{ \sum_{n=0}^{N-1} x(n) e^{-j\frac{\pi}{N}(2k+1)(n+n_0)} \right\} \\
&= \mathbb{R} \left\{ e^{-j\frac{\pi}{N}(2k+1)(n_0)} \sum_{n=0}^{N-1} x(n) e^{-j\frac{\pi}{N}(2k+1)(n)} \right\} \\
&= \mathbb{R} \left\{ e^{-j\frac{\pi}{N}(2k+1)(n_0)} \sum_{n=0}^{N-1} x(n) e^{-j\frac{\pi}{N}(2k)(n)} e^{-j\frac{\pi}{N}(1)(n)} \right\} \\
&= \mathbb{R} \left\{ e^{-j\frac{\pi}{N}(2k+1)(n_0)} \sum_{n=0}^{N-1} x(n) e^{-j\frac{\pi}{N}(1)(n)} e^{-j\frac{\pi}{N}(2k)(n)} \right\} \\
&= \mathbb{R} \left\{ e^{-j\frac{\pi}{N}(2k+1)(n_0)} \sum_{n=0}^{N-1} \left[x(n) e^{-j\frac{\pi}{N}(1)(n)} \right] e^{-j\frac{\pi}{N}(2k)(n)} \right\}
\end{aligned} \tag{6.5}$$

Obzirom na trodjelnost Formule, realizacija se sastoji od tri faze:

1.) pre-rotacija prema:

$$x^{pre}(n) = \left[x(n) e^{-j\frac{\pi}{N}(1)(n)} \right], \quad 0 \leq n \leq (N-1) \tag{6.6}$$

2.) FFT transformacija:

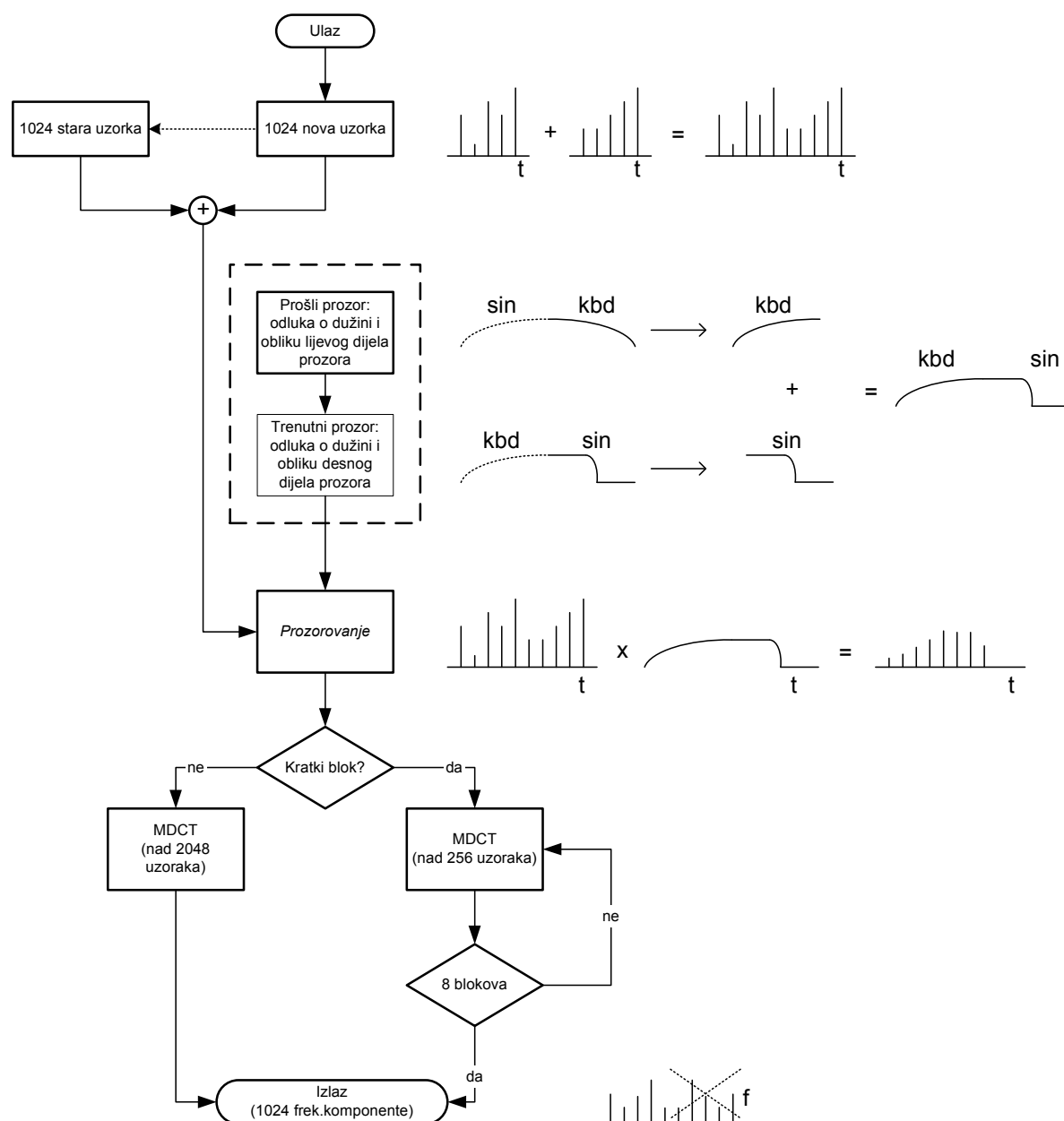
$$X^{FFT}(k) = \sum_{n=0}^{N-1} x^{pre}(n) e^{-j\frac{\pi}{N}(2k)(n)}, \quad 0 \leq k \leq \left(\frac{N}{2} - 1 \right) \tag{6.7}$$

3.) post-rotacija prema:

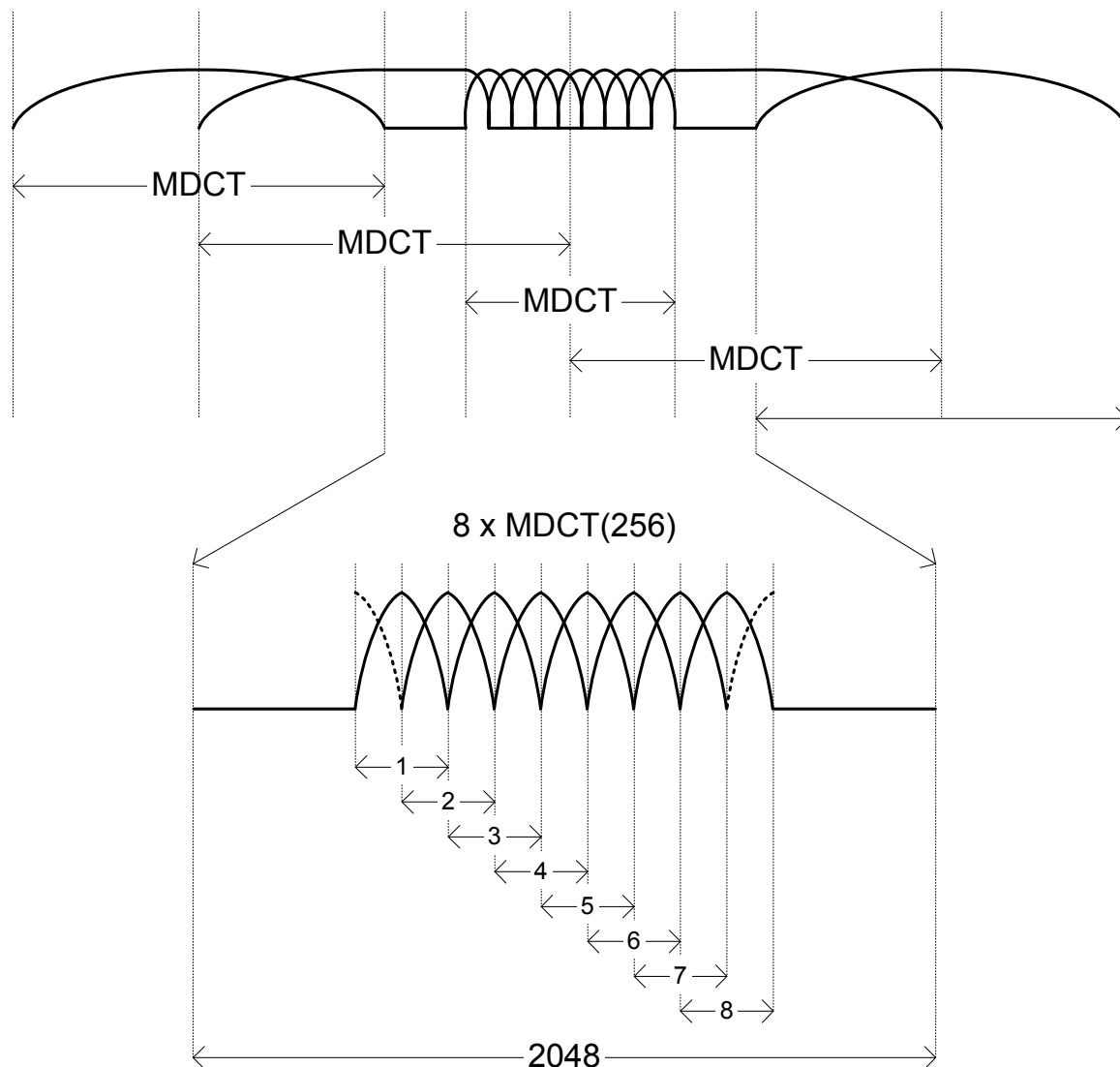
$$X(k) = e^{-j\frac{\pi}{N}(2k+1)(n_0)} X^{FFT}(k), \quad 0 \leq k \leq \left(\frac{N}{2} - 1 \right) \tag{6.8}$$

Realizacija je vrlo efikasna jer se zasniva na FFT transformaciji dužine $N/4$.

Slika 6.7: *Prozorovanje* i MDCT transformacija. Ulaz u MDCT transformaciju čine 1024 nova i 1024 stara uzorka, ukupno 2048. Prošli prozor određuje tip prozora za stara 1024 uzorka, dok se nova 1024 uzorka množe sa prozorom definiranim psihoakustičkom analizom tekućeg bloka. Nakon *prozorovanja*, ovisno o tipu bloka vrši se jedna MDCT transformacija nad 2048 uzoraka ili 8 uzastopnih transformacija, svaka nad 256 polovično prekrivenih uzoraka. U oba slučaja, rezultat su 1024 nove frekvencijske komponente.



Slika 6.8: Standardom su definirane dvije dužine MDCT transformacije. Standardni blok podrazumijeva jednu MDCT transformaciju nad cijelim blokom od 2048 uzoraka, dok se unutar kratkog bloka vrši 8 MDCT transformacija, svaka nad 256 uzoraka. Rezultat i jedne i druge transformacije su 1024 frekvencijske komponente. Kratkim blokom se smanjuje vremenska rezolucija signala, pa se kvantizacijski šum ne širi preko uobičajenog intervala od 2048 uzoraka, nego unutar 8 puta manjeg intervala. Na ovaj način se sprečava efekt pred-jeke koji je vrlo izražen kod tranzijentnih signala.



6.4.6 Limitiranje frekvencijskog raspona

Kod kodiranja složenih audio signala uzorkovanih frekvencijama većim od 32 kHz, većinom se frekvencijski raspon limitira na određeni iznos manji od *Nyquistove* frekvencije $\frac{f_s}{2}$. Tako se najčešće upotrebljavani signal frekvencije uzorkovanja 44.1 kHz obično limitira na 16 do 18 kHz. Danas je općeprihvaćeno mišljenje da se limitiranjem frekvencijskog opsega na 16 kHz ne narušava kvaliteta signala, odnosno da je nemoguće razlučiti između originalnog i limitiranog signala, dakako ako se radi o složenom audio signalu. Kako se limitiranjem frekvencijskog raspona značajan broj frekvencijskih komponenti postavlja na nulu, dolazi i do osjetne uštede u bitskom protoku jer se nule kodiraju sa beznačajnom potrošnjom bita. Imajući ovo u vidu, RAAC koder vrši limitiranje frekvencijskog opsega na 15 kHz. Limitiranje se vrši na način da se frekvencijske komponente više od 15 kHz na izlazu iz MDCT modula brišu.

Proračun se vrši na sljedeći način: proračuna se pozicija frekvencijske komponente koja odgovara frekvenciji limita, pa se sve komponente iznad ove granice postavljaju na nulu. Proračun se vrši posebno za standardni, a posebno za kratki prozor.

$$\begin{aligned} bin_dugacki &= f_{cutoff} \times \frac{1024}{\left(\frac{f_{sampl.}}{2}\right)} \\ bin_kratki &= f_{cutoff} \times \frac{128}{\left(\frac{f_{sampl.}}{2}\right)} \end{aligned} \quad (6.9)$$

Npr. kod frekvencije uzorkovanja od 44.1 kHz, i željene granične frekvencije od 15 kHz, brišu se sve komponente od 697 do 1024 kod standardnog bloka, i od 87 do 128 kod kratkog bloka. Ušteda je značajna, dok je razlika u kvaliteti signala zanemariva u velikoj većini slučajeva.

6.4.7 Obrada spektra

Moduli za obradu spektra definirani unutar *Low Complexity* audio objekta koji su izvedeni u RAAC koderu uključuju: vremensko uobličavanje šuma (engl. *Temporal Noise Shaping* – *TNS*), percepcijska zamjena šuma (engl. *Perceptual Noise Substitution* – *PNS*), intenzitet-

stereo (engl. *Intensity-stereo* – *IS*) i zbroj/razlika (engl. *Summ/difference* ili engl. *Mid/side-MS*). Izvedbi ovih modula posvećeno je Poglavlje 8.

6.4.8 Kvantizacija

Nakon što su MDCT transformacijom proračunate 1024 frekvencijske komponente, i nakon što je izvršena obrada dobivenih komponenti putem TNS, PNS, IS i MS modula, vrši se njihova kvantizacija i Huffmanovo kodiranje.

Modul za kvantizaciju je dio koda koji presudno utječe na kvalitetu koda. Kvantizacija je proces u kojem se dio informacija bespovratno gubi pa je sama jako bitna sa stanovišta konačne kvalitete kodiranog signala.

Ulaz u modul za kvantizaciju čine:

- 1) 1024 frekvencijske komponente,
- 2) niz omjera *signal/prag maskiranja* iz psihoakustičkog modula,
- 3) broj bita koji je na raspolaganju za kodiranje bloka od 1024 frekvencijske komponente (zajedno sa ostalim sporednim informacijama),
- 4) raspored pojaseva kvantizatora,
- 5) 11 Huffmanovih tablica za kodiranje spektra i jedna Huffmanova tablica za kodiranje kvantizatora.

Rezultat kvantizacije se sastoji od:

- 1) Huffmanom kodiranih kvantizatora (jedan za svaki pojas kvantizatora),
- 2) Huffmanom kodiranih kvantiziranih vrijednosti frekvencijskih komponenti,
- 3) rasporeda Huffmanovih tablica kojima je kodiran spektar.

Proces kvantizacije je u RAAC koderu realiziran preko metode *dvostruke petlje*. Dvije petlje – unutrašnja i vanjska – imaju različite uloge: unutrašnja petlja provjerava omjer *signal/prag maskiranja* i shodno njegovom iznosu određuje stupanj kvantizacije pojedinih pojaseva kvantizatora, dok vanjska petlja prebrojava broj bita potreban za kodiranje cijelog bloka i u slučaju da je broj bita prevelik, pojačava stupanj kvantizacije svih pojaseva odjednom.

Formula za dekvantizaciju je definirana standardom (Formula 5.9).

Maksimalna amplituda za f_{kv} je 8191 (1FFF hex – maksimalno 17 bita).

Iz formule za dekvantizaciju dobije se formula za kvantizaciju:

$$f_{kv} = \text{int} \left[\left(f \times 2^{\frac{1}{4}(\text{kvantizator}-100)} \right)^{\frac{3}{4}} \right] \quad (6.10)$$

Da bi Formula 6.10 bila pogodna za metodu *dvostruke petlje* potrebno je kvantizator podijeliti u dva dijela: prvi koji je zajednički za sve pojaseve kvantizatora i drugi koji je specifičan za svaki posebni pojas kvantizatora. Također, predznaci ovih razdijeljenih kvantizatora trebaju biti suprotni, jer: povećanje zajedničkog treba pojačati kvantizaciju (svih pojaseva), dok povećanje pojedinačnog treba smanjiti kvantizaciju (samo pojasa na koji se odnosi). Formula koja zadovoljava ove zahtjeve je:

$$f_{kv} = \text{int} \left[\left(f \times 2^{\frac{1}{4}(\text{pojasni_kvantozator}-\text{zajednicki_kvantizator})} \right)^{\frac{3}{4}} \right] \quad (6.11)$$

Logično je da:

$$\text{pojasni_kvantozator} - \text{zajednicki_kvantizator} = \text{kvantizator} - 100 \quad (6.12)$$

Proces kvantizacije započinje određivanjem početne vrijednosti za *zajednicki_kvantizator*.

Početne vrijednosti za *pojedinačni_kvantizator* za svaki pojas su *nula*.

Početna vrijednost za *zajednicki_kvantizator* se određuje metodom sukcesivne aproksimacije, a s obzirom na maksimalnu frekvencijsku komponentu unutar bloka od 1024 komponente i obzirom na broj bita koji je na raspolaganju za kodiranje jednog bloka. Početna vrijednost *zajednicki_kvantizator* mora biti takva da kodiranje bloka zahtijeva više bita nego što je dozvoljeno. Na ovaj način se ostavlja prostora *dvostrukoj petlji* da reorganizira kvantizatore poštujući zadane omjere *signal/šum* iz psihoakustičkog modula, na način da je kodiranje bloka moguće sa zadanim brojem bita.

Proračun vrijednosti *zajednicki_kvantizator* počinje od određivanja maksimalne frekvencijske komponente unutar bloka. Iz maksimalne frekvencijske komponente pomoću Formule 6.10 slijedi formula za proračun početne vrijednosti za *zajednicki_kvantizator*:

$$zajednicki_kvantizator_pocetni = \frac{4}{3} \times \frac{4}{1} \times \log_2 \left(\frac{\max_frek^{\frac{3}{4}}}{8191} \right) \quad (6.13)$$

Sa ovom vrijednošću se ulazi u proces sukcesivne aproksimacije kojim se određuje vrijednost *zajednicki_kvantizator* prije ulaska u *dvostruku petlju*. Proces je opisan sljedećim pseudokodom:

```

zajednicki_kvantizator = zajednicki_kvantizator_pocetni
korak = 64
smjer = porast

DO
{
    kvantiziraj_spektar()
    potrebno_bita = prebroj_bite()

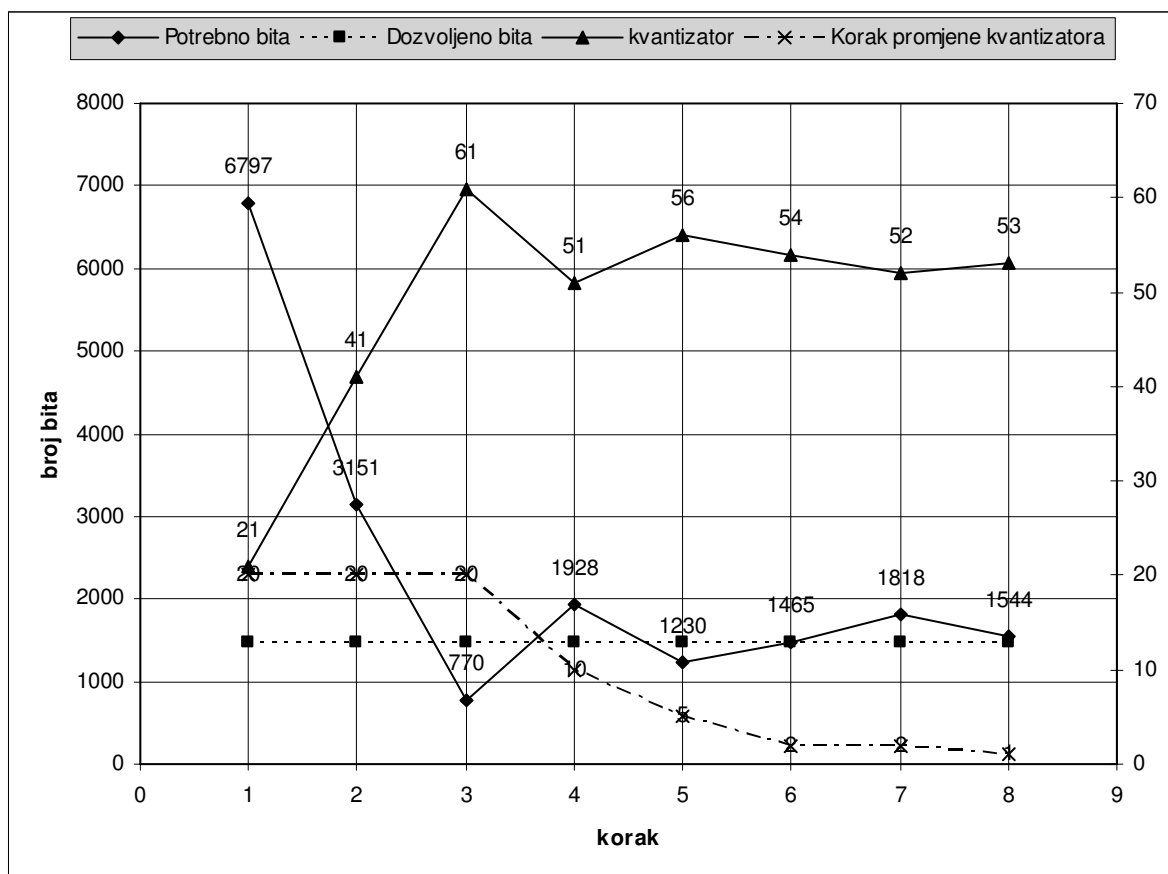
    IF (potrebno_bita < dozvoljeno_bita AND korak != 1)
    {
        IF(smjer == porast)
        {
            korak = korak/2
            smjer = pad
        }

        zajednicki_kvantizator -= korak
    }
    ELSE
    {
        IF(smjer == pad)
        {
            korak = korak/2
            smjer = porast
        }

        zajednicki_kvantizator += korak
    }
}
WHILE (korak != 1)

```

Slika 6.9: Primjer određivanja vrijednosti *zajednicki_kvantizator_pocetni*. Sukcesivnom aproksimacijom, broj bita se približava zadanom (u ovom primjeru 1485) sve dok korak promjene ne dosegne 1. Na kraju procesa *zajednicki_kvantizator_pocetni* ima vrijednost (u ovom primjeru 53) koja zajedno sa vrijednostima *pojedinačni_kvantizator* (sve nule) kodira blok sa nešto više bita nego što je dozvoljeno (ovdje 1514). Sa ovim vrijednostima se započinje proces kodiranja u *dvostrukoj petlji*.



Dakle, sukcesivnom aproksimacijom *zajednicki_kvantizator_pocetni* konvergira vrijednosti s kojom se blok kodira sa nešto više bita nego što je zadano (primjer prikazan na Slici 6.9). Nakon što je određena vrijednost *zajednicki_kvantizator_pocetni* pristupa se procesu *dvostruke petlje*.

Unutarnja petlja vrši kvantizaciju svih pojaseva i proračunava šum unesen kvantizacijom. Ukoliko je šum određenog pojasa veći od dozvoljenog – koji je proračunat u psihoakustičkom

modulu – povećava se vrijednost *pojedinačni_kvantizator* za spomenuti pojas. Na ovaj način se šum pojasa smanjuje, ali se povećava broj bita potreban za kodiranje bloka.

Nakon što su svi pojasevi provjereni i *pojačani*, unutarnja petlja je završena. Slijedi proračun broja bita koji je potreban za kodiranje bloka sa novim iznosima *pojedinačni_kvantizator*.

Ukoliko je broj bita veći od dozvoljenog, povećava se vrijednost *zajednički_kvantizator* i ponovo se započinje unutarnja petlja koja ponovo poboljšava omjer *signal/šum* u pojedinim pojasevima. Proces se iterativno ponavlja sve dok se ne postigne kodiranje bloka manjim brojem bita nego što je zadano.

Proces *dvostruke petlje* je prikazan sljedećim pseudo-kodom:

```
DO
{
    DO
    {
        kvantiziraj_spektar()
        šum_pojasa = proračunaj_šum_pojasa()

        IF(šum_pojasa > dozvoljeni_šum)
        {
            pojedinačni_kvantizator ++
        }
    }
    WHILE(svi_pojasevi_proracunati)

    potrebno_bita = prebroj_bite()

    IF(potrebno_bita > dozvoljeno_bita)
    {
        zajednički_kvantizator ++
    }
}
WHILE(potrebno_bita > dozvoljeno_bita)
```

6.4.9 Huffmanovo kodiranje

Nakon što je završen proces određivanja kvantizatora (dvostruka petlja) potrebno je prebrojati bite koji su potrebni da bi se blok podataka kodirao sa zadanim kvantizatorima. Prebrojavanje bita je *de facto* potpuno kodiranje bloka, s tom razlikom da se unaprijed ne zna hoće li izvršeno kodiranje zadovoljiti zadani broj bita i time biti usvojeno ili će biti izvršeno novo kodiranje u sljedećem prolasku kroz vanjsku petlju u primjeni metode *dvostruke petlje*.

Prebrojavanje bita se može podijeliti u četiri koraka:

- 1) izbor optimalnih Huffmanovih tablica za svaki pojas kvantizatora,
- 2) kodiranje izabranih Huffmanovih tablica,
- 3) kodiranje spektra pomoću izabranih Huffmanovih tablica,
- 4) kodiranje kvantizatora.

Izuzmemo li kodiranje Huffmanovih tablica, prebrojavanje bita se može svesti na samo Huffmanovo kodiranje: kvantizatora i spektralnih komponenti.

Analizirajmo spomenuta četiri koraka kodiranja bez gubitaka.

Korak 1

Za svaki pojas kvantizatora prvo se pronalazi Huffmanova tablica pomoću koje se zadani pojas kodira sa najmanjim brojem bita. Za uspješno izvršenje ovog procesa, potrebno je kodirati pojas uzastopno sa nekoliko tablica, a zatim izabrati tablicu koja daje najbolji rezultat. Pošto izbor tablice ovisi o maksimalnoj vrijednosti koju tablica može kodirati (Tablica 6.1), prvo se za svaki pojas kvantizatora određuje maksimalni element. Nakon što su na osnovu maksimalnog elementa određene moguće tablice, prebrojava se broj bita koji je potreban za kodiranje pojasa sa svakom od mogućih tablica. Selektira se tablica koja daje najmanji broj bita, odnosno najbolju kompresiju. Proces se ponavlja za sve pojaseve kvantizatora.

Korak 2

Nakon što su definirane, Huffmanove tablice se kodiraju za prijenos u kodiranom toku. U kodirani tok se upisuje broj tablice, a zatim broj uzastopnih pojaseva kvantizatora koji se kodiraju izabranom tablicom.

Korak 3

Nakon definiranja i kodiranja Huffmanovih tablica, pristupa se kodiranju spektra. Iz definiranih Huffmanovih tablica vade se kodne riječi koje odgovaraju kvantiziranim spektralnim komponentama, i pamte za spremanje u kodirani tok. Ovisno o izabranoj tablici, kodiraju se dvije ili četiri riječi zajedno. Proces teče na sljedeći način:

- na osnovu kvantiziranih frekvencijskih komponenti $q[i]$ proračunava se vrijednost `indeks` koja predstavlja redni broj riječi unutar izabrane Huffmanove tablice,
- riječ unutar tablice pod rednim brojem `indeks` se sprema u kodirani tok.

Tablica 6.1: Parametri Huffmanovih tablica za kodiranje kvantiziranih frekvencijskih komponenata. Tablice se razlikuju po maksimalnoj vrijednosti koju mogu zapisati, po načinu kodiranja negativnih vrijednosti i po broju spektralnih komponenti koje se zajedno kodiraju. Poseban slučaj predstavljaju tablice 0 i 11: tablica 0 signalizira da su u izabranom pojasu sve vrijednosti nula, dok tablica 11 omogućava poseban vid kodiranja vrijednosti većih od 15.

Redni broj Huffmanove tablice	Predznak posebno u kodiranom toku	Broj riječi	Maksimalna apsolutna vrijednost
0	-	-	0
1	ne	4	1
2	ne	4	1
3	da	4	2
4	da	4	2
5	ne	2	4
6	ne	2	4
7	da	2	7
8	da	2	7
9	da	2	12
10	da	2	12
11	da	2	(15) 8191

Formula po kojoj se proračunava redni broj kodne riječi unutar Huffmanove tablice definirana je standardom (procesom dekodiranja), a razlikuje se od tablice do tablice (Tablica 6.2).

Ukoliko kodirani pojas sadrži vrijednosti veće od 15, kodiranje se provodi na poseban način, a u kodiranom toku se to signalizira izborom Huffmanove tablice broj 11. Vrijednosti manje ili jednake 15 se kodiraju prema formuli iz Tablice 6.2. Ako je vrijednost veća od 15, za kodiranje se upotrebljava posebna metoda, koja se u kodiranom toku signalizira sa kodiranjem broja 16 putem tablice 11. Kada dekodir dekodira broj 16 iz tablice 11, to mu je znak da u kodiranom toku slijede dvije vrijednosti koje na poseban način kodiraju dvije spektralne komponente koje su veće od 15.

Svaka od dvije spektralne komponente se dekodira na sljedeći način: prvo se iz kodiranog toka zaredom uzimaju jedinice koliko ih ima (broj jedinica označimo sa N). Nakon jedinica zapisana je jedna nula, nakon koje slijedi niz od $N+4$ bita koji predstavljaju cijeli broj (označimo ga sa L). Formula kojom se dekodira kvantizirana spektralna komponenta glasi:

$$q = 2^{N+4} + L \quad (6.14)$$

Tablica 6.2: Formule za proračun rednog broja kodne riječi unutar Huffmanove tablice. Po dvije tablice dijele istu formulu. Tablica 11 formulom kodira vrijednosti do 15, a za veće vrijednosti se primjenjuje posebna metoda.

Redni broj Tablice	Formula za redni broj kodne riječi unutar tablice
1, 2	$\text{index} = 27q[i] + 9*q[i+1] + 3*q[i+2] + q[i+3] + 40$
3, 4	$\text{index} = 27*ABSq[i] + 9*ABSq[i+1] + 3*ABSq[i+2] + ABSq[i+3]$
5, 6	$\text{index} = 9*q[i] + q[i+1] + 40$
7, 8	$\text{index} = 8*ABSq[i] + ABSq[i+1]$
9, 10	$\text{index} = 13*ABSq[i] + ABSq[i+1]$
11	$\text{index} = 17*ABSq[i] + ABSq[i+1]$

Kodiranje se izvodi preko sljedeće formule:

$$\text{kod} = q + (2^M - 1) \times 2^{M+5} - 2^{M+4} \quad (6.15)$$

gdje je M proračunat na sljedeći način:

```

M = -1
x = q/16
DO
{
    M = M+1
    x = x/2
}
WHILE (x >= 1)

```

Tablica 6.3: Primjeri kodiranja vrijednosti većih od 15 pomoću posebne metode preko Huffmanove tablice 11. Pošto je najveća vrijednost koje treba kodirati 8191, najduža riječ u kodiranom toku ima 21 bit.

Kodirani tok	N	L	Dekodirani broj q
0 0000	0	0	16
0 1111	0	15	31
1 0 11111	1	31	63
11111111 0 11111111111111	8	4095	8191

Proces kodiranja po dvije i po četiri kvantizirane vrijednosti se ponavlja sve dok se sve 1024 komponente spektra ne zapišu pomoću kodnih riječi iz Huffmanovih tablica.

Korak 4

Osim rednih brojeva Huffmanovih tablica i Huffmanovih kodnih riječi, za dekodiranje je neophodno u kodiranom toku prenijeti i kvantizatore pomoću kojih se kvantizirane vrijednosti spektra dekvantiziraju. Za svaki pojas se prenosi jedan kvantizator.

Kvantizatori se kodiraju pomoću jedne, standardom propisane, Huffmanove tablice, koja omogućava kodiranje vrijednosti od 0 do 256. Pošto se vrijednost susjednih kvantizatora od pojasa do pojasa ne mijenjaju skokovito, kvantizatori se prvo diferencijalno kodiraju. Zatim se vade kodne riječi iz Huffmanove tablice koje se spremaju u kodirani tok. Maksimalna razlika dva susjedna kvantizatora je 60, pa tablica ima 120 kodnih riječi. Prvi kvantizator se diferencijalno kodira prema `global_gain` elementu kodiranog toka. Kako se za `global_gain` obično bira vrijednost prvog kvantizatora, onda je prvi diferencijalno proračunati kvantizator nula.

Nakon ova četiri koraka, završen je proces kodiranja jednog bloka podataka, a podatci su spremni za zapis u kodirani tok.

6.4.10 Grupiranje kratkih prozora

Broj kvantizatora kod niza od osam kratkih prozora je znatno veći nego kod standardnog bloka. Svaki kratki prozor je zasebna cjelina koja sadrži određeni broj kvantizatora koji je ovisan o frekvenciji uzorkovanja. Kako se za svaki pojas u kodiranom toku prenosi jedan kvantizator onda je količina informacija za osam kratkih prozora prevelika da bi se postigla kvaliteta kodiranja kao kod standardnog prozora.

Za frekvenciju uzorkovanja 44100 Hz broj pojaseva kvantizatora za standardni prozor je 49, a za kratki 14. Za osam kratkih prozora broj pojaseva kvantizatora iznosi 112.

Da bi se smanjio broj kvantizatora koji se prenose za osam kratkih prozora unutar bloka, standardom je definirano grupiranje kratkih prozora. Grupirani prozori dijele iste kvantizatore. Grupirati se može samo niz susjednih prozora, a mogući su svi načini grupiranja. Za uštedu bita najefikasnije je udruživanje svih osam prozora u jednu grupu (1x8 slučaj), dok je percepcijski najbolje da svaki prozor ima svoje zasebne kvantizatore (8x1 slučaj). Zadatak koda je da na određeni način napravi balans između ova dva zahtjeva.

RAAC koder grupira kratke prozore na osnovu njihove energije. Za svaki prozor se proračunava ukupna energija frekvencijskih komponenti, koja se zatim uspoređuje sa

energijom susjednog prozora. Ukoliko je razlika u energijama ispod praga k prozori se udružuju u jednu grupu, nakon čega se prelazi na usporedbu sljedećeg para susjednih prozora. Prozori se udružuju u jedinstvenu grupu sve dok se ne uoči značajan skok u energiji između dva prozora. Tada se prelazi na formiranje nove grupe prozora sa sličnom energijom. Proces se ponavlja sve dok se ne izanalizira svih osam prozora. Prag k je eksperimentalno određen i iznosi od 1.5 do 3.

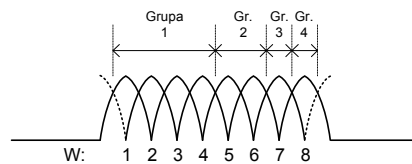
Algoritam *energetske usporedbe* je izabran zato što se počelo od pretpostavke da stacionarni signali koji imaju ujednačenu energiju od bloka do bloka (spektar je sličan, ne pojavljuju se nove frekvencije) imaju i slične percepcijske karakteristike, pa se njihovim udruživanjem signal neće previše percepcijski degradirati zbog upotrebe istih kvantizatora.

Nakon udruživanja niza prozora potrebno je, za svaki pojas kvantizatora, odlučiti od kojeg od prozora izabrati prag maskiranja kao zajednički za sve udružene prozore. RAAC koder ovdje upotrebljava metodu *najgoreg slučaja*: za svaki pojas se uspoređuje prag maskiranja svih udruženih prozora, a bira se najniži prag maskiranja. Na ovaj se način poštuju principi psihoakustike, s tim da će neki pojasevi nepotrebno biti kvantizirani kvalitetnije nego što je preporučeno psihoakustičkim modulom.

U kodiranom toku su spektralni koeficijenti ispremješteni prema izabranom grupiranju. Način premještanja je prikazan na Slici 6.10. Na ovaj način se grupiraju slični koeficijenti pa je Huffmanovo kodiranje efikasnije.

Slika 6.10: Redoslijed spektralnih komponenti u kodiranom toku kod grupiranih kratkih prozora. Dan je primjer kada su prozori grupirani u četiri grupe dužina 4, 2, 1 i 1 prozor. Kod frekvencije uzorkovanja od 44.1 kHz, grupiranjem se broj kvantizatora smanjuje sa 112 na 56.

Grupiranje kratkih prozora u četiri grupe dužina 4, 2, 1 i 1 prozora:



Pripadajući redoslijed spektralnih komponenti:

→

```

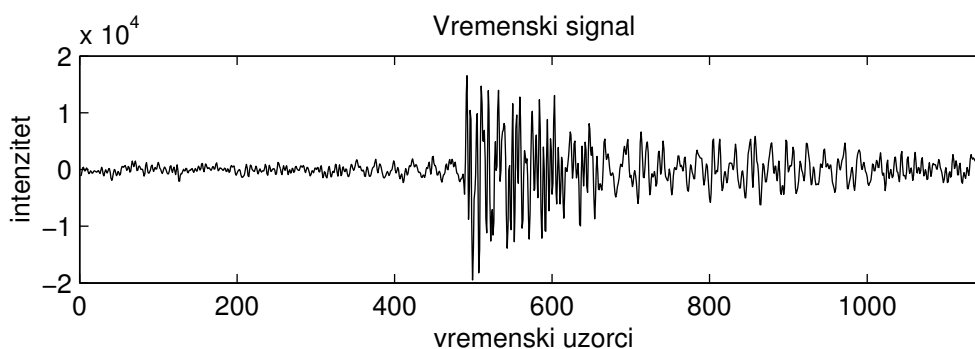
K[sf1w1] K[sf1w2] K[sf1w3] K[sf1w4]
K[sf2w1] K[sf2w2] K[sf2w3] K[sf2w4]
.
.
.
K[sfnw1] K[sfnw2] K[sfnw3] K[sfnw4]
K[sf1w5] K[sf1w6]
K[sf2w5] K[sf2w6]
.
.
.
K[sfnw5] K[sfnw6]
K[sf1w7]
K[sf2w7]
.
.
.
K[sfnw7]
K[sf1w8]
K[sf2w8]
.
.
.
K[sfnw8]

```

$K[sf_{xwy}]$ - niz spektralnih komponenti unutar pojasa kvantizatora 'x' prozora 'y'

Za ilustraciju algoritma grupiranja izabran je tipičan tranzijentni dio vremenskog signala prikazan na Slici 6.11. Promjenom praga k u odnosu energija susjednih pojaseva kod kojeg se donosi odluka o započinjanju nove grupe, dobiveni su različiti načini grupiranja. Rezultat je prikazan u Tablici 6.4.

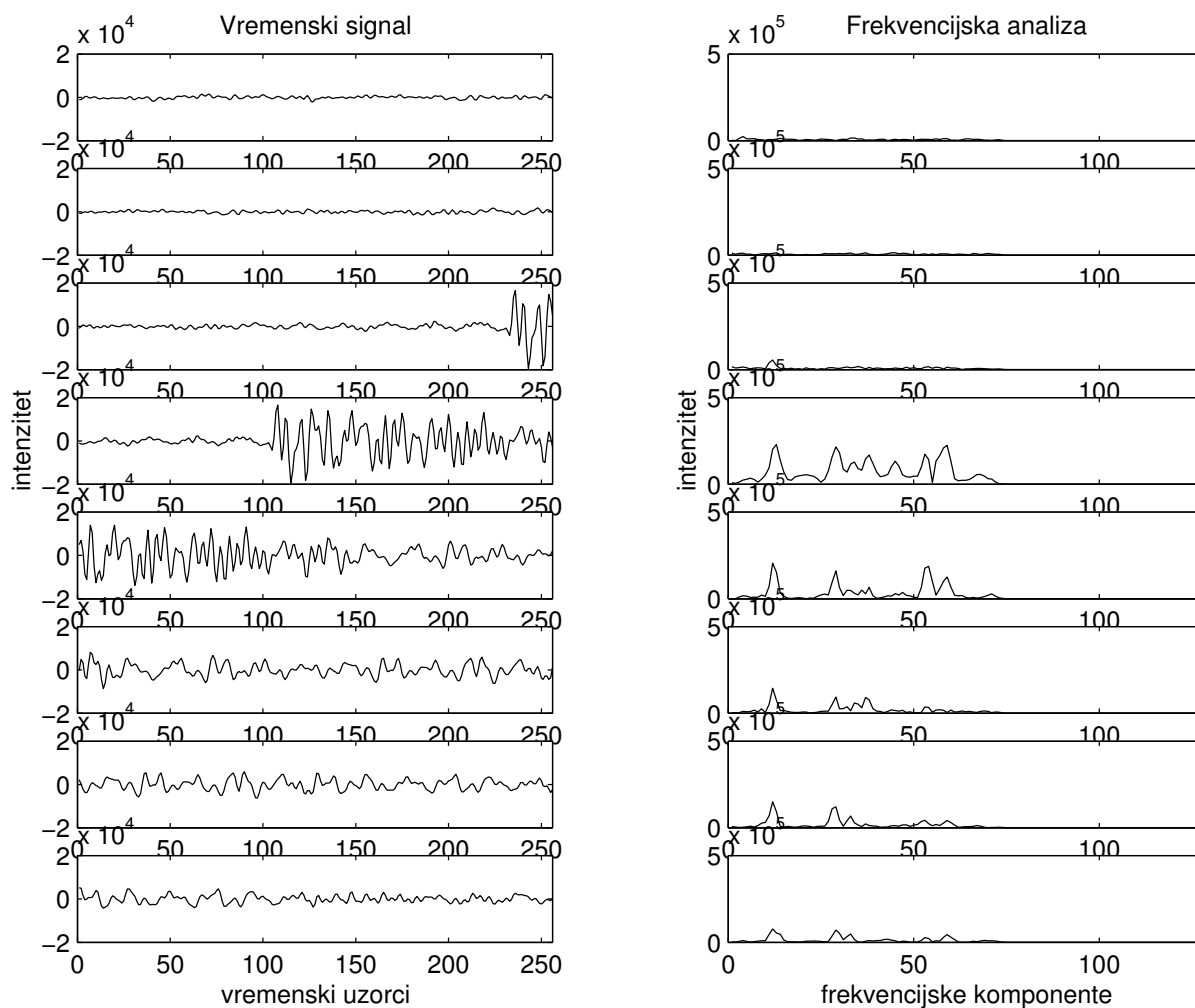
Slika 6.11: Vremenski signala uzet kao primjer za analizu grupiranja kratkih prozora.



Pogledaju li se *odoka* frekvencijski odzivi pojedinih dijelova signala (Slika 6.12), primjećuje se logično grupiranje prozora u najmanje dvije grupe – prva bi imala tri, a druga pet prozora. RAAC algoritam korektno proračunava ovaj slučaj grupiranja, i to kod energetskega praga k iznosa 3. Stroži slučaj bi bio prag 2 ili 1.5, dok su pragovi ispod 1.5 neefikasni u grupiranju. S druge strane, prag veći od 3 dovodi do prevelikog grupiranja.

Zaključujemo da se energije susjednih pojaseva moraju razlikovati za više od 1.5 i za manje od 3 puta da bi se započelo formiranje nove grupe prozora. U RAAC koderu ovaj prag je postavljen na 2.5.

Slika 6.12: Grupiranje 8 kratkih prozora unutar bloka. Frekvencijska analiza pokazuje logičko grupiranje u dvije grupe: prva tri i zadnjih pet blokova.



Tablica 6.4: Izabrano grupiranje prema Slici 6.12.

Prag k : odnos energija susjednih pojaseva	Proračunato grupiranje	
	Broj grupa	Raspored grupa
1,1	6	212111
1,3	5	32111
1,5	4	3212
2	3	332
3	2	35
5	1	8

6.4.11 *Bit-rezervoar i bitovi za popunjavanje*

Broj bita koji je na raspolaganju za kodiranje jednog bloka podataka zadan je sa željenim bitskim protokom i frekvencijom uzorkovanja ulaznog signala (Formula 4.6). Kako koder nije u mogućnosti precizno (*u bit*) ispoštovati zadani bitski protok za svaki blok, potrebno je voditi računa o ukupnom bitskom protoku. Za ispunjenje ovog zadatka upotrebljavaju se *bit-rezervoar* i *bitovi za popunjavanje* (engl. *fill-bits*).

U bit-rezervoar se sprema višak bitova koji je na raspolaganju trenutnom bloku, ukoliko neki od prošlih blokova nisu potrošili dozvoljeni broj bita. Na ovaj način se zahtjevniji dijelovi signala mogu kodirati sa više bita od prosječnog broja, jer se neki od percepcijski manje zahtjevnih blokova kodirao sa manje bita nego što je prosječno. Ukupni bitski protok će ostati očuvan.

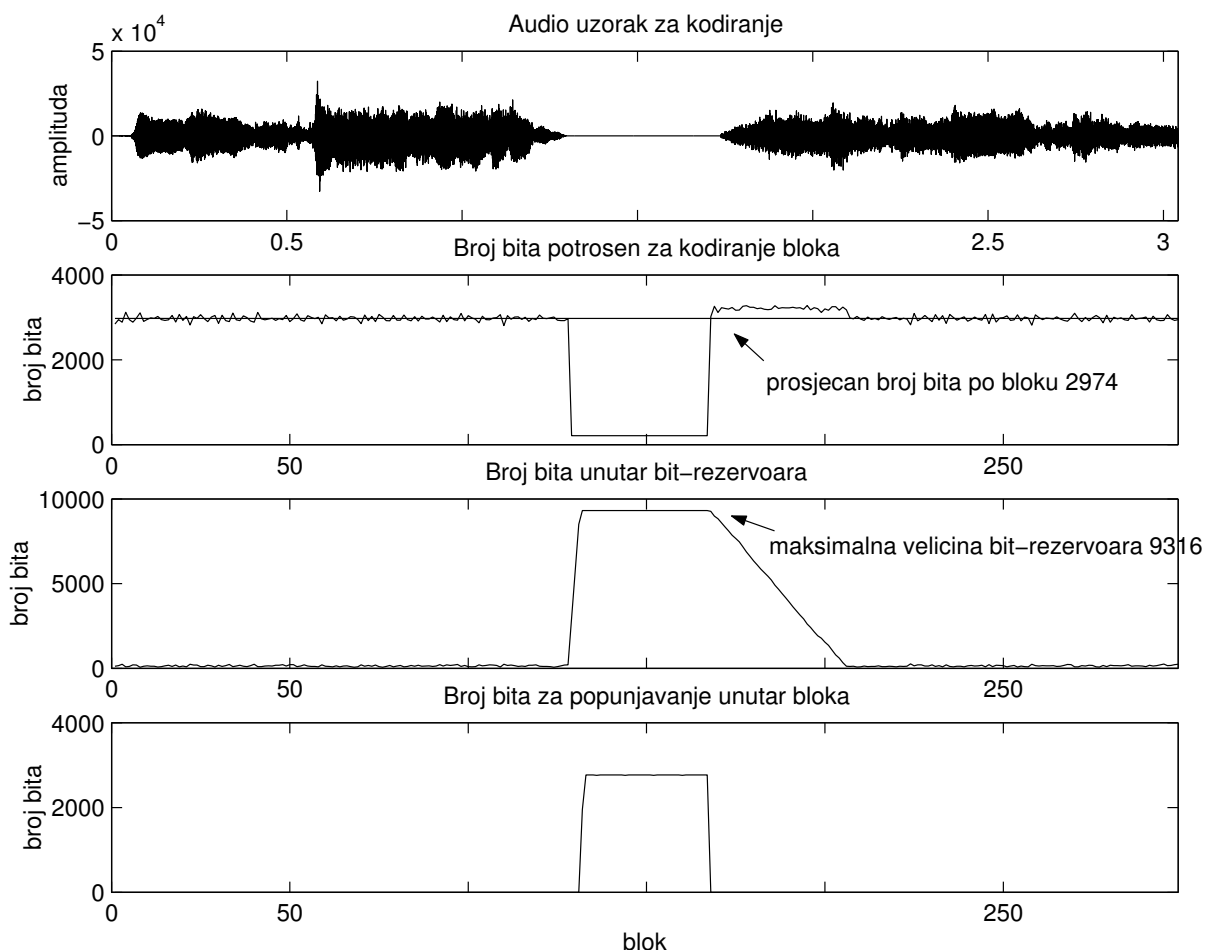
Maksimalni broj bita koji se može čuvati u bit-rezervoaru zadan je sa maksimalnom veličinom bloka za kodiranje koja je standardom zadana i iznosi 6144 bita po kanalu.

Npr. neka je željeni bitski protok 128000 b/s stereo, a frekvencija uzorkovanja 44.1 kHz: prosječna veličina bloka je 2972 bita. Maksimalan broj bita koji bit-rezervoar smije *posuditi* jednom bloku iznosi:

$$6144 \times 2 - 2972 = 9316 \quad (6.16)$$

Poseban slučaj se javlja kada bit-rezervoar postane pun. U ovom slučaju, da bi se održao zadani bitski protok, upotrebljavaju se tzv. *bitovi za popunjavanje*. Primjer je prikazan na Slici 6.13. Blokovi podataka se kodiraju sa većim ili manjim oscilacijama oko srednjeg bitskog protoka, sve dok se u signalu ne pojavi tišina. Blokovi tišine se kodiraju jednostavno, sa vrlo malim brojem bita (~200). Bit-rezervoar se vrlo brzo napuni, pa postaje nemoguće održavanje zadanog bitskog protoka. U ovom slučaju se upotrebljavaju bitovi za popunjavanje. U svaki blok se ubacuje onoliko bitova za popunjavanje koliko je potrebno da se postigne zadani bitski protok (što u sintaksi kodiranog toka podrazumijeva umetanje nekoliko *fill-elemenata*). Nakon što se opet pojavi zahtjevniji dio audio signala, bitovi za popunjavanje postaju suvišni, a preostali bitovi unutar bit-rezervoara se postepeno troše na kodiranje sukcesivnih audio blokova. Nakon nekog vremena bit-rezervoar je sveden na uobičajenu veličinu.

Slika 6.13: Potrošnja bita za audio uzorak sa tišinom. Nakon nastupanja tišine, blokovi podataka se kodiraju sa vrlo malom potrošnjom bita, pa je za održavanje zadanog bitskog protoka potrebno kodirani tok popuniti sa *bitovima za popunjavanje*. Nakon završetka tišine, sačuvan je pun bit-rezervoar koji se troši na kvalitetnije kodiranje nastupajućih blokova signala. Usprkos tome što su blokovi kodirani sa različitim brojem bita, zadani bitski protok je očuvan.



6.4.12 Kodirani tok

MPEG-4 GA kodirani tok je podijeljen u blokove označene kao *raw_data_block* elemente kodiranog toka. Svaki blok podataka sadrži potpunu i jednoznačnu informaciju o 1024 vremenska uzorka kodiranog audio signala. Jedan blok podataka RAAC koda može sadržavati sljedeće elemente toka:

- 1) zaglavlje,
- 2) `single_channel_element` (za mono signal),

- 3) `channel_pair_element` (za stereo signal),
- 4) `fill_element`.

Moguće je pojavljivanje jednog *single_channel_element* ili jednog *channel_pair_element* elementa, dok je broj *fill_element* elemenata neograničen.

6.4.12.1 Zaglavlje

U informativnom dijelu standarda su predložena dva tipa zaglavlja: ADTS i ADIF.

ADTS zaglavlje

ADTS (engl. *Audio Data Transport Stream*) je zaglavlje namijenjeno kodiranom toku koji se namjerava prenositi preko medija sklonih greškama (Internet, mobilne komunikacije). ADTS zaglavlje se sastoji od stalnog zaglavlja i promjenljivog zaglavlja i ponavlja se u svakom bloku podataka. Ukoliko dođe do prekida u toku, ponovna sinkronizacija je moguća od bilo kojeg bloka jer su sve informacije potrebne dekoderu sadržane u stalnom zaglavlju.

Upotrebom ADTS zaglavlja svaki blok kodiranog toka se opterećuje sa 58 bita koliko je zaglavlje veliko. Kod stereo signala frekvencije uzorkovanja 44.1 kHz koji se kodira sa 128000 b/s, jedan blok ima na raspolaganju 2974 bita, pa zaglavlje zauzima skoro 2% prostora.

ADIF zaglavlje

ADIF (engl. *Audio Data Interchange Format*) je zaglavlje namijenjeno kodiranom toku pohranjenom na fiksnim medijima. Ovo zaglavlje se upisuje samo jedanput na početku kodiranog toka.

Slika 6.14: Dva tipa zaglavlja predložena u informativnom dijelu MPEG-4 standarda. ADIF zaglavlje se upisuje samo jednom na početku kodiranog toka, dok se ADTS zaglavlje ponavlja prije svakog bloka, tako omogućavajući resinkronizaciju eventualno prekinutog toka.

ADIF Zaglavlje	Blok 1	Blok 2	Blok 3	Blok 4	
ADTS Zaglavlje	Blok 1	ADTS Zaglavlje	Blok 2	ADTS Zaglavlje	Blok 3

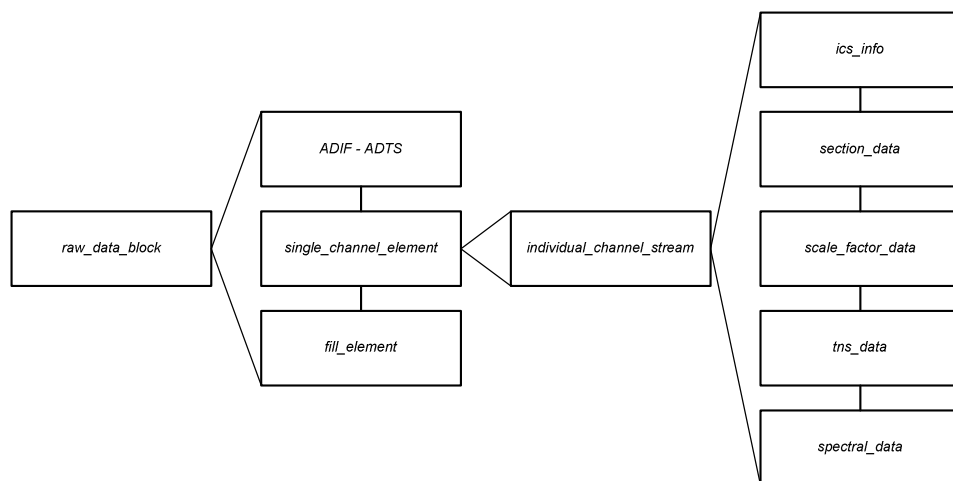
6.4.12.2 Single_channel_element

Single_channel_element se upotrebljava za kodiranje jednog mono audio kanala koji ne dijeli nikakve informacije sa drugim kanalima. Audio signal je predstavljen sintaksom definiranom sa *individual_channel_stream* elementom.

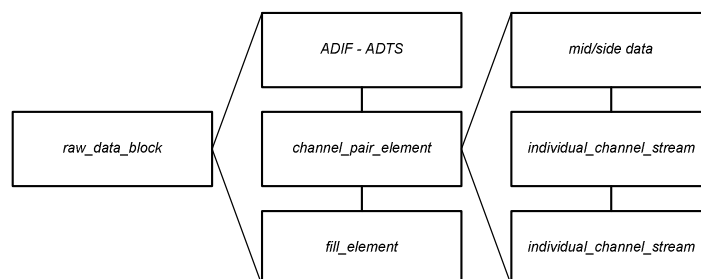
6.4.12.3 Channel_pair_element

Channel_pair_element se upotrebljava za kodiranje dva audio kanala. Dva kanala se kodiraju svaki sa posebnim *individual_channel_stream* elementom. Sintaksa omogućava kanalima dijeljenje zajedničkih informacija. Element se sastoji od informacije o upotrebi *zbroj/razlika* kodiranja između dva kanala (jedan bit za svaki pojas kvantizatora) i od dva zasebna *individual_channel_stream* elementa koji mogu, a ne moraju, dijeliti informacije.

Slika 6.15: Kodirani tok kod kodiranja jednokanalnog (mono) signala.



Slika 6.16: Kodirani tok kod kodiranja dvokanalnog (stereo) signala.



6.4.12.4 Individual_channel_stream

Individual_channel_stream element definira kodiranje jednog audio kanala. Jedan kanal je u potpunosti definiran sa ovim elementom ako se nalazi unutar *single_channel_element* elementa. Ukoliko se nalazi unutar *channel_pair_element* elementa, onda *individual_channel_stream* može dijeliti *ics_info* sa prvim *individual_channel_stream* elementom.

Individual_channel_stream se sastoji od sljedećih dijelova: *ics_info*, *section_data*, *scale_factor_data*, *tns_data* i *spectral_data*.

Ics_info

Ics_info je element u kojem su sadržane sljedeće informacije o tekućem bloku: dužina prozora, oblik prozora, broj kvantizatora i način grupiranja kratkih prozora.

Dva *individual_channel_stream* elementa koji se nalaze unutar *channel_pair_element* elementa, mogu dijeliti jedan *ics_info* element ukoliko im se podudaraju sve sadržane informacije.

Section_data

Section_data sadrži informacije o rasporedu Huffmanovih tablica po pojedinim pojasevima kvantizatora.

Tablica 6.5: Namjena Huffmanovih tablica. Tablice 13, 14 i 15 se upotrebljavaju za signaliziranje posebnog kodiranja PNS i IS algoritmima. Tablice 0 do 11 su standardne tablice za kodiranje kvantiziranih spektralnih vrijednosti (u parovima ili kvartetima).

Broj Huffmanove tablice	Objašnjenje
0 do 11	Tablice za kodiranje kvantiziranih spektralnih koeficijenata.
13	Tablica za kodiranje PNS vrijednosti.
14 i 15	Tablica za kodiranje IS vrijednosti.

Scale_factor_data

Scale_factor_data element sadrži Huffmanom kodirane kvantizatore (jedan kvantizator za svaki pojas kvantizatora). Unutar *scale_factor_data* elementa nalaze se izmiješane i PNS i IS vrijednosti koje su također Huffmanom kodirane, ali prema Tablici 6.5.

Tns_data

Tns_data element sadrži sve informacije vezane uz TNS filtriranje signala: broj filtara koji se upotrebljavaju unutar bloka, dužinu područja signala na koji se filter primjenjuje (u pojasevima kvantizatora), red filtra i same koeficijente filtra.

Spectral_data

Spectral_data element sadrži Huffmanom kodirane kvantizirane spektralne koeficijente.

6.4.12.5 Fill_element

Fill_element se upotrebljava za popunjavanje bloka percepcijski nebitnim bitovima, da bi se postigao zadani bitski protok u slučajevima kada to nije moguće postići regularnim kodiranjem.

6.5 Zaključak

U ovom poglavlju je opisana osnovna funkcionalnost RAAC audio koda. Prikazani su načini realizacije osnovnih dijelova koda. Iako je realizacija ovih dijelova usko vezana uz standard, RAAC ipak uvodi ponegdje značajne novine, a ponegdje manja poboljšanja. Pažnju treba obratiti na izvedbe:

- MDCT transformacije preko FFT transformacije,
- odabira dužine bloka i oblika prozora,
- grupiranja kratkih prozora,
- modula za kvantizaciju preko metode dvostruke petlje i
- upravljanja *bit-rezervom* i *bitovima za popunjavanje*.

U sljedećim poglavljima biti će opisane realizacije psihoakustičkog modula i modula za obradu spektra koji se obično nazivaju MPEG-4 *alati* (engl. *tools*).

Poglavlje 7

7 Psihoakustički model izveden u RAAC koderu

7.1 Uvod

Psihoakustički modul je jedan od najkritičnijih dijelova svakog percepcijskog audio koda. O psihoakustičkom modelu ovisi do koje razine će psihoakustički principi biti eksploatirani unutar koda i koliko će koder uspjeti iskoristiti spomenute principe pri postizanju svog cilja – što veće kompresije originalnog audio signala, bez narušavanja kvalitete njegovih percepcijski bitnih dijelova.

Kao što je već ranije rečeno, psihoakustički modul nije standardiziran. Unutar MPEG-4 GA standarda, *informativni* dio predlaže određenu izvedbu psihoakustičkog modula, međutim spomenuta izvedba nije obavezna: na dizajneru koda je da odluči kakav psihoakustički model će primijeniti. Na kraju krajeva, bilo bi besmisleno standardizirati sve dijelove koda jer bi u tom slučaju postojala jedna jedina mogućnost izvedbe, pa bi svi koderi bili jednaki. Ovako, svaki koder koristi svoj način analize psihoakustičkih svojstava signala i svaki se razlikuje po dobivenim rezultatima. Na publici je da ocijeni valjanost svake pojedinačne izvedbe.

7.2 Cilj psihoakustičke analize

Glavni cilj psihoakustičke analize je proračun minimalnog *praga maskiranja* unutar svakog frekvencijskog (kritičnog) pojasa ulaznog signala. Na osnovu ulaznog vremenskog signala, psihoakustički modul proračunava prag maskiranja – prag maskiranja se šalje kvantizacijskom modulu čiji je zadatak kvantizirati signal, na način da nakon dekvantizacije

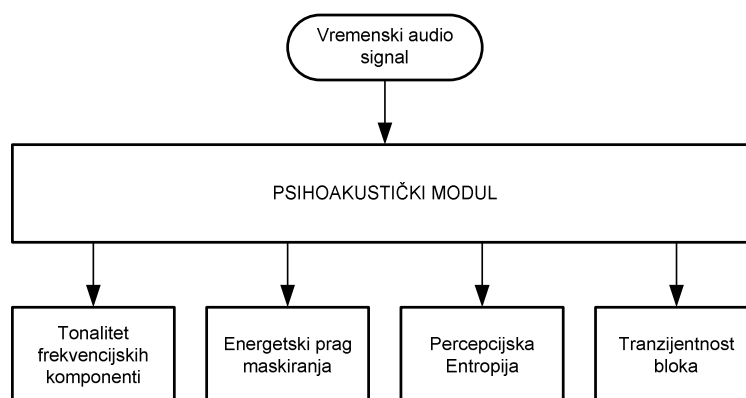
šum unesen u signal bude ispod praga maskiranja. Na taj način šum unesen u signal kvantizacijom ostaje nečujan, a signal ostaje percepcijski jednak originalu, iako on to objektivno nije.

Psihoakustički modul realiziran u RAAC koderu osim osnovnog proračuna minimalnog praga maskiranja na kojem se zasniva cijeli proces kvantizacije, daje i druge informacije koje su presudne za eksploatiranje pojedinih algoritama MPEG-4 GA standarda. Tako tonalitet frekvencijskih komponenti igra odlučujuću ulogu kod primjene algoritma za percepcijsku zamjenu šuma (PNS), a upotrebljava se i kod *intenzitet-stereo* (IS) algoritma. Također, i rezultat proračuna percepcijske entropije signala se upotrebljava unutar PNS algoritma.

Ukupno uzevši, RAAC psihoakustički modul daje sljedeće informacije kao rezultat psihoakustičke analize:

- 1) minimalni prag maskiranja po kritičnim pojasevima,
- 2) tonalitet svake frekvencijske komponente,
- 3) tranzijentnost cijelog bloka,
- 4) percepcijsku entropiju cijelog bloka.

Slika 7.1: Psihoakustički modul je presudan za kvalitetu kodiranog signala: osim energetskog praga maskiranja koji se koristi kod kvantizacije, psihoakustički modul daje informacije i o drugim svojstvima signala kao što su stacionarnost/tranzijentnost i tonalitet. Ove informacije igraju veliku ulogu pri eksploataciji određenih MPEG-4 GA algoritama.



7.3 Psihoakustički model 2

RAAC koder koristi izmijenjen psihoakustički model predložen unutar MPEG-4 informativnog dijela standarda. Spomenuti psihoakustički model je poznat pod imenom *Psihoakustički model 2*, a uveden je u informativnom dijelu MPEG-1 audio standarda. Izmijenjeni su gotovo svi značajni dijelovi predloženog modela, dok je zadržana samo glavna koncepcija proračuna energetskega praga maskiranja. Najvažnije izmjene se odnose na izbor funkcije *širenja maskiranja*, proračun tonaliteta signala i proračun tranzijentnosti signala, dakle na bitnije dijelove modula.

7.4 Proces psihoakustičke analize

Psihoakustička analiza počinje formiranjem ulaznog bloka podataka na kojem će se analiza obaviti. Ulaz u blok za psihoakustičku analizu je niz od 2048 uzoraka vremenskog signala – to je isti ulazni signal koji se dovodi i na ulaz standardiziranog bloka za vremensko-frekvencijsku transformaciju signala, na ulaz MDCT filtarskog sloga. Zbog poznatih uvjeta koje filtarski slog mora zadovoljavati, svaki ulazni blok za analizu se polovično preklapa sa prethodnim blokom – on se dakle sastoji od pola uzoraka iz prethodnog bloka i pola novih uzoraka. Osim trenutnog bloka, modul mora imati na raspolaganju i zadnja dva bloka signala zbog algoritma za proračun tonalnosti signala koji se zasniva na predviđanju amplitude i faze signala preko niza ulaznih blokova.

Pošto psihoakustički modul eksploatira psihoakustička svojstva signala, a koja je najjednostavnije predočiti u frekvencijskoj domeni, prvi korak u psihoakustičkoj analizi je prebacivanje signala iz vremenske u frekvencijsku domenu. Kako još jedan sastavni dio koda (filtarski slog) radi isto, dakle vrši vremensko-frekvencijsku transformaciju signala, postavlja se pitanje zašto psihoakustički modul mora imati posebnu vremensko-frekvencijsku transformaciju?

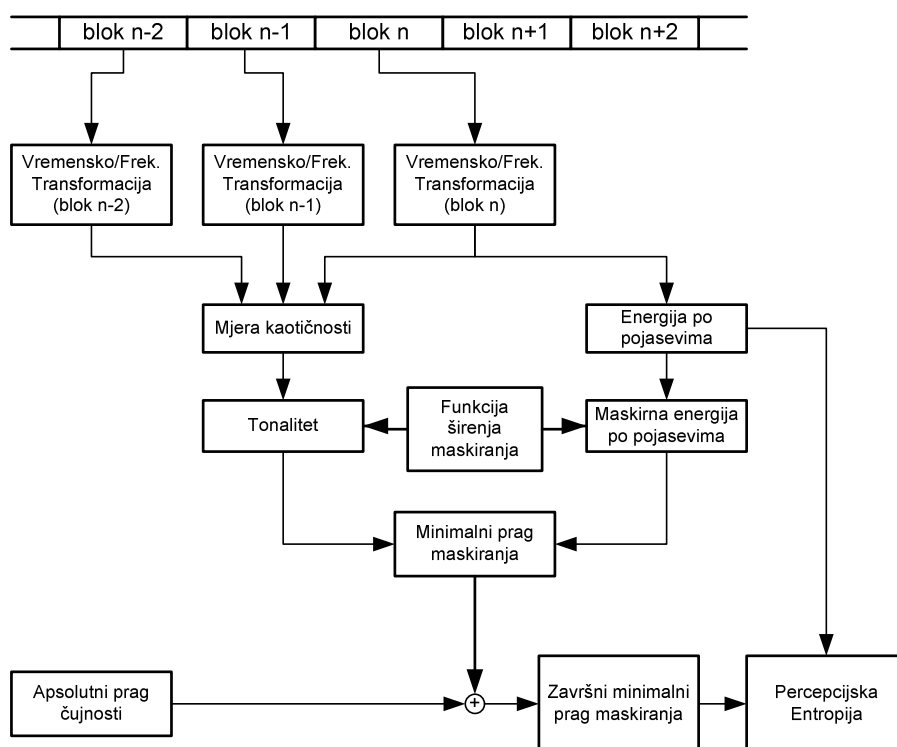
Filtarski slog koristi modificiranu diskretnu kosinusnu transformaciju (MDCT) zbog njenih svojstava idealne rekonstrukcije, poništavanja aliasinga, itd. Ova svojstva međutim, nisu bitna za psihoakustičku analizu. Psihoakustički modul zahtijeva što bolju frekvencijsku rezoluciju signala, informaciju o intenzitetu i fazi svake komponente, a to MDCT transformacija ne donosi. Zbog ovog razloga se u većini slučajeva, unutar psihoakustičkog modula vrši posebna vremensko-frekvencijska transformacija. Treba napomenuti da postoje izvedbe koda koje i unutar psihoakustičkog modula koriste (doduše proširenu) MDCT transformaciju, međutim kvaliteta spomenutih koda ostaje upitna.

RAAC koder u psihoakustičkom modulu koristi diskretnu Fourierovu transformaciju (DFT). DFT je vrlo pogodna za psihoakustičku analizu zbog sljedećih svojstava:

- 1) ima vrlo dobru frekvencijsku rezoluciju,
- 2) donosi informaciju o fazi signala,
- 3) postoje brzi algoritmi za proračun (FFT).

Nakon DFT transformacije, dobivene frekvencijske komponente se grupiraju u frekvencijske pojaseve koji aproksimativno odgovaraju kritičnim pojasevima ljudskog slušnog sistema. Obzirom na rezoluciju transformacije, nisko-frekvencijski kritični pojasevi će biti predstavljeni sa samo jednom ili nekoliko frekvencijskih komponenti, dok će visoko-frekvencijski pojasevi sadržavati na desetine frekvencijskih komponenti.

Slika 7.2: Blok shema proračuna unutar psihoakustičkog modela.



Nakon grupiranja frekvencijskih komponenti pristupa se proračunu energetskog praga maskiranja unutar svakog kritičnog pojasa. Za proračun energetskog praga maskiranja potrebno je prvo proračunati tonalitet (preko *kaotičnosti*), zatim primjenom funkcije *širenja maskiranja* izračunati i efektivno maskiranje u svakom pojasu. Proces psihoakustičke analize

(Slika 7.2) završava proračunom minimalnog praga maskiranja unutar svakog pojasa, kombinacijom apsolutnog praga maskiranja sa proračunatim.

Prikažimo opisani proces psihoakustičke analize na konkretnom primjeru, korak po korak.

7.5 Tok psihoakustičke analize

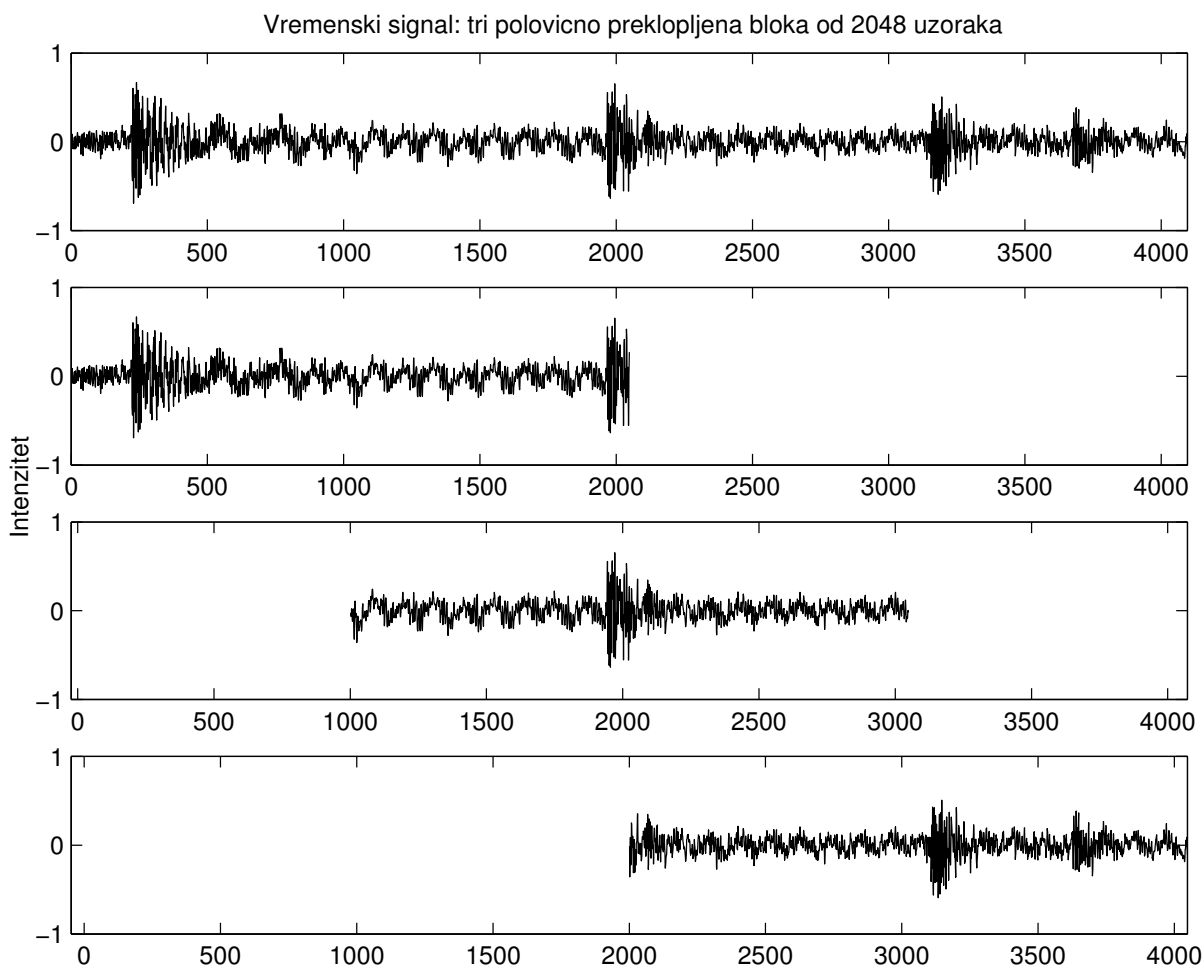
Primjer psihoakustičke analize je dan na audio signalu frekvencije uzorkovanja od 44.1 kHz.

7.5.1 Ulazni vremenski signal

Ulaz u psihoakustički modul je vremenski signal. Zbog polovičnog preklapanja, svaki blok se sastoji od 1024 uzorka iz prošlog bloka i 1024 nova uzorka. Ukupna dužina bloka zadana standardom iznosi 2048 uzoraka. Na Slici 7.3 je prikazan ulazni signal i njegova podjela na niz od tri polovično preklopljena bloka.

U ovom primjeru biti će izvršena psihoakustička analiza zadnjeg od tri bloka, dok su prva dva neophodna zbog metode predviđanja kojom se određuje tonalitet signala.

Slika 7.3: Vremenski signal se dijeli na blokove od 2048 uzoraka. Svaki blok se sastoji od 1024 uzorka starog boka i 1024 nova uzorka. Slika prikazuje tri uzastopna bloka koji se jedan za drugim šalju u psihoakustički modul. Analiza u ovom primjeru se odnosi na zadnji blok (uzorci 2048 do 4096).

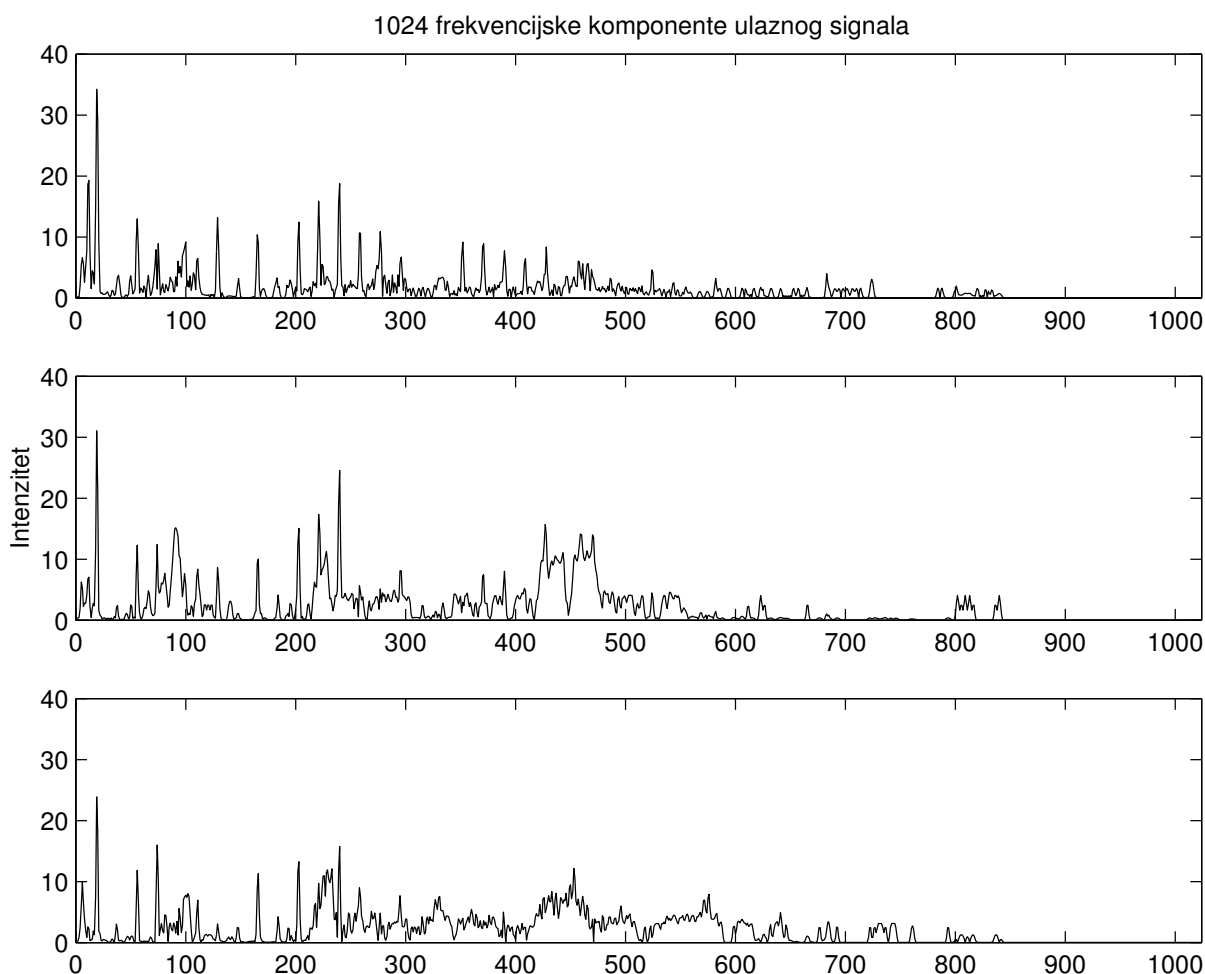


7.5.2 Frekvencijska domena

Nakon formiranja ulaznog bloka podataka, prije DFT transformacije, vrši se množenje (*prozorovanje*) cijelog bloka sa koeficijentima Hannovog prozora. Zatim se vrši diskretna Fourierova transformacija koja kao rezultat daje informaciju o amplitudi i fazi 1024 frekvencijske komponente signala.

Ovim postupkom, signal je prebačen u frekvencijsku domenu (Slika 7.4), u kojoj se cijeli proces vodi do samog kraja.

Slika 7.4: Nakon diskretne Fourierove transformacije, tri uzastopna bloka su predstavljena u frekvencijskoj domeni, u kojoj se vrše sve daljnje analize.



Napomenimo da se u jednom koraku ne rade DFT transformacije nad tri bloka, nego samo nad tekućim. Rezultati DFT transformacije za zadnja dva bloka su sačuvani za vrijeme njihove vlastite analize.

7.5.3 Energija po maskirnim pojasevima

Pošto je cilj psihoakustičke analize dobivanje po jednog minimalnog praga maskiranja za svaki kritični pojas, pristupa se preslikavanju energija frekvencijskih komponenti u energije maskirnim pojasevima. Pošto se širina maskirnih pojaseva mijenja od pojasa do pojasa, cijeli proces je izrazito nelinearan. Zato je na prvi pogled teško vizualno povezati početnu frekvencijsku analizu sa energijama maskirnih pojaseva.

Standardom su za svaku frekvenciju uzorkovanja ulaznog signala zadane posebne tablice sa ukupnim brojem i širinama pojaseva. Podržan je širok spektar frekvencija uzorkovanja

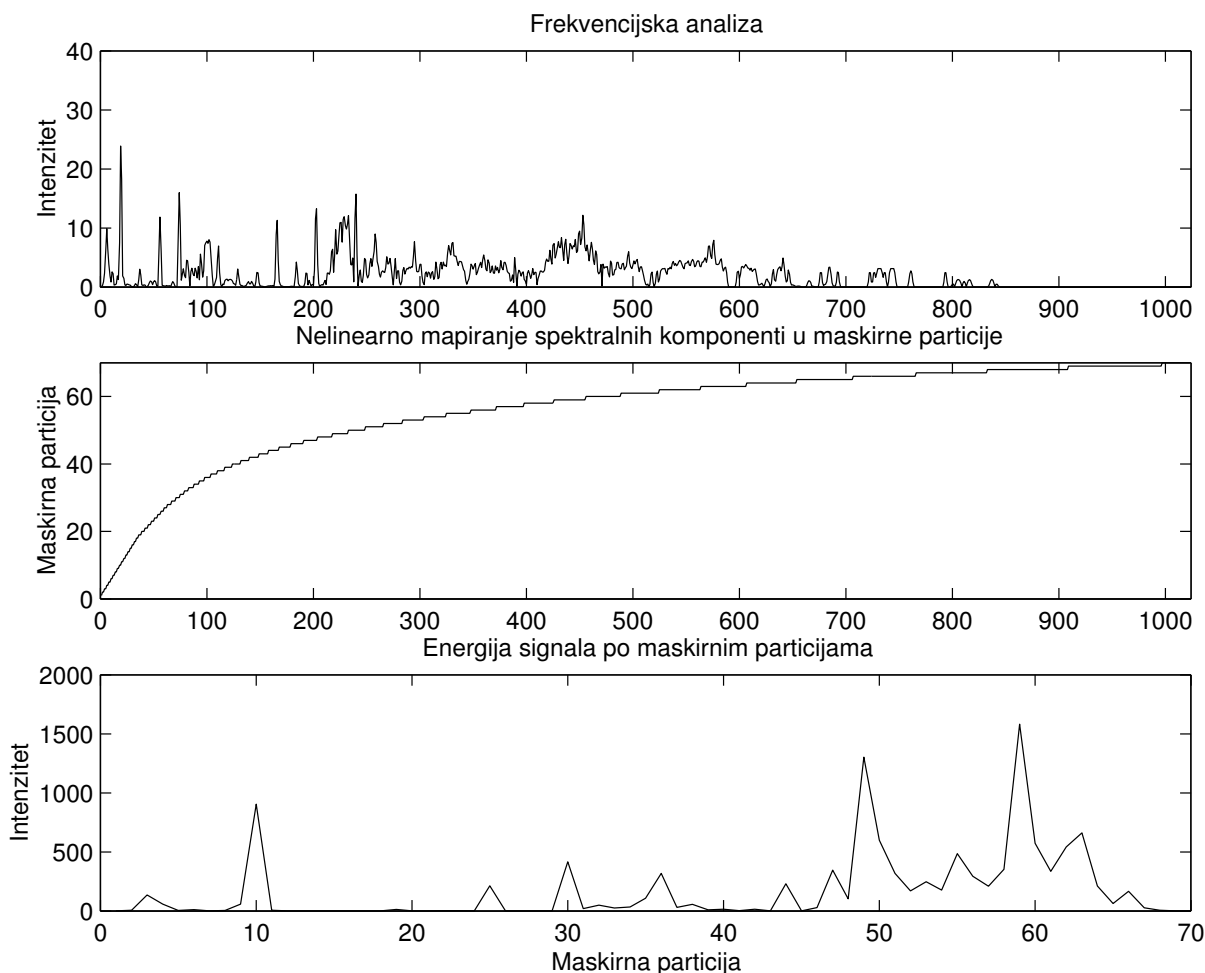
ulaznog signala: 8, 11.025, 12, 16, 22.05, 24, 32, 44.1, 48, 64, 88.2 i 96 kHz. Treba napomenuti da spomenute širine pojaseva samo aproksimiraju širine kritičnih pojaseva ljudskog slušnog sistema, i da je njihov broj ograničen tako da broj faktora kvantizacije koji je jednak broju pojaseva ne bude prevelik da ne bi suviše opterećivao kodirani tok.

Unutar psihoakustičkog modela moguće je korištenje točnije podjele frekvencijskog spektra na maskirne pojaseve od one definirane standardom. Točnija podjela bi bolje odgovarala kritičnim pojasevima ljudskog slušnog sistema. RAAC upravo to i radi: psihoakustički modul koristi precizniju podjelu spektra na maskirne pojaseve koja je definirana u informativnom dijelu standarda, i unutar njih proračunava minimalne pragove maskiranja. Na kraju procesa analize dobiveni pragovi maskiranja se preslikavaju u pojaseve koje zahtijeva standard.

Logično je da je broj pojaseva veći unutar psihoakustičkog modula od onih definiranih standardom. Npr. za frekvenciju uzorkovanja od 44.1 kHz standard dijeli spektar od 22.05 kHz na 49 pojaseva, dok se unutar psihoakustičkog modela spektar dijeli na 70 maskirnih pojaseva koji vjernije odražavaju širine stvarnih kritičnih pojaseva ljudskog slušnog sistema.

Rezultat procesa preslikavanja frekvencijskih komponenti u maskirne pojaseve je prikazan na Slici 7.5. Maskirni pojasevi se još nazivaju i *kritični* pojasevi zato što se odnose na *kritične pojaseve* ljudskog slušnog sistema.

Slika 7.5: Preslikavanje frekvencijskih komponenti u maskirne pojaseve. Širina maskirnih pojaseva je najmanja na niskim frekvencijama, gdje po samo nekoliko ili čak samo jedna frekvencijska komponenta čini energiju cijelog pojasa, dok na visokim frekvencijama u energiji pojasa učestvuje i preko 50 frekvencijskih komponenti. Za ovakvu raspodjelu je *kriv* ljudski slušni sistem koji ima vrlo malu selektivnost na visokim frekvencijama.



7.5.4 Maskirna energija i funkcija širenja maskiranja

Nakon preslikavanja energije u maskirne pojaseve, vrši se proračun ukupne *maskirne* energije u svakom pojasu. Ovaj proces je neophodan da bi se u proračun uveo utjecaj jednog pojasa na maskiranje unutar susjednih pojaseva. Cijeli proces tzv. *širenja maskiranja* se bazira na vrlo bitnoj funkciji *širenja maskiranja*. Funkcija širenja maskiranja odražava utjecaj jednog pojasa na drugi, a trebala bi što vjernije aproksimirati maskiranje predstavljeno Slikom 3.3.

RAAC koder upotrebljava funkciju maskiranja predstavljenu sljedećom formulom:

$$s_{dB} = \begin{cases} 27 \times (i - l) \\ \left[-24 - \frac{230}{f_c(l)} + 0.2 \times 10 \log_{10}(E) \right] \times (i - l) \end{cases} \quad (7.1)$$

gdje je

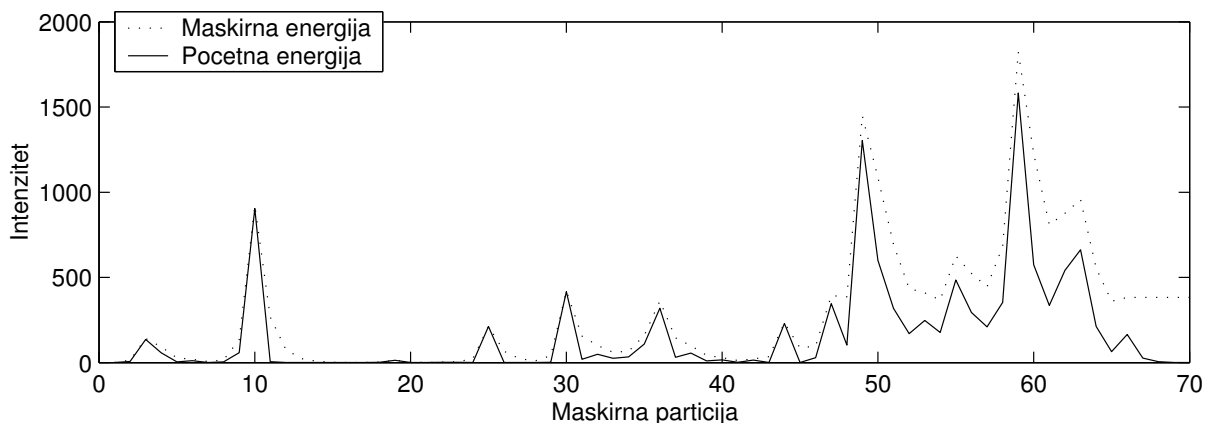
- i Bark vrijednost pojasa koji maskira,
- l Bark vrijednost pojasa koji biva maskiran,
- E energija pojasa, i
- f_c centralna frekvencija pojasa i .

Prvi član funkcije predstavlja maskiranje unatrag, koje je konstantno i , prema mjerenjima, iznosi približno 27 dB/Bark.

Drugi član funkcije predstavlja maskiranje unaprijed, a sastoji se od tri pod-člana. Prvi pod-član je konstantan i predstavlja minimalno maskiranje unaprijed od 24 dB/Bark (preporuča se dodatno eksperimentiranje sa ovim članom jer, prema novijim istraživanjima, puno jače maskiranje od 10 dB/Bark daje sasvim dobre rezultate). Drugi pod-član odražava utjecaj centralne frekvencije na maskiranje na način da je maskiranje pojačano na jako niskim frekvencijama. Treći pod-član u funkciji predstavlja poznati efekt pojačanja maskiranja unaprijed sa povećanjem energije maskera.

Nakon primjene funkcije širenja maskiranja dobivena je krivulja maskirne energije po pojasevima prikazana na Slici 7.6.

Slika 7.6: Nakon primjene funkcije širenja maskiranja, energija po kritičnim pojasevima se *razlila*. Dobivena je efektivna energija koja utječe na maskiranje svakog pojasa.



Nakon određivanja maskirne energije po pojasevima, za određivanje praga maskiranja potrebno je još odrediti tonalitet pojedinih particija, i to zbog efekta nejednakog maskiranja tonalnih i netonalnih komponenti signala.

7.5.5 Proračun tonaliteta

RAAC koder računa tonalitet metodom predviđanja (Poglavlje 4.2.2.3.2). Zadane su sljedeće vrijednosti:

$r_k(t)$ – intenzitet k -te komponente signala u bloku t ,

$f_k(t)$ – faza k -te komponente signala u bloku t .

Moguće su dvije pretpostavke obzirom na izgled amplitude i faze čistog tonalnog signala.

Prva pretpostavka je da čist tonalni signal ima linearnu fazu i linearnu amplitudu:

$$r_{k_pred}(t) = r_k(t-1) + (r_k(t-1) - r_k(t-2)) \quad (7.2)$$

$$f_{k_pred}(t) = f_k(t-1) + (f_k(t-1) - f_k(t-2)) \quad (7.3)$$

Druga pretpostavka je da čist tonalni signal ima linearnu fazu i konstantnu amplitudu:

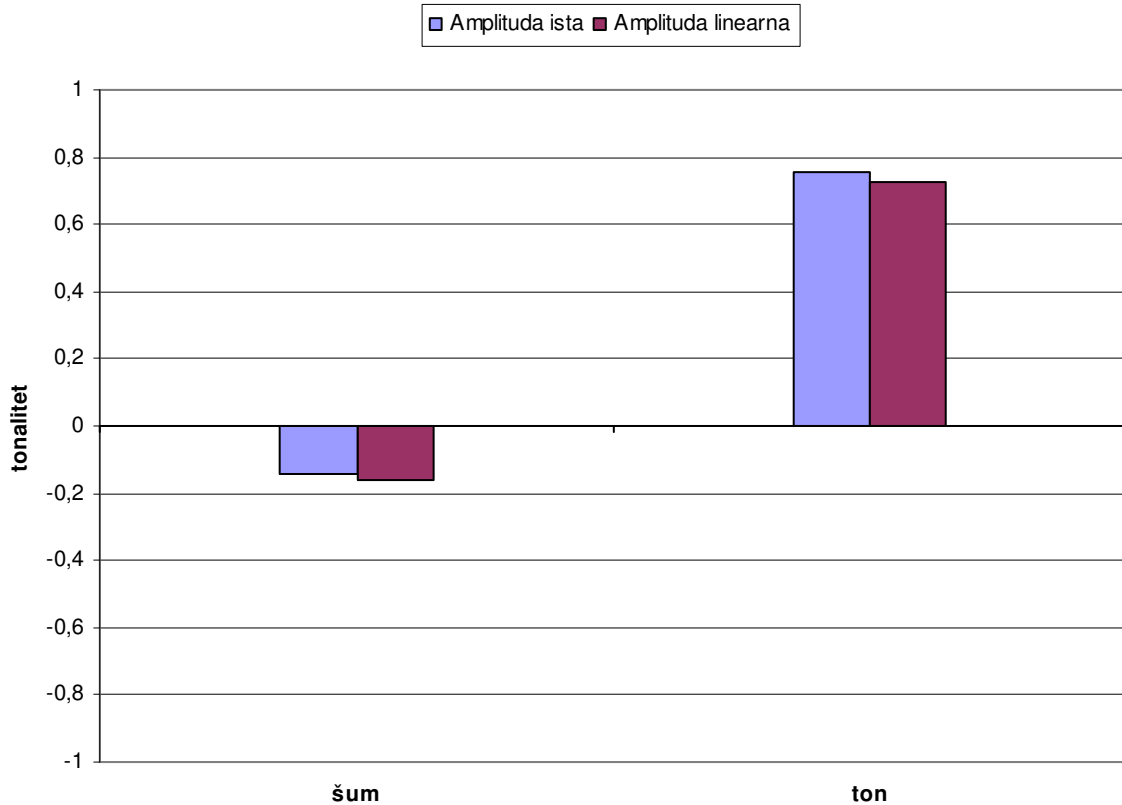
$$r_{k_pred}(t) = r_k(t-1) \quad (7.4)$$

RAAC algoritam polazi od pretpostavke da stacionarni tonalni signali imaju konstantnu amplitudu koja se ne mijenja s vremenom i fazu koja je linearna.

Za efikasno predviđanje, algoritmu moraju biti na raspolaganju informacije o intenzitetu i fazi signala u tekućem bloku i u dva prethodna bloka. Pomoću tih informacija računaju se predviđene vrijednosti amplitude i faze, nakon čega se usporede sa stvarnim vrijednostima.

Primjer usporedbe dva algoritma predviđanja dan je na Slici 7.7. Očito je da predviđanje konstantne amplitude daje bolji rezultat nego predviđanje linearne amplitude pa se ova metoda upotrebljava u RAAC koderu.

Slika 7.7: Usporedba proračuna tonalnosti na dva načina za dva različita signala, čisti ton i bijeli šum. RAAC koder koristi algoritam koji predviđa konstantnu amplitudu i linearnu fazu jer daje veći tonaliteta nego algoritam koji predviđa linearnu amplitudu i linearnu fazu.



Nakon izračuna predviđenih vrijednosti, računa se tzv. *mjera kaotičnosti* ili *mjera nepredvidivosti* preko euklidske udaljenosti između predviđene i stvarne vrijednosti:

$$mk_k(t) = \frac{Eukl.Udalj. \left[(r_{k_pred}(t), f_{k_pred}(t)), (r_k(t), f_k(t)) \right]}{r_k(t) + abs(r_{k_pred}(t))} \quad (7.5)$$

gdje je

$$Eukl.Udalj.(x(\varphi), y(\varphi)) = \sqrt{(x \cos \varphi - y \cos \varphi)^2 + (x \sin \varphi - y \sin \varphi)^2} \quad (7.6)$$

Mjera kaotičnosti se jednostavno preslikava u interval od -1 do 1 proračunom *mjere tonaliteta* formulom:

$$mt_k(t) = -0.299 - 0.43 * \log_{10} mk_k(t) \quad (7.7)$$

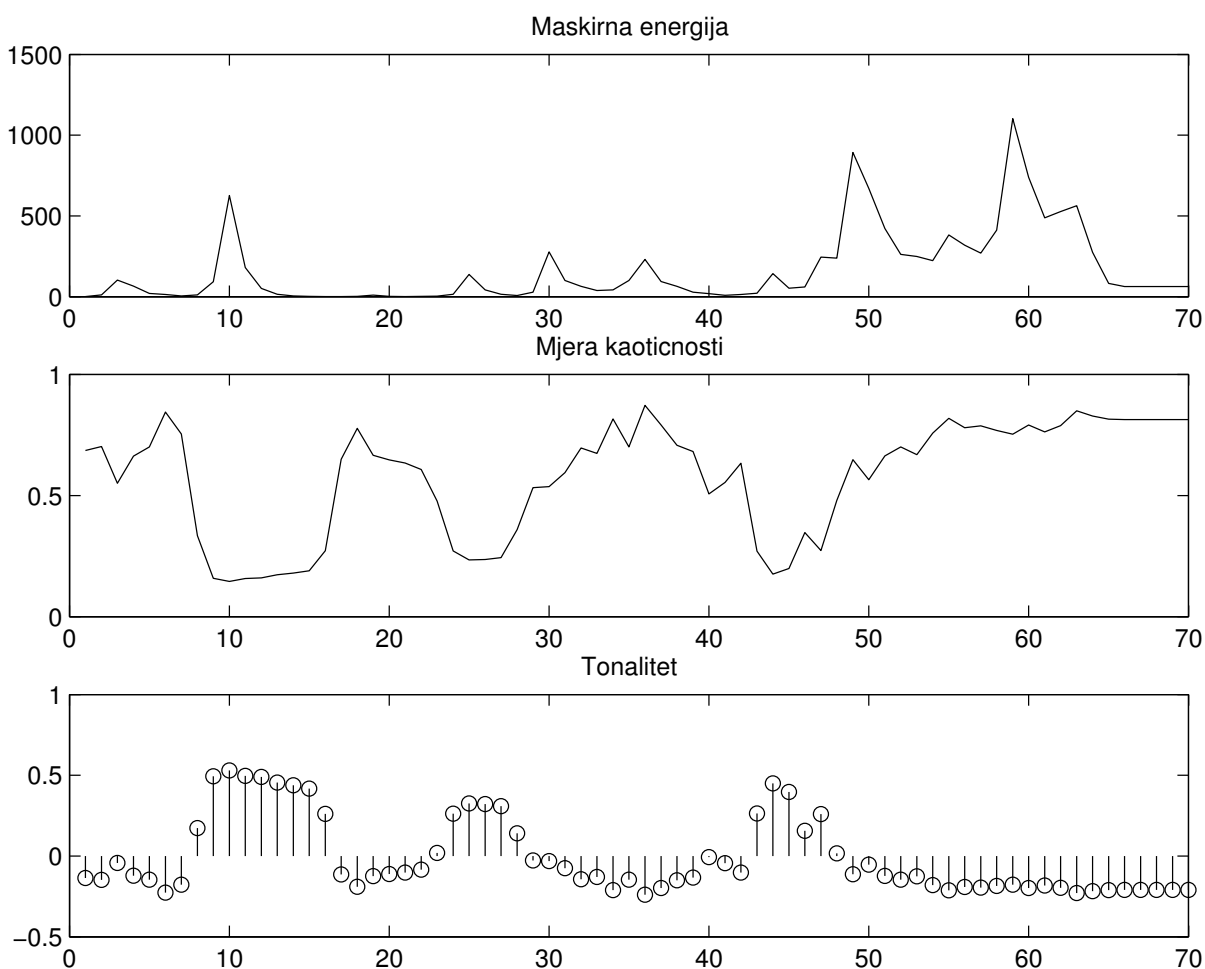
Tonalne komponente imaju vrijednost bližu 1, dok netonalne komponente imaju vrijednosti bliže nuli i negativne.

Proračunati tonalitet za vremenski signal sa Slike 7.3 prikazan je na Slici 7.8.

7.5.6 Omjer *signal/šum*

Razlog zbog kojeg se računa tonalitet signala je različitost svojstava maskiranja koja imaju tonalni i netonalni signali. Kao što je objašnjeno u Poglavlju 3.4, netonalni signali jače maskiraju od tonalnih signala (Slika 3.6). Informacija o tonalnosti signala se koristi kod proračuna zahtijevanog *signal/šum* omjera.

Slika 7.8: Tonalitet signala se računa iz *mjere kaotičnosti*, a mjera kaotičnosti metodom predviđanja: što predvidljiviji signal to manje kaotičan. Sasvim tonalan signal teži vrijednosti 1 dok netonalni signali daju vrijednosti manje od 0.



RAAC uzima sljedeće vrijednosti maskiranja za tonalne i netonalne (šum) signale:

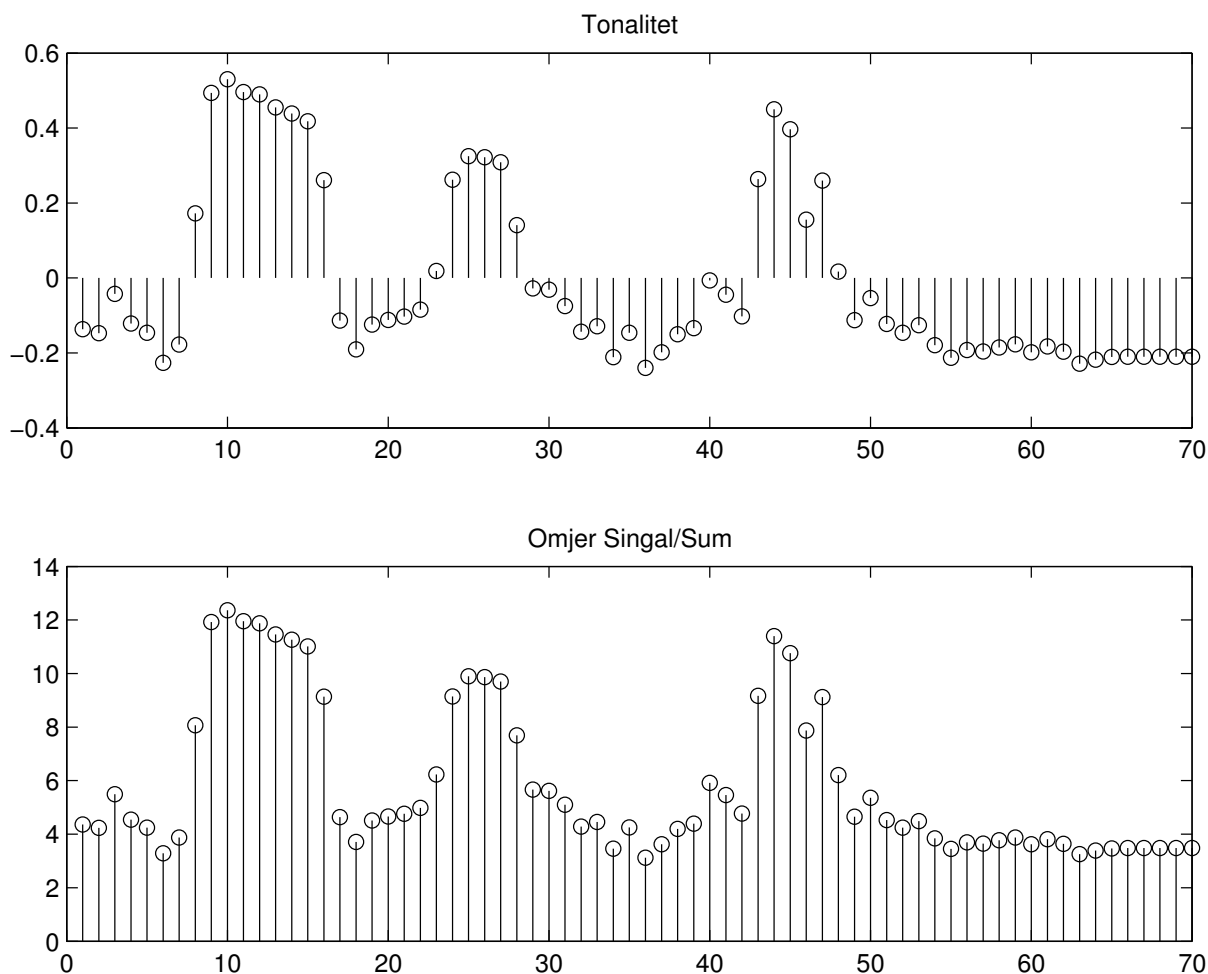
$$\begin{aligned} TonMask\check{S}um &= 18[dB] \\ \check{S}umMaskTon &= 6[dB] \end{aligned} \quad (7.8)$$

Pomoću ovih vrijednosti računa se zahtijevani omjer *signal/šum*:

$$Signal\check{S}um = mt \times TonMask\check{S}um + (1 - mt) \times \check{S}umMaskTon \quad (7.9)$$

gdje je *mt* mjera tonalnosti.

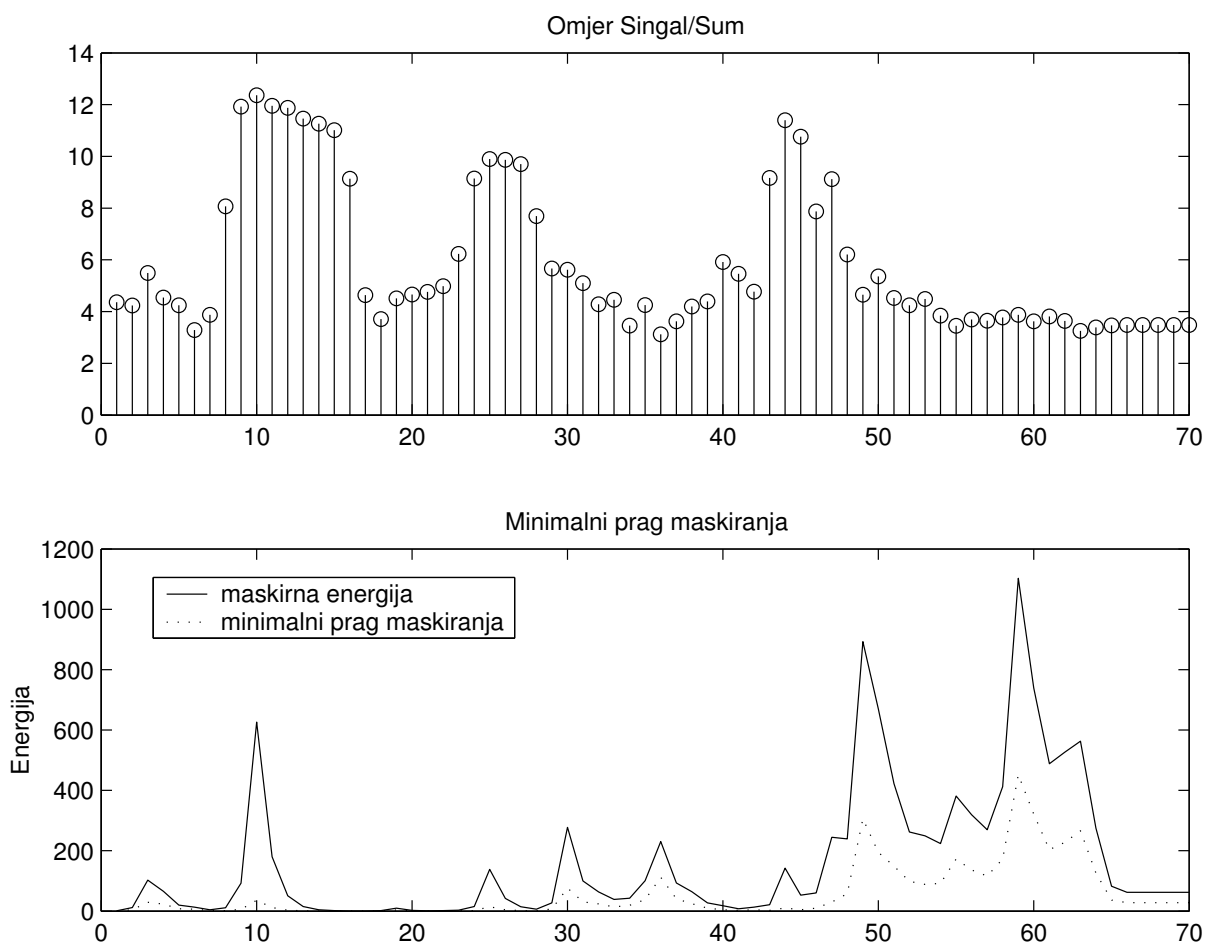
Slika 7.9: Na osnovu tonaliteta, proračunava se omjer *signal/šum*. Šumne komponente signala jače maskiraju nego tonalne pa je dozvoljeni omjer *signal/šum* veći u tonalnim dijelovima signala.



7.5.7 Energetski prag maskiranja

Nakon što je izračunat omjer *signal/šum* i maskirna energija po pojasevima, može se pristupiti proračunu minimalnog praga maskiranja. Potrebno je u svakom pojasu ove dvije vrijednosti podijeliti prema Formuli 7.10. Dobiveni prag maskiranja je prikazan na Slici 7.10.

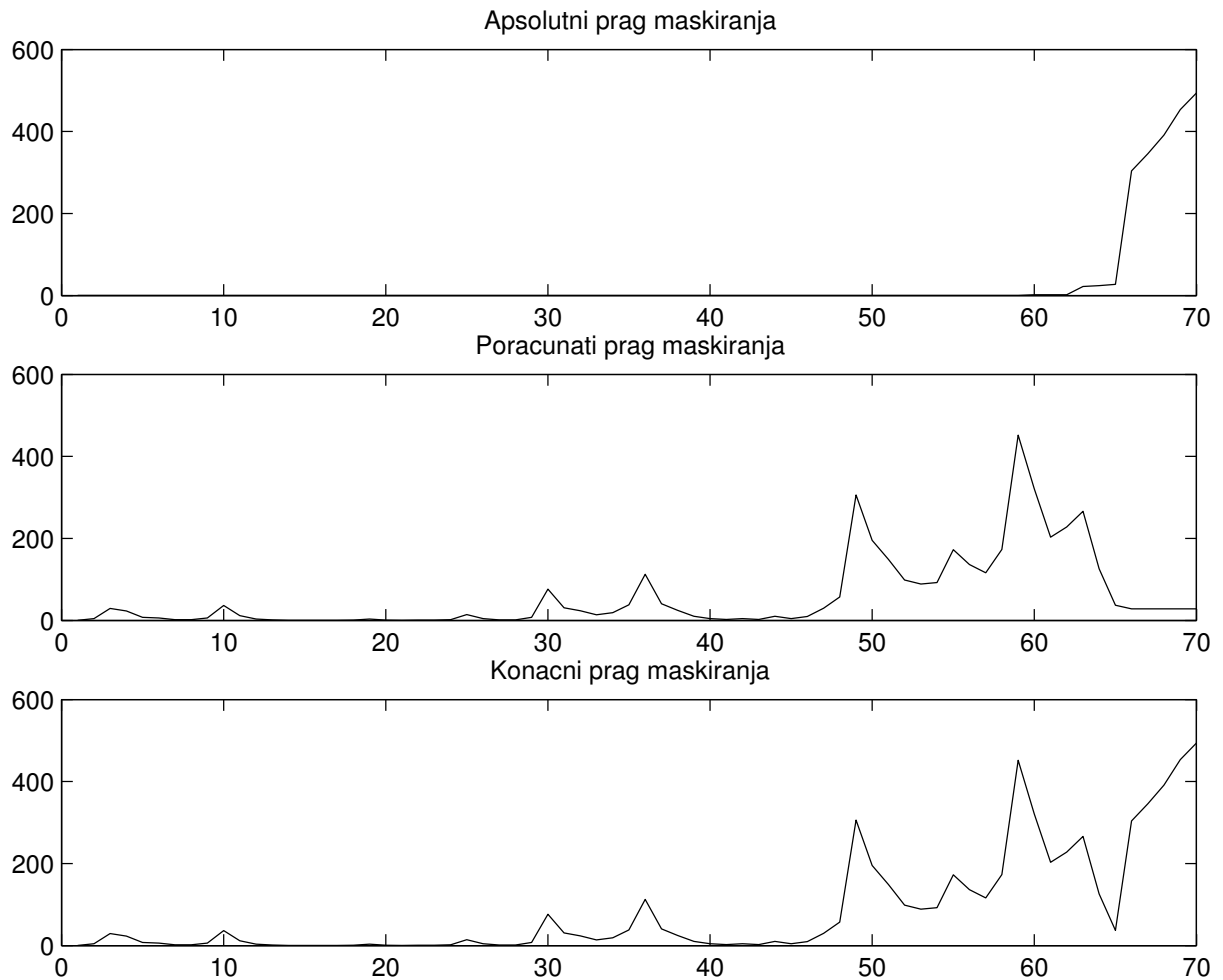
$$\frac{\text{Signal}}{\text{Šum}} = \frac{\text{Energija Signala}}{\text{Prag Maskiranja}} \Rightarrow \text{Prag Maskiranja} = \frac{\text{Energija Signala}}{\frac{\text{Signal}}{\text{Šum}}} \quad (7.10)$$

Slika 7.10: Proračun maskirne energije prema omjeru *signal/šum*.

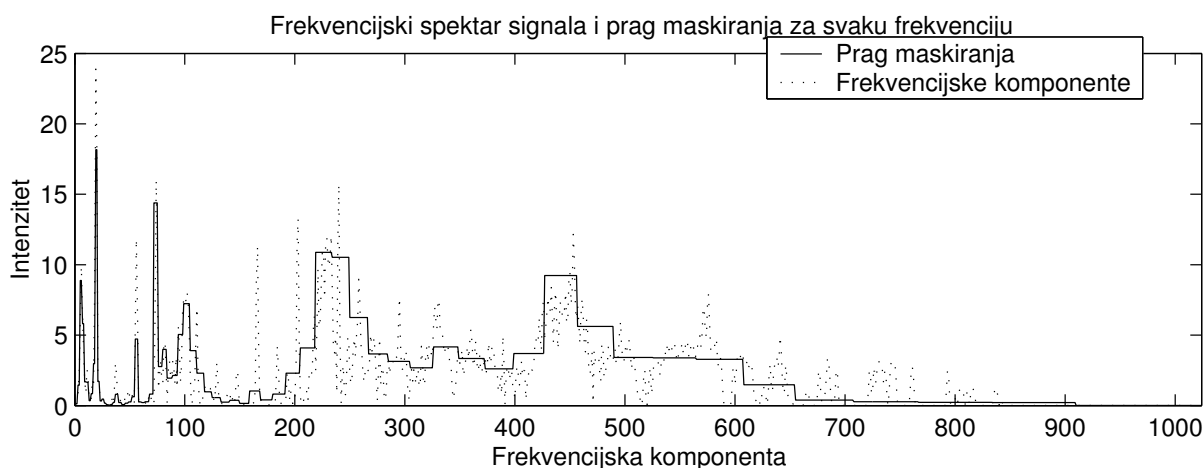
Na ovaj način je dobiven energetska prag maskiranja. Njega još treba usporediti sa krivuljom *apsolutnog praga čujnosti*, na način da se energetska prag na mjestima gdje je niži od apsolutnog praga čujnosti, podigne i poistovjeti sa krivuljom apsolutnog praga čujnosti. Proces je prikazan na slici 7.11.

Krivulja apsolutnog praga maskiranja je uzeta iz tablica informativnog dijela standarda, a može se i proračunati prema Formuli 3.1.

Slika 7.11: Superpozicija apsolutnog praga čujnosti sa proračunatim minimalnim pragom maskiranja. Rezultat superpozicije je konačni prag maskiranja.



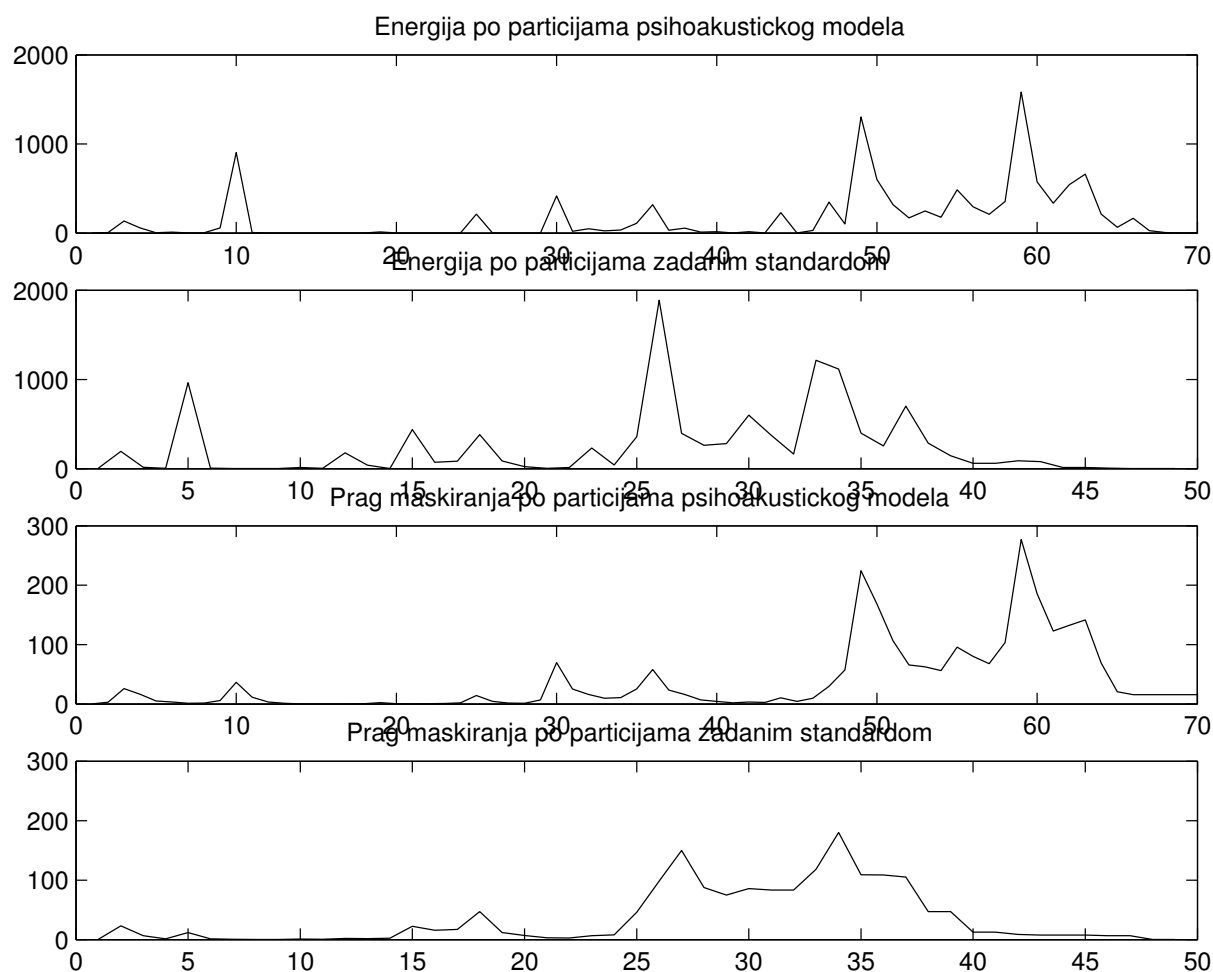
Nakon proračuna minimalnog praga maskiranja po particijama psihoakustičkog modela, prebacimo prag maskiranja iz *Bark* domene kritičnih pojaseva u frekvencijsku domenu, radi lakše vizualizacije dobivenog rezultata (Slika 7.12). U samom koderu ovaj korak nije potreban.

Slika 7.12: Prag maskiranja u frekvencijskoj domeni.

Nakon proračuna minimalnog praga maskiranja unutar particija psihoakustičkog modela, pristupa se preslikavanju dobivenih rezultata u domenu particija zadanih standardom. Kako je već rečeno, standardizirane širine particija su određene kompromisom između što bolje aproksimacije kritičnih pojaseva sa jedne strane, i smanjenja složenosti sa druge strane. Zbog ovog razloga se unutar psihoakustičke analize upotrebljava finija podjela frekvencijskog područja prema kritičnim pojasevima ljudskog slušnog sistema pa je rezultate na kraju potrebno prilagoditi standardiziranim širinama pojasa.

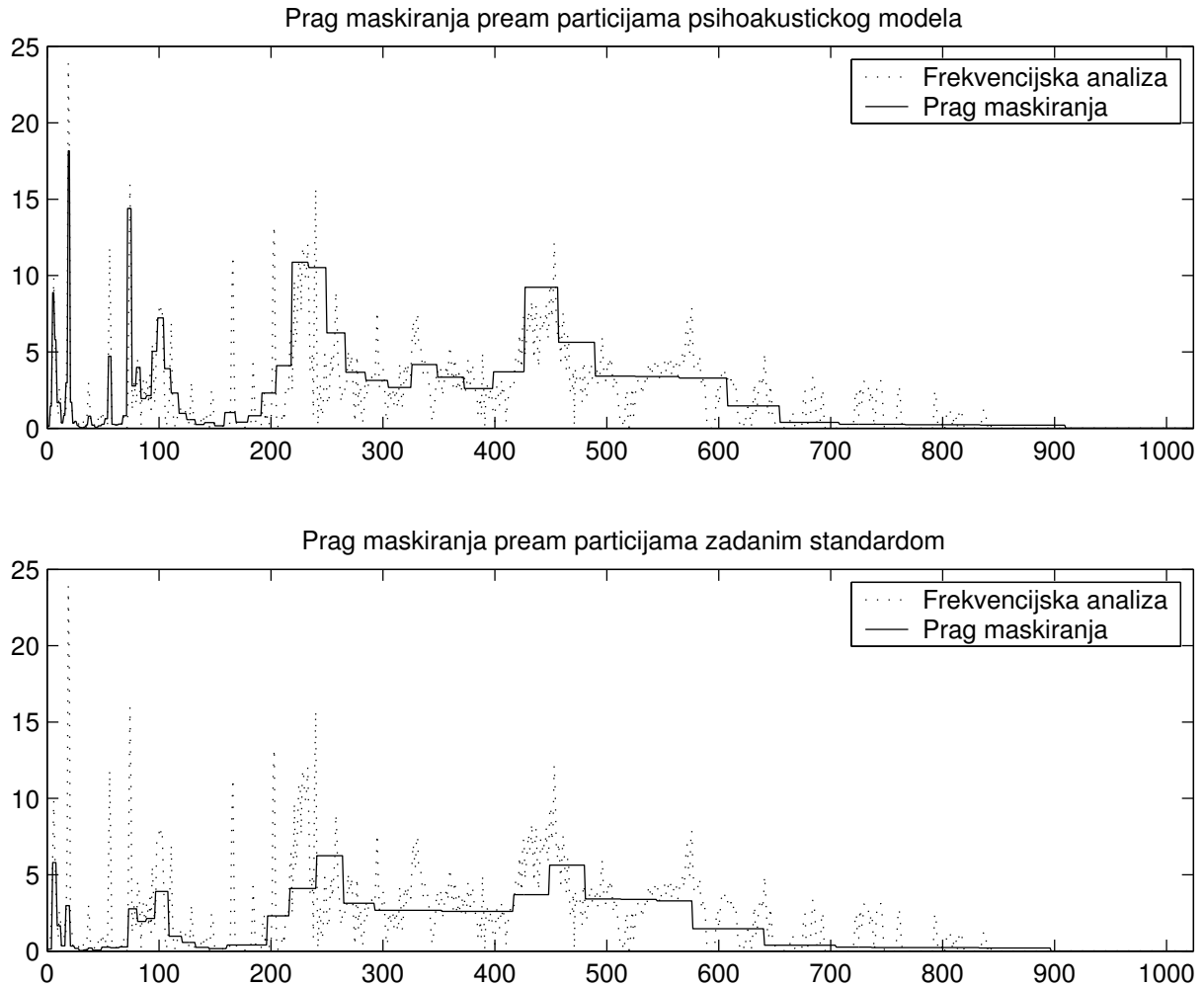
Za frekvenciju uzorkovanja od 44.1 kHz (kao u primjeru) standardom je zadano 49 pojaseva za koje je potrebno proračunati pragove maskiranja. Unutar psihoakustičkog modela upotrebljava se bolja rezolucija od 70 pojaseva. Na slici 7.13 je prikazano preslikavanje energije i pragova maskiranja iz jedne u drugu domenu.

Slika 7.13: Nakon proračuna pragova maskiranja unutar 70 pojaseva u psihoakustičkom modulu, pristupa se preslikavanju pragova maskiranja u 49 pojaseva zadanih standardom (za signal frekvencije uzorkovanja 44.1 kHz).



Ovom operacijom je završen proračun pragova maskiranja. Na slici 7.14 je prikazano poređenje dobivenih pragova maskiranja po frekvencijskim komponentama.

Slika 7.14: Prag maskiranja se preslikava iz maskirnih particija upotrebljivanih u psihoakustičkom modelu u particije zadane standardom. U slučaju da jednu particiju zadanu standardom prekriva više particija psihoakustičkog standarda, prag maskiranja za zadanu particiju postaje minimalni prag maskiranja psihoakustičkih particija koje ga prekrivaju. Ovaj efekt se najviše primjećuje na niskim frekvencijama.



7.5.8 Percepcijska entropija

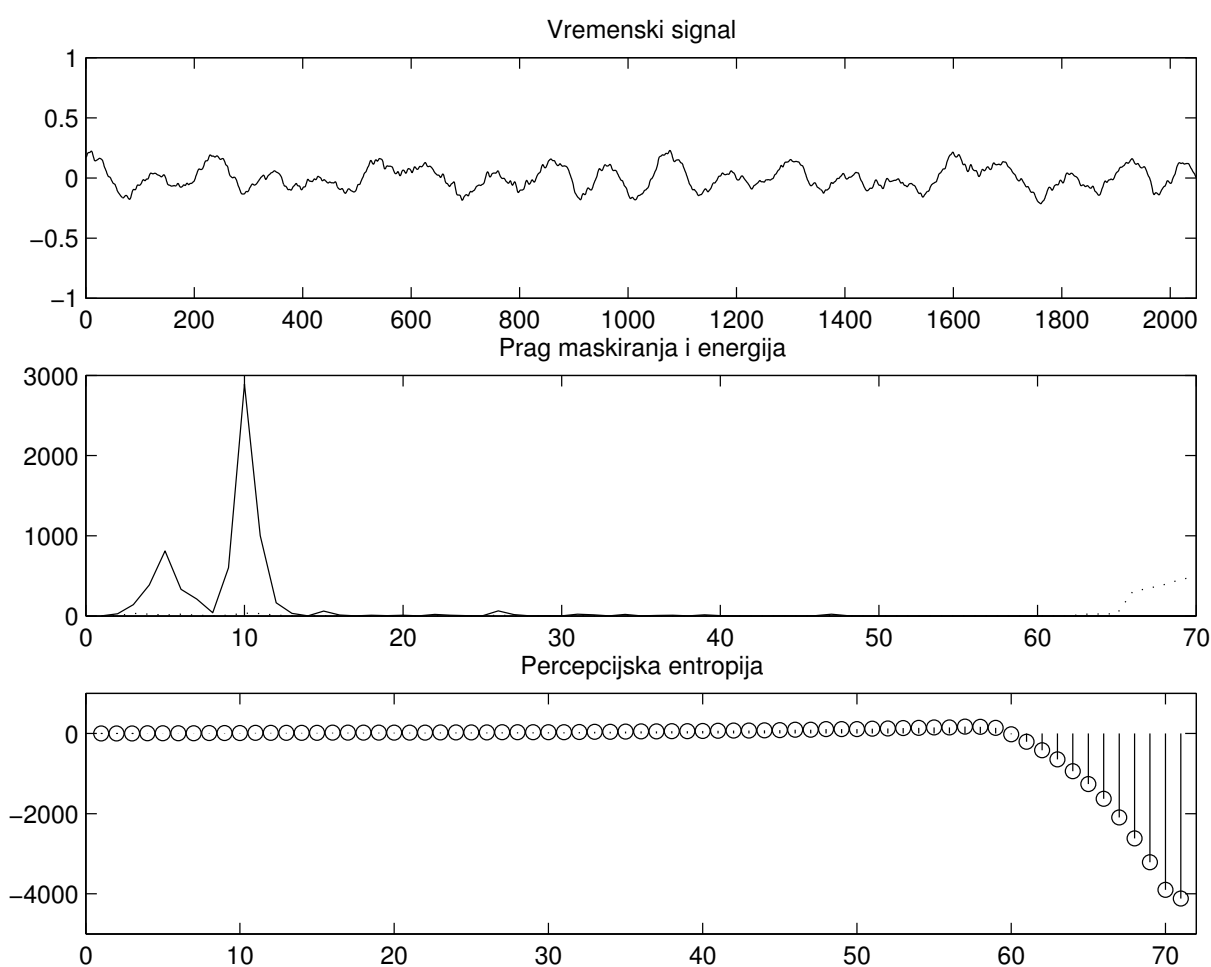
RAAC koder računa percepcijsku entropiju pomoću omjera energije svake particije i minimalnog praga maskiranja svake particije, kako je predloženo u informativnom dijelu standarda. Ukupna entropija signala se računa kao zbroj proračunatih entropija po particijama:

$$PE_{part} \approx \frac{\text{Energija}_{part}}{\text{Prag Maskiranja}_{part}} \quad (7.11)$$

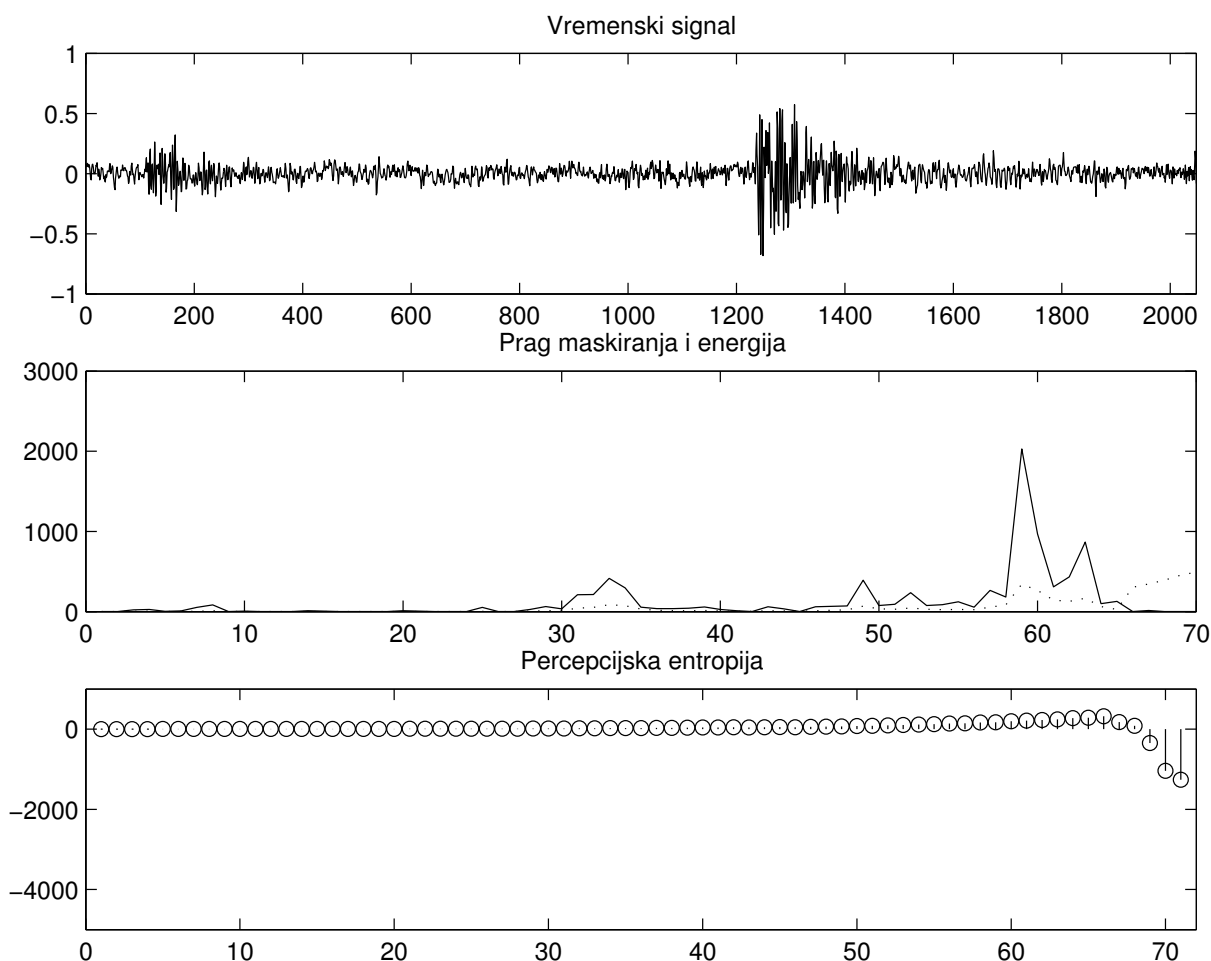
$$PE_{uk} = -\sum PE_{part}$$

Veću entropiju imaju složeniji, više kaotični, nestacionarni signali. Kod ovakvih signala izražene su više frekvencije (tranzijenti su većinom visoko-frekvencijski), pa je prag maskiranja na visokih frekvencijama viši nego kod stacionarnih signala. Samim tim i omjer energije i praga maskiranja je manji, pa ukupna entropija raste. Slike 7.15 i 7.16 potvrđuju ovakvo razmišljanje. Entropija stacionarnog signala je puno manja nego entropija složenog, nestacionarnog tranzijenta.

Slika 7.15: Percepcijska entropija stacionarnog signala. Konačna vrijednost je manja od -4000.



Slika 7.16: Percepcijska entropija tranzijentnog signala. Konačna vrijednost je veća od -1000.



Informaciju o entropiji bloka signala RAAC koristi na dva mjesta. Prvo, percepcijska entropija se koristi kod definiranja tipa bloka obzirom na tranzijentnost. Drugo, odluka o upotrebi PNS algoritma također ovisi o percepcijskoj entropiji signala.

7.5.9 Detekcija tranzijentnosti signala

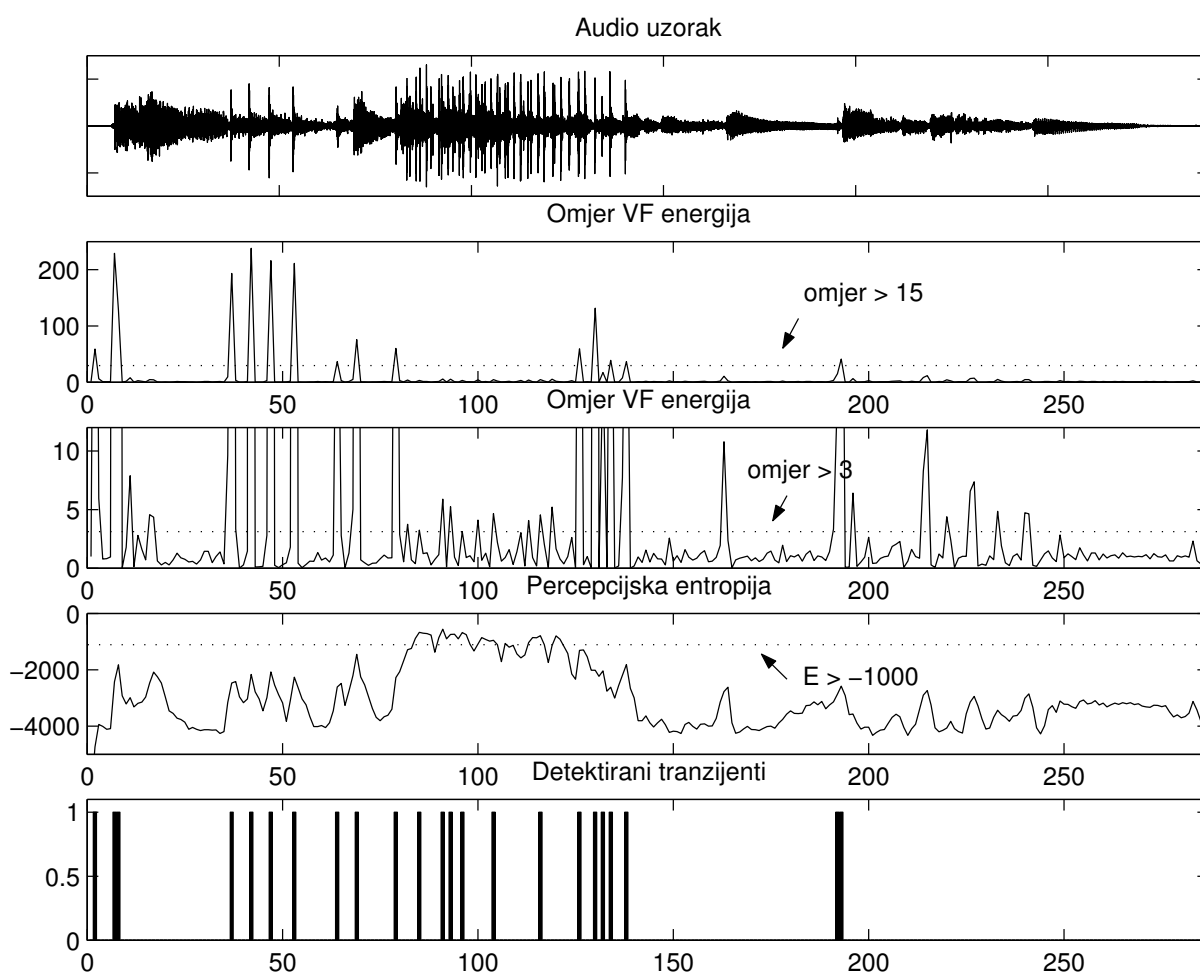
Detekcija tranzijentnosti u RAAC koderu počiva na kombinaciji dvije metode objašnjene u Poglavlju 4.2.2.3.3. Spomenute metode su: *Detekcija tranzijenta analizom distribucije energije* i *Detekcija tranzijenta proračunom percepcijske entropije*.

Detekcija analizom distribucije energije je izvedena proračunom omjera energija visokofrekvencijskih komponenti između trenutnog i zadnjeg bloka. U obzir se uzima samo energija frekvencija iznad 5 kHz jer su tranzijenti pretežito visokofrekvencijski. Percepcijska entropija se računa prema Formuli 7.11.

Dobiveni omjer VF energija se pamti, nakon čega se njegov iznos zajedno sa iznosom percepcijske entropije upotrebljava za donošenje odluke o tranzijentnosti signala. Odluka se donosi na način da se blok audio signala proglašava tranzijentnim ukoliko je ispunjen jedan od sljedeća dva uvjeta:

- 1) omjer energija dva susjedna bloka veći je od $m1$,
- 2) omjer energija dva susjedna bloka veći je od $m2$ ($m2 < m1$) i entropija je veća od $m3$.

Slika 7.17: Detekcija tranzijenta kombiniranjem dva algoritma za audio uzorak kastanjete (engl. *castanets*). Blok se bezuvjetno proglašava tranzijentnim ukoliko je skok u VF energiji veći od 15. Ukoliko je skok veći od 3 puta, potrebno je i da je entropija veća od -1000, jer se time dodatno obezbjeđuje sigurnost algoritma.



Sljedeće vrijednosti parametara su eksperimentalno potvrđene kao prihvatljive:

$$m1 = 15$$

$$m2 = 3$$

$$m3 = -1000$$

Primjer upotrebe algoritma za detekciju tranzijentnosti za poznati audio uzorak *castanets* sa odabranim parametrima prikazan je na Slici 7.17.

7.6 Zaključak

RAAC izvedba psihoakustičkog modela bazirana je na modelu predloženom u informativnom dijelu standarda. RAAC međutim unosi i značajne promjene. Glavna promjena koja presudno utječe na rezultat psihoakustičke analize je funkcija širenja maskiranja. Ova funkcija u RAAC koderu je složenija od jednostavne trokut funkcije predložene standardom. Sljedeća promjena se odnosi na algoritam predviđanja kojim se računa mjera kaotičnosti signala, a iz nje i tonalitet komponenti signala. Algoritam predviđa konstantnost amplitude tonalnog signala, za razliku od prijedloga iznesenog u standardu kojim se predviđa linearni rast amplitude tonalnog signala. Također, predložen je sasvim nov postupak računanja tranzijentnosti signala.

Realizirani psihoakustički model će najbolje biti verificiran slušnim testovima samog RAAC koda.

Poglavlje 8

8 MPEG-4 *alati* izvedeni u RAAC koderu

8.1 *Vremensko uobličavanje šuma*

8.1.1 Općenito

Vremensko uobličavanje šuma (engl. *Temporal Noise Shaping – TNS*) je jedna od novijih metoda koja se u audio kodiranju primjenjuje za suzbijanje poznatog efekta pred-jeke. Prvi put je predstavljena 1996. godine [37], [38], [39].

TNS metoda je prvenstveno motivirana činjenicom da su za transparentno audio kodiranje još uvijek najveći problem tranzijentni i VF audio signali. Glavni razlog je vremenski aspekt maskiranja: da bismo postigli percepcijski transparentno kodiranje, kvantizacijski šum ne smije prekoračiti vremenski ovisan prag maskiranja. U praksi ovaj cilj nije lako postići jer se šum uveden kvantizacijom u frekvencijskoj domeni širi kroz cijeli blok u vremenskoj domeni. MPEG-4 GA koder vrši vremensko-frekvencijsku transformaciju nad blokovima dužine 2048 uzoraka, što je kod frekvencije uzorkovanja 44.1 kHz jednako 46.44 ms. Dakle, šum se širi kroz cijeli blok od nešto manje od 50 milisekundi. Ovakvo širenje šuma predstavlja problem u slučaju kada je ulazni signal tranzijentan: šum će se preliti kroz cijeli vremenski blok, a prije nastupa tranzijenta biti će jako izražen jer ne postoji signal koji bi ga maskirao (Slika 4.7). Prema svojstvima ljudskog slušnog sistema, šum pred-jeke će biti maskiran samo ako se prostire manje od 2 do 5 milisekundi prije pojave tranzijenta (Slika 3.9).

TNS algoritam ima svoj izvor u dvije činjenice:

- 1) svojstvo vremensko-frekvencijske dualnosti između spektralne envelope i kvadratne Hilbertove envelope, i
- 2) uobličavanje kvantizacijskog šuma kodiranjem metodom predviđanja u otvorenoj petlji.

Razmotrimo najprije gornja dva svojstva.

8.1.2 Linearno predviđanje u vremenskoj domeni

Metoda kojom se na osnovu prethodnih uzoraka ulaznog signala predviđa vrijednost sljedećeg uzorka naziva se metodom predviđanja. Najznačajnija primjena predviđanja je kompresija signala koja se ostvaruje eliminiranjem redundancije u signalu. Umjesto kodiranja samog ulaznog signala, kodira se razlika stvarne i predviđene vrijednosti, te se na taj način postiže određena kompresija.

Predviđena vrijednost signala se najčešće računa kao linearna kombinacija prošlih uzoraka:

$$y(n) = \sum_{i=1}^N a(i) \times x(n-i) \quad (8.1)$$

pa se spomenuta metoda naziva metodom *linearnog predviđanja*. Formula 8.1 može se protumačiti i kao konvolucija ulaznog signala x sa koeficijentima predviđanja $a(i)$, što ustvari predstavlja filtriranje ulaznog signala FIR filtrom. Koeficijenti $a(i)$ se odabiru tako da se na izabrani način minimizira greška predviđanja. Najčešće se minimizira srednja kvadratna greška:

$$E = \sum_{n=-\infty}^{+\infty} e^2(n) = \sum_{n=-\infty}^{+\infty} \left(x(n) - \sum_{i=1}^N a(i) \times x(n-i) \right)^2 \quad (8.2)$$

Pronalaženjem minimuma od E :

$$\frac{\partial E}{\partial a(i)} = 0, \quad 1 \leq i \leq N \quad (8.3)$$

dobiva se niz od N linearnih jednadžbi po $a(i)$ koje se mogu efikasno riješiti rekurzivnom *Levinson-Durbinovom* metodom [70] koja nije računski intenzivna pa se često upotrebljava u računarstvu.

Efikasnost metode predviđanja u vremenskoj domeni ovisi ponajprije o dvije stvari. Prvo, mora postojati linearna zavisnost između susjednih uzoraka vremenskog signala, što je slučaj kod izrazito tonalnih signala. Drugo, blok podataka nad kojim se vrši linearno predviđanje mora biti stacionaran.

8.1.3 Linearno predviđanje u frekvencijskoj domeni

Prije otkrića TNS algoritma, vladalo je uvjerenje da u frekvencijskoj domeni ne postoji način pristupa vremenskim svojstvima signala. Ova pretpostavka sama po sebi nije stajala, jer su sve informacije o signalu prisutne i u frekvencijskoj domeni samo su prikazane na drugi način. Nejasan je međutim bio način na koji bi se iz frekvencijske domene pristupilo vremenskoj domeni, pa se dizajn audio koderu fokusirao na metodu izmjene dužine blokova kao metodu koja bi suzbila efekt pred-jeke preko povećanja vremenske rezolucije signala.

Da bismo pokazali kako iz frekvencijske domene pristupiti vremenskim karakteristikama signala, analizirajmo prvo svojstvo *dualnosti* između frekvencijske gustoća snage signala (engl. *power spectral density* – *PSD*) i kvadratne Hilbertove envelope.

8.1.4 Dualnost

Izrazimo ovisnost prošlih i trenutnih uzoraka vremenskog signala preko autokorelacijske funkcije (koja na određeni način mjeri sličnost između trenutnog signala i njegove zakašnjele verzije):

$$R_{xx}(t) = \int x(\tau) \times x^*(\tau - t) d\tau \quad (8.4)$$

Fourierova transformacija autokorelacijske funkcije $R_{xx}(t)$

$$F \left\{ \int x(\tau) \times x^*(\tau - t) d\tau \right\} \quad (8.5)$$

predstavlja frekvencijsku gustoću snage signala (PSD). Iz PSD direktno se računa mjera spektralne uravnnjenosti (engl. *Spectral Flatness Measure – SFM*) koja predstavlja iznos redundancije koja postoji u analiziranom signalu. SFM teži nuli za sasvim predvidive tonalne signale, dok je za *bijeli* šum ona približno jednaka 1.

Slijedeći svojstvo dualnosti, analizirajmo sljedeću formulu (koja predstavlja kvadrat Hilbertove envelope realnog signala x):

$$F^{-1} \left\{ \int X(\zeta) \times X^*(\zeta - f) d\zeta \right\} \quad (8.6)$$

Hilbertova envelope signala je direktno povezana sa autokorelacijskom funkcijom svoga frekvencijskog spektra. Drugim riječima, envelope vremenskog signala je direktno povezana sa autokorelacijskom funkcijom u frekvencijskoj domeni – ona u stvari jeste inverzna Fourierova transformacija spektralne autokorelacijske funkcije.

Iz ovog razmatranja izvlačimo konačni zaključak: frekvencijska gustoća snage signala i kvadratna Hilbertova envelope predstavljaju dualne aspekte glede vremenske i frekvencijske domene istog signala: iznos redundancije signala koji se može dobiti predviđajući preko frekvencijskih komponenata određen je *neuravnnjenošću* Hilbertove envelope na isti način na koji je redundancija dobivena predviđanjem u vremenskoj domeni određena sa *uravnnjenošću* funkcije frekvencijske gustoće snage signala. Dakle, što je veća neuravnnjenost (kaotičnost) vremenske envelope signala, to bolje se može predviđati preko frekvencijskih komponenti (Slika 8.1). Ovo se upravo događa kod tranzijentnih signala, gdje se opisano svojstvo i koristi.

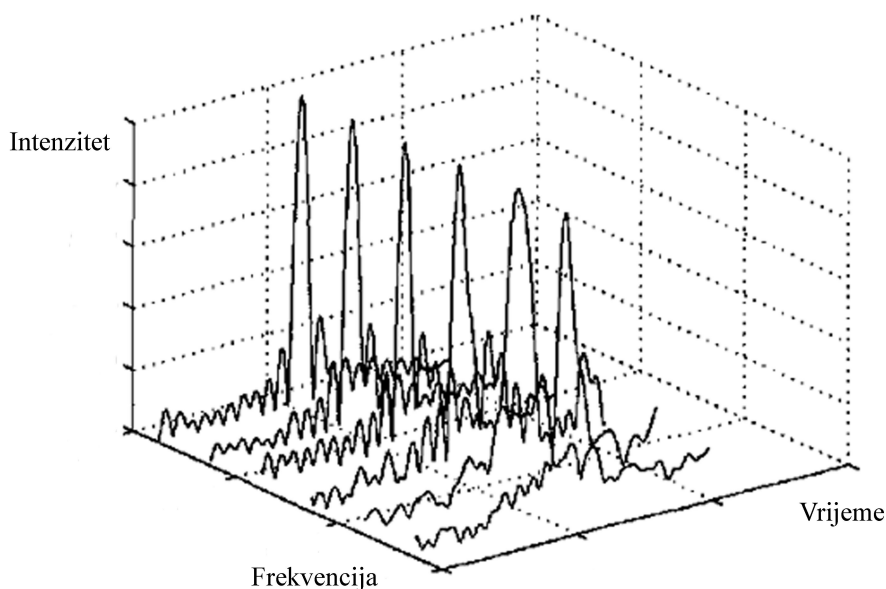
8.1.5 Predviđanje metodama otvorene i zatvorene petlje

Predviđanje se može vršiti dvjema standardnim metodama koje se nazivaju metoda otvorene petlje (engl. *open-loop*) i metoda zatvorene petlje (engl. *closed-loop*). Predviđanje u otvorenoj petlji zasniva se na predviđanju originalnog signala, dok se predviđanje u zatvorenoj petlji zasniva na predviđanju već prije sintetiziranog signala.

Izbor metode otvorene petlje kod TNS algoritma je od presudne važnosti zbog sljedeće činjenice: kod predviđanja metodom otvorene petlje, nakon dekodiranja, spektar signala greške je nalik na spektar originalnog signala, dok je kod predviđanja metodom zatvorene

petlje spektar signala greške ravan (bijeli šum). Ova činjenica se koristi kod TNS algoritma da bi se šum smjestio ispod signala i tako bio njime maskiran.

Slika 8.1: Hilbertove envelope tranzijentnog signala, predstavljene u nekoliko frekvencijskih pojasa. Envelope signala su slične, pa je predviđanje u frekvencijskoj domeni vrlo efikasno.



Efikasnost metode predviđanja u frekvencijskoj domeni, slično predviđanju u vremenskoj domeni, ovisi o dvije stvari. Prvo, mora postojati linearna zavisnost između susjednih frekvencijskih komponenti signala, što je slučaj kod izraženo tranzijentnih signala. Također, signal mora imati sličnu frekvencijsku envelopu preko svih pojaseva unutar područja primjene predviđanja (Slika 8.1).

8.1.6 Zaključak

Kombiniramo li prethodna dva slučaja, slučaj dualnosti i predviđanja metodom otvorene petlje, dolazimo do sljedećeg zaključka: ako se predviđanje metodom otvorene petlje primijeni na frekvencijske komponente signala, vremenski oblik kvantizacijskog šuma nakon dekodiranja biti će prilagođen envelopi vremenskog signala.

Ovo svojstvo se koristi na način da se kvantizacijski šum perpcijskog audio koderu ciljano stavi ispod samog signala. Na ovaj se način postiže skrivanje šuma ispod samog signala, pa se izbjegava neugodno širenje šuma kod tranzijentnih signala.

8.1.7 TNS realizacija u RAAC koderu

TNS algoritam se lako uklapa u klasičnu strukturu percepcijskog audio koder kakav je i MPEG-4 GA koder. Blok TNS filtriranja ubacuje se u arhitekturu koder odmah nakon filteranskog sloga (Slika 5.2), tako da se na spektralnim komponentama vrši filtriranje u mjestu (engl. *in-place*), a daljnji proračun se nastavlja na filtriranim komponentama. Komponente dobivene vremensko-frekvencijskom transformacijom se zamjenjuju filtriranim vrijednostima. Na strani dekoder, proces je obrnut: vrši se inverzno filtriranje u mjestu, inverznim predviđanjem/filtriranjem.

Kao što je i rečeno, blok sa TNS filtriranjem se ubacuje u arhitekturu koder odmah na početku, nakon vremensko-frekvencijske transformacije. Prije početka samog filtriranja treba odrediti nekoliko parametara koji su neophodni za definiranje rada samog TNS bloka.

Prvo, unaprijed se izabere frekvencijski raspon signala nad kojim će TNS filtriranje biti primijenjeno. Kako je TNS filtriranje namijenjeno za suzbijanje efekta pred-jeke, a on se javlja kod jako tranzijentnih signala, koji opet imaju jako izražene visoke frekvencije, zaključujemo da je primjena TNS filtriranja nepotrebna na niskim frekvencijama. Obično se TNS filtriranje primjenjuje na frekvencijama višim od oko 3 kHz, pa sve do najviše frekvencije koja se može doseći sa jednim filtrom, obzirom na red filtra. Standardom je definirana upotreba jednog ili više filtera unutar jednog audio bloka. U slučaju da se primjenjuje više filtera, svaki se filter primjenjuje na ciljani, uži opseg frekvencija.

Također, prije primjene TNS filtriranja, potrebno je definirati *dobitak predviđanja* (engl. *prediction gain*) koji mora biti postignut da bi se filtriranje uopće isplatilo. RAAC koder uzima vrijednost 1.5, kao granicu ispod koje se TNS ne primjenjuje.

Također, potrebno je definirati i red TNS filtra. Iako je ova veličina standardom limitirana na 20 kod *Main* profila, 12 kod *Low Complexity* profila, i 7 kod kratkih blokova bez obzira na profil, ipak se uvijek ne upotrebljava maksimalni red filtra nego se na osnovu veličine proračunatih koeficijenata red filtra dodatno smanjuje. Preporučuje se zanemarivanje koeficijenata na *repu* filtra ako su manji od određene minimalne vrijednosti. RAAC koder zanemaruje koeficijente manje od 0.2.

Postupak primjene TNS algoritma započinje proračunom LPC (engl. *Linear Prediction Coefficients*) koeficijenata. Za proračun LPC koeficijenata upotrebljava se standardna *Levinson-Durbinova* rekurzivna metoda. Kao rezultat proračuna dobije se niz *koeficijenata*

refleksije i *dobitak predviđanja*, na osnovu kojeg se odlučuje o uključenju TNS filtriranja u tekući audio blok ili ne.

Nakon proračuna *refleksijskih* koeficijenata, vrši se njihovo kvantiziranje na 3 ili 4 bita. RAAC uvijek vrši kvantizaciju na 4 bita. Na osnovu ovih koeficijenata, određuje se red filtra, tako da se koeficijenti na kraju filtra koji su manji od 0.2 izbacuju. Broj preostalih koeficijenata određuje red filtra.

Nakon kvantiziranja refleksijskih koeficijenata vrši se filtriranje *u mjestu* nad MDCT frekvencijskim koeficijentima. Prije filtriranja potrebno je iz refleksijskih koeficijenata koji su kvantizirani, dobiti završne LPC koeficijente. Ovo se postiže poznatom *koračnom* procedurom (engl. *step-up*). Objasnimo prvo ulogu refleksijskih koeficijenata. Pošto je stabilnost filtra jako ovisna o LPC koeficijentima, a oni se u procesu kvantizacije mogu znatno promijeniti, onda se proces kvantizacije provodi nad refleksijskim koeficijentima čije male promjene ne utječu znatno na stabilnost filtra. Zbog ovog razloga se u kodiranom toku i prenose refleksijski, a ne završni LPC koeficijenti.

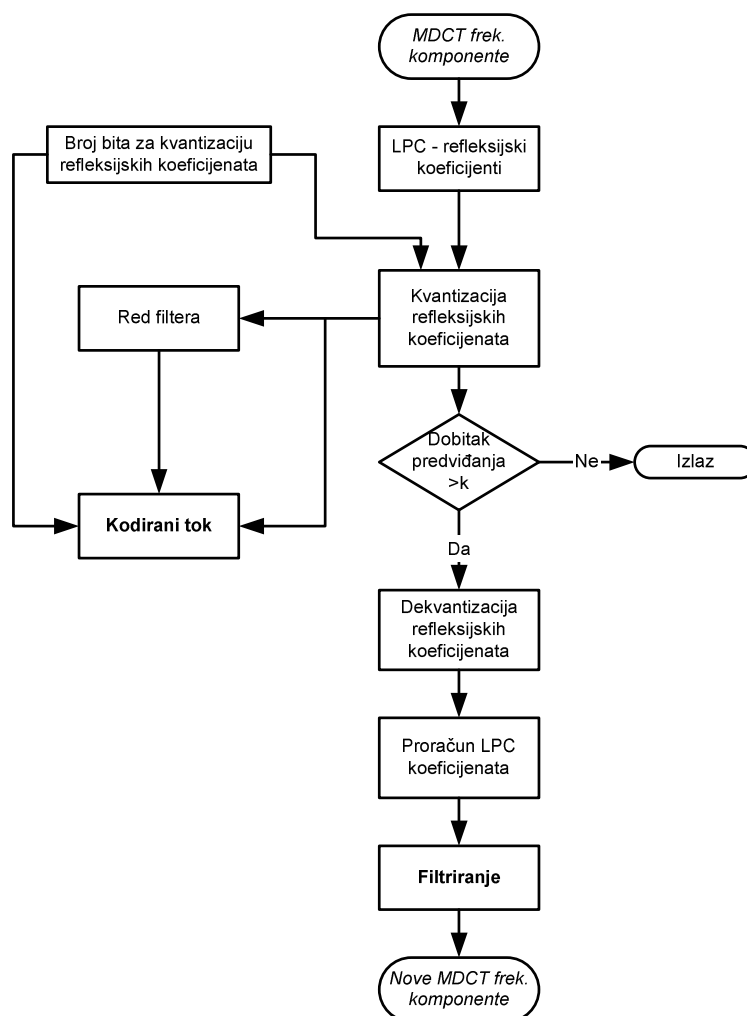
Dakle, procesom dekvantizacije vrijednosti refleksijskih koeficijenata (koje se prenose u kodiranom toku) dobiju se dekvantizirani refleksijski koeficijenti na osnovu kojih se računaju završni LPC koeficijenti. LPC koeficijenti se upotrebljavaju kao koeficijenti TNS filtra. Oni se standardiziranom metodom primjenjuju na MDCT frekvencijske komponente audio signala. Ovaj, u stvari FIR filter, je filter bez polova (engl. *all-zero*), za razliku od inverznog filtriranja na strani dekodera koje se vrši sa filtrom bez nula (engl. *all-pole*). Formula filtra na strani koda je sljedeća:

$$y(n) = x(n) + a(1) * x(n-1) + \dots + a(RF) * x(n-RF) \quad (8.7)$$

gdje su $a(\)$ LPC koeficijenti, a RF proračunati red filtra.

Na Slici 8.2 je prikazan tok primjene TNS algoritma u RAAC koderu.

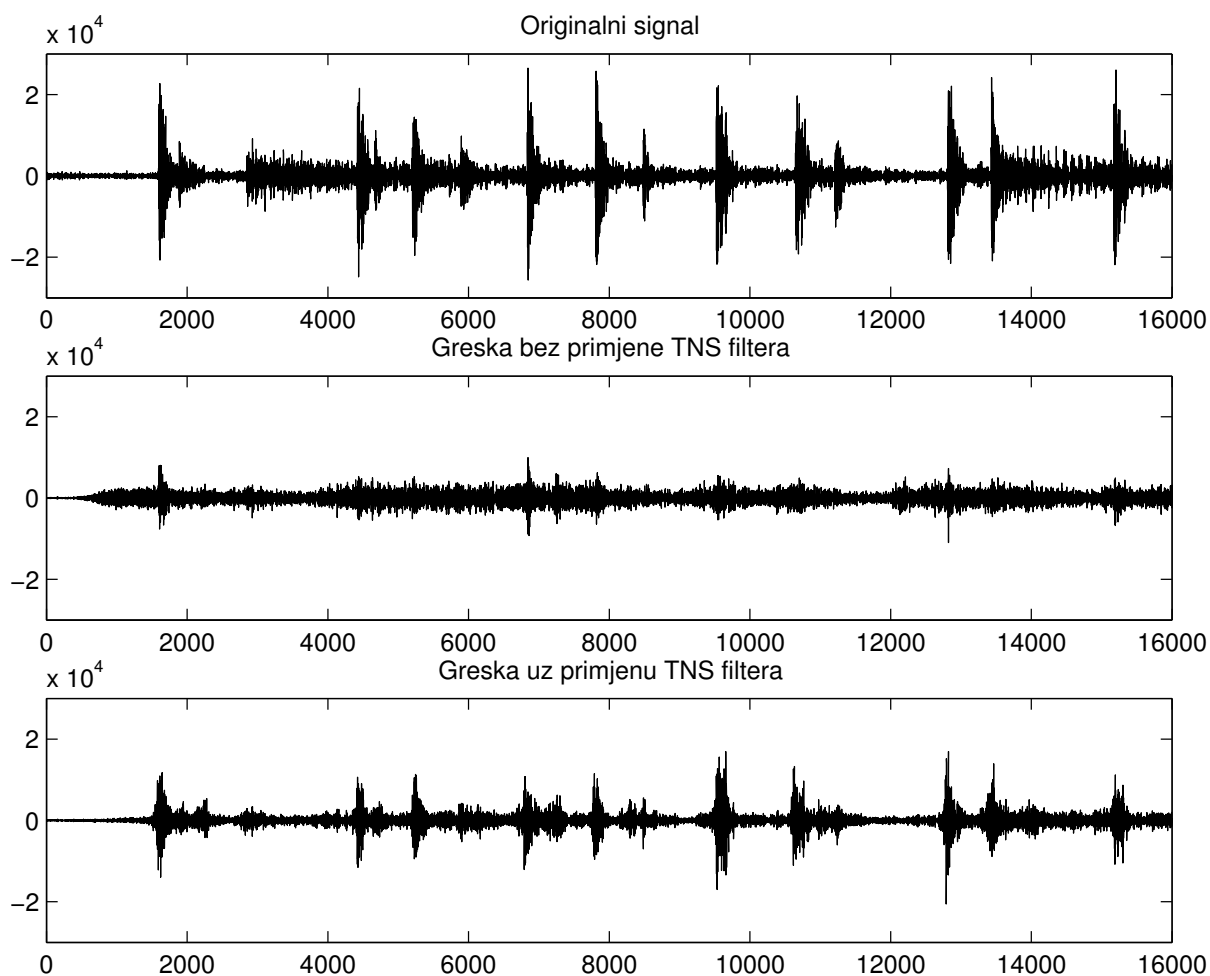
Slika 8.2: Blok dijagram primjene procesa TNS filtriranja. TNS filtriranje se primjenjuje odmah nakon filtarskog sloga. Efikasnost primjene LPC predviđanja u frekvencijskoj domeni ovisi o karakteristikama signala: što je signal više tranzijentan, to bolje je moguće predviđati u frekvencijskoj domeni. Kvantitativno, efikasnost je određena sa vrijednosti *dobitka predviđanja*: ukoliko ova vrijednost nije iznad određenog praga (~ 1.5), TNS se ne primjenjuje.



Daljnje kodiranje se nastavlja nad setom filtriranih MDCT frekvencijskih komponenti.

Na Slici 8.3 prikazan je primjer upotrebe TNS filtriranja kod RAAC koderu. Očito je da je šum u vremenskoj domeni (koji je uveden kvantizacijom) nakon primjene TNS algoritma postao proporcionalan amplitudi signala. Bez primjene TNS filtriranja kvantizacijski šum je gotovo jednolično raspoređen bez obzira na amplitudu signala.

Slika 8.3: Razlika originalnog i rekonstruiranog signala sa i bez upotrebe TNS filtriranja. Bez primjene TNS filtra, greška je ujednačena i neznatno ovisi o envelope signala. Primjenom TNS filtra sa predviđanjem metodom otvorene petlje, greška postaje proporcionalna signalu. Na ovaj način se jači šum stavlja ispod jačeg signala pa tako biva znatnije maskiran. Ova činjenica najviše pogoduje smanjenju širenja šuma kod tranzijentnih signala, pa se upotrebljava kao zamjena metodi promjene dužine blokova.



Utjecaj TNS filtriranja na konačnu kvalitetu signala je kvantitativno predstavljen u [38]. Nakon subjektivnih testiranja, zaključeno je da TNS filtriranje najviše pomaže u kodiranju govora gdje je tradicionalna metoda promjene dužine blokova neefikasna. Subjektivni testovi su pokazali porast od skoro 1.0 na MOS (engl. *Mean Opinion Scale*) skali za testni uzorak njemačkog govora. TNS filter je pomogao u suzbijanju fenomena tzv. dvostrukog govornika (engl. *double speak*) koji se manifestira u pojavi drugog tihog govornika u pozadini originalnog kao posljedica kvantizacijskog šuma. Određeno poboljšanje je uočeno i kod

tranzijentnih dijelova signala *ksilofon* instrumenta (+0.3). Kod drugih signala TNS filtriranje nije dalo znatnih poboljšanja.

Rezultati upotrebe TNS algoritma kod RAAC koderu su prikazani u Poglavlju 9.2.8.7.

8.2 *Združeni-stereo* kodiranje

8.2.1 Općenito

U MPEG-4 GA standardu definirane su dvije metode *združenog-stereo* kodiranja:

- 1) *Zbroj/razlika* kodiranje i
- 2) *Intenzitet-stereo* kodiranje.

Zbroj/razlika metoda se zasniva na principima eliminacije sličnosti između različitih kanala stereo signala [53]. Iako se na prvi pogled čini da postoje velike sličnosti između dva kanala stereo signala u vremenskoj domeni, to u praksi nije slučaj. Činjenica je da samo frekvencijski spektar pokazuje jake sličnosti koje bi se mogle iskoristiti u audio kodiranju. Zamislimo jednostavno stereo snimanje zvuka: ako je kašnjenje signala zbog sobne akustike manje od rezolucije filtarskog sloga (npr. 50 ms), rezultatni spektar signala će se pojaviti u oba kanala stereo signala. Došlo je dakle, do suvišnog ponavljanja informacije.

Intenzitet-stereo metoda nastoji izbaciti irelevantne informacije iz signala također se oslanjajući na sličnost dva kanala [8] [25] [35] [36]. Imajući na umu činjenicu da je ljudski slušni sistem manje osjetljiv na fazu signala na višim frekvencijama (Poglavlje 2.6), dolazimo do zaključka da je moguće iz signala izbaciti irelevantne dijelove koji sadrže informaciju o izvoru visoko-frekvencijskog stereo zvuka – dolazi li zvuk iz lijevog ili desnog zvučnika sasvim je irelevantno na visokim frekvencijama jer čovjek to ne čuje.

8.2.2 *Zbroj/razlika* kodiranje

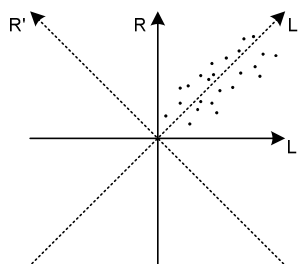
8.2.2.1 Općenito

Počecima *združenog-stereo* kodiranja smatra se *zbroj/razlika* (engl. *summ-difference* ili engl. *mid/side* – *MS*) kodiranje stereo signala. Umjesto kodiranja jednog po jednog kanala posebno,

kodiran je prvo zbroj, a onda i razlika signala. Na ovaj način je iskorištavana sličnost dva signala, te se postizala ušteda.

Zbroj/razlika kodiranje možemo grafički zamisliti prema Slici 8.4. Ako je većina podataka koncentrirana oko osi L' , a to znači da su stereo signali slični, doći će do značajne uštede kodiranjem zbroja i razlike kanala, jer će razlika (*razlika* kanal) biti blizu nule. Proces se može zamisliti na način da je signal rotiran oko osi L' (45°). Međutim, za stvarne signale vrlo je teško očekivati ovakvu koreliranost između lijevog i desnog kanala stereo signala.

Slika 8.4: Rotiramo li jako koreliran stereo signal oko osi L' , rezultirajući R' kanal će biti koncentriran oko nule. Na ovaj način značajno se štedi na osnovu eliminiranja redundancije lijevog i desnog kanala. Za stvarne signale je teško dobiti ovakvu sličnost lijevog i desnog kanala, pa se predlaže prilagođavanje kuta rotacije karakteristikama signala.



Glavne karakteristike *zbroj/razlika* kodiranja su:

- smanjenje redundancije signala kroz cijeli frekvencijski spektar,
- ušteda u bitskom protoku jako ovisna o sličnosti dva kanala,
- moguća rekonstrukcija bez gubitaka zanemari li se kvantizacija.

8.2.2.2 RAAC izvedba *zbroj/razlika* kodiranja

Zbroj/razlika kodiranje je vrlo fleksibilno uklopljeno u MPEG-4 GA standard. Za svaki pojas kvantizatora odlučuje se o upotrebi *zbroj/razlika* ili običnog *ljevi/desni* kodiranja, a u kodiranom toku se to signalizira jednim bitom. Moguće je imati proizvoljan broj prebacivanja između ova dva tipa kodiranja bez dodatnog pogoršanja kvalitete signala.

Proces kodiranja teče na sljedeći način:

- donosi se odluka o *zbroj/razlika* kodiranju za određeni pojas kvantizatora,
- izvršava se transformacija $l-d$ u $z-r$ prema formuli:

$$\begin{aligned} z &= \frac{l+d}{2} \\ r &= \frac{l-d}{2} \end{aligned} \quad (8.8)$$

- nastavlja se procedura kao i kod običnog *lijevi/desni* kodiranja.

Glavni problem kod *zbroj/razlika* kodiranja je donošenje odluke u kojim pojasevima kvantizatora će se upotrijebiti, a u kojim ne, obzirom da *zbroj/razlika* kodiranje mora donijeti uštedu u potrošnji bitova za određeni dio signala.

Jedna od raširenijih metoda realizacije je sljedeća: određeni dio signala se kodira na oba načina, a zatim se vidi koja metoda koristi manje bitova. Ovaj način je međutim nepogodan zbog nefleksibilnosti i sporosti. Nefleksibilan je zato što ne može iskoristiti mogućnosti upotrebe *zbroj/razlika* kodiranja unutar pojedinih pojaseva, nego bi se odluka donijela na bazi cijelog bloka. Metoda je spora zato što bi zahtijevala dupli proces kvantizacije i kodiranja.

U RAAC koderu je realizirana drugačija i istodobno jedna od prilično konzervativnih metoda, koja ne osigurava do kraja iskorištavanje mogućnosti *zbroj/razlika* kodiranja za određeni blok signala, ali zato donosi siguran dobitak u bitskom protoku u pojasevima u kojima je primijenjena. Također, i računski nije zahtjevana pa ne usporava proces kodiranja.

Odluka o kodiranju svakog pojasa kvantizatora se donosi na sljedeći način:

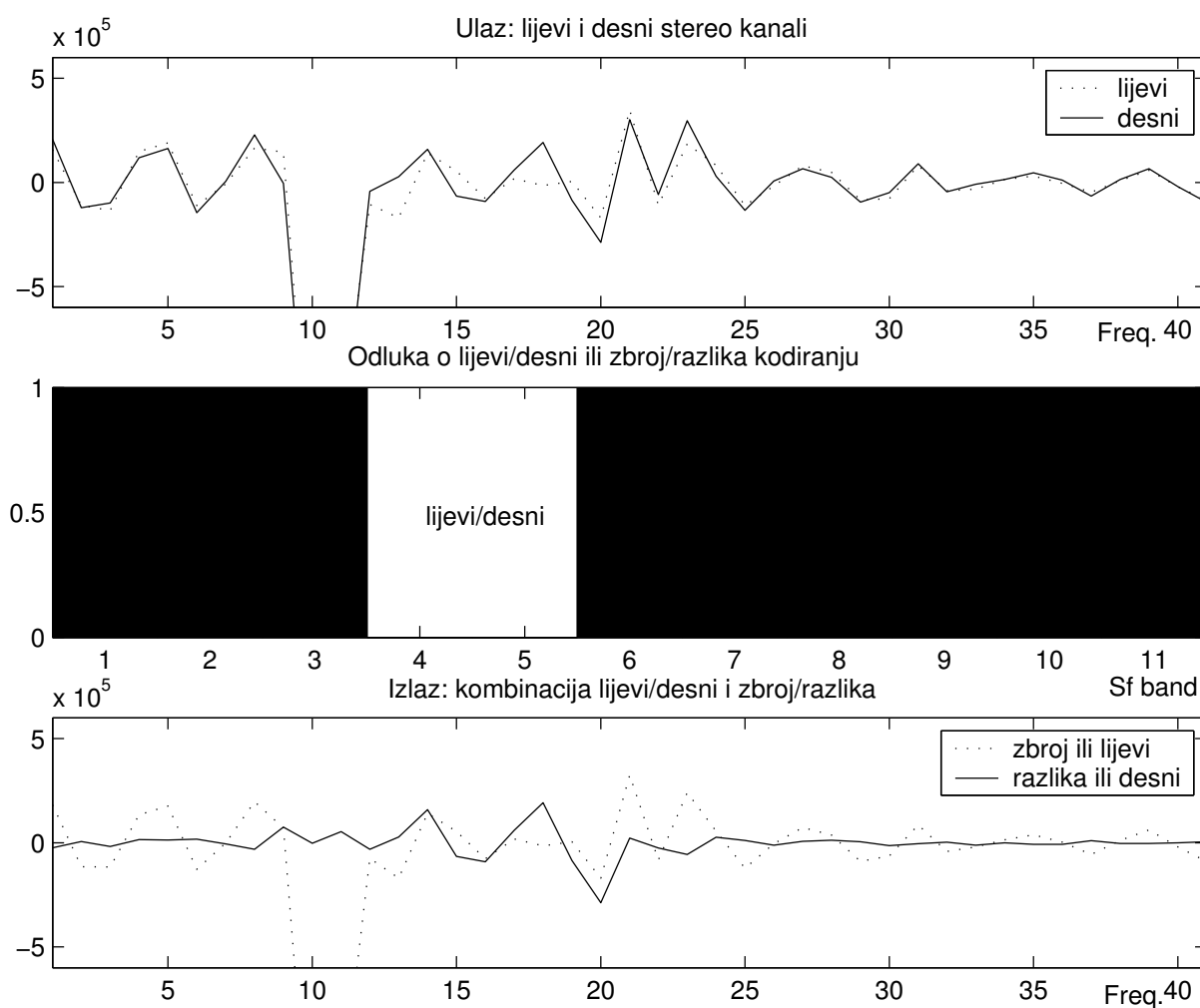
- unutar svakog pojasa uspoređuje se intenzitet frekvencijskih komponenti lijevog i desnog kanala,
- komponente čiji se intenzitet međusobno ne razlikuje za više od unaprijed postavljenog omjera *omjer_zr* markiraju se kao *da_zr* komponente,
- nakon što su sve komponente analizirane, u pojasu se uspoređi broj *da_zr* komponenti sa brojem *ne_zr* komponenti,
- ako je udio *da_zr* komponenti iznad određene granice *granica_zr*, cijeli pojas se kodira kao *zbroj/razlika*, u suprotnom pojas se kodira kao standardni *lijevi/desni* pojas.

Relativno stroga granica za proglašenje jednog para frekvencijskih komponenti kao *da_zr* komponenti, bila bi da se ne razlikuju za više od 15% do 25% (vrijednost *omjer_zr*). Druga

granica koja određuje udio da_{zr} komponenti u pojasu bila bi oko 80% (vrijednost $granica_{zr}$). Na ovaj način se osigurava da se samo jako slični signali kodiraju kao $zbroj/razlika$ kanali. Dobitak na bitskom protoku kod ovako sličnih kanala je osiguran. Jasno je da sasvim isti kanali daju 50% uštede i da će ta ušteda biti manja što su kanali različitiji.

Na Slici 8.5 prikazan je primjer upotrebe $zbroj/razlika$ kodiranja.

Slika 8.5: Odluka o *lijevi/desni* ili *zbroj/razlika* (engl. *mid/side*) kodiranju donosi se na bazi pojasa kvantizatora. Samo dijelovi signala koji su jako slični kodiraju se sa $zbroj/razlika$ pretvorbom. Na slici su svi pojasevi osim 4. i 5. dovoljno slični pa se u njima upotrebljava $zbroj/razlika$ kodiranje. Rezultirajući *razlika* kanal je sasvim blizak nuli što garantira dobitak u bitskom protoku. Pojasevi broj 4 i 5 se kodiraju bez $zbroj/razlika$ pretvorbe jer su suviše različiti.



Treba napomenuti da *zbroj/razlika* kodiranje značajnije ne opterećuje kodirani tok sa dodatnim informacijama. Za svaki pojas se prenosi po jedan bit koji određuje da li se za spomenuti pojas *zbroj/razlika* kodiranje upotrebljava ili ne. Za frekvenciju uzorkovanja od 44.1 kHz, broj pojaseva je 49, pa je opterećenje kodiranog toka sa maksimalno 49 dodatnih bita zanemarivo u usporedbi sa uštedom koju upotreba *zbroj/razlika* kodiranja donosi.

8.2.3 *Intenzitet-stereo* kodiranje

8.2.3.1 Općenito

Poopćimo *zbroj/razlika* metodu na način da rotiranje signala ne bude uvijek isto, nego da se kut rotacije konstantno prilagođava signalu. Nazovimo novodobiveni signal *intenzitet* i i *greška* g .

Rotaciju lijevog l i desnog d kanala možemo opisati sljedećom matričnom jednažbom:

$$\begin{pmatrix} i \\ g \end{pmatrix} = \begin{pmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{pmatrix} x \begin{pmatrix} l \\ d \end{pmatrix} \quad (8.9)$$

Ako je kut $\alpha = 0^\circ$ dobije se početni slučaj kada se kodiraju kanali zasebno. Ako je kut $\alpha = 45^\circ$ dobije se *zbroj/razlika* slučaj.

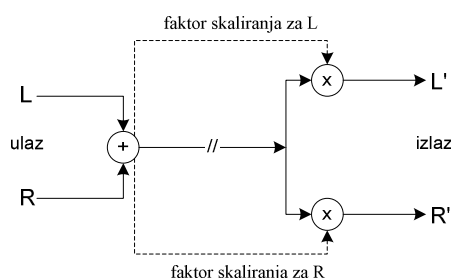
Poopćena metoda kodiranja, međutim, u praksi nije pokazala znatnije uštede. Tek su određena pojednostavljivanja donijela značajnije pomake u smanjenju redundancije signala.

Prvo pojednostavljenje je izbacivanje *greška* g komponente. Ovom pojednostavljenom metodom kodira se samo *intenzitet* signal, kut α i nešto dodatnih informacija. Rekonstruirani signal se sastoji od različitih verzija *intenzitet* signala, dakle amplituda je različita dok je informacija o fazi postala ista. Međutim, kako je uho neosjetljivo na fazu signala na visokim frekvencijama, nego samo na vremensku envelopu energije signala, koja je očuvana, neće doći do narušavanja kvalitete signala.

Drugo pojednostavljenje se zasniva na ideji da se kodirani signal rekonstruira na osnovu zbroja signala lijevog i desnog kanala, umjesto da se oba signala rotiraju. Na ovaj način se cijeli postupak pojednostavljuje (Slika 8.6). Kvantizira se samo jedan kanal koji predstavlja zbroj početnog lijevog i desnog kanala. Također, prenose se i informacije o faktorima

skaliranja svakog kanala posebno (Slika 8.6) ili se prenosi jedan faktor skaliranja. Ako se prenosi jedan faktor skaliranja, onda se prenosi i dodatna informacija o percipiranoj poziciji stereo slike signala (omjer *lijevi/desni*). Ova pojednostavljena metoda *intenzitet-greška* kodiranja naziva se *intenzitet-stereo* metoda i usvojena je kao dio MPEG-4 GA standarda.

Slika 8.6: *Intenzitet-stereo* kodiranje – pojednostavljena ideja *združeni-stereo* kodiranja. Prenosi se samo ukupni intenzitet signala i za svaki kanal faktor skaliranja. Na ovaj način sačuvana je informacija o energiji, dok je informacija o fazi signala izgubljena. Faza je međutim, na visokim frekvencijama irelevantna.



Zaključak je sljedeći:

- *Intenzitet-stereo* kodiranje doprinosi smanjenju redundancije na višim frekvencijama,
- zbog gubitka informacije o fazi, općenito uzevši gubi se valni oblik originalnog signala,
- unutar pojasa gdje se primjenjuje, algoritam štedi 50% prostora jer se u kodiranom toku prenosi samo jedan kanal iz kojeg se rekonstruiraju dva,
- pošto se algoritam zasniva na intenzitetu signala, potrebno je precizno odrediti energiju signala (npr. *FFT*),
- dozvoljena primjena je samo na visokim frekvencijama, najsigurnije iznad 10 kHz.

8.2.3.2 Problemi *intenzitet-stereo* kodiranja

Sljedeće činjenice su glavni nedostaci *intenzitet-stereo* kodiranja:

- ako stereo kanali nisu korelirani, dekodirani signal može biti dosta drugačiji od originalnog u visoko-frekvencijskom dijelu spektra,
- jaki tonalni signali nisu pogodni za ovu vrstu kodiranja jer se pojavljuju jako izobličeni,
- često prelaženje sa običnog *lijevi/desni* kodiranja na *intenzitet-stereo* kodiranje je nepoželjno jer može unijeti dodatni šum,

- posebna skupina zvukova kao što je npr. *pljesak* može izgubiti svoju stereo karakteristiku i skoro se sasvim koncentrirati prema centru između lijevog i desnog kanal.

Združeni-stereo i efekt prvenstva

Združeni-stereo kodiranje može poremetiti vremenske odnose među signalima i na taj način utjecati i na lokalizaciju signala na osnovu efekta prvenstva (Poglavlje 2.7).

Združeni-stereo i BMLD efekt

BMLD efekt (Poglavlje 2.9) dolazi do izražaja kod kodiranja stereo signala, gdje postoje dva maskirajuća signala (ispravan dio signala iz svakog kanala) i dva maskirana signala (šumovi kvantizacije svakog signala). Kodira li se stereo signal kao jedan združeni signal, a zatim se taj signal prezentira kao stereo, može doći do pojave *BMLD* efekta – određeni šum signali koji su u mono verziji bili maskirani mogu postati čujni kada se prezentiraju kao razdvojeni, stereo signal.

8.2.3.3 RAAC izvedba *intenzitet-stereo* kodiranja

Dekodiranje *intenzitet-stereo* signala standardizirano je na sljedeći način:

- lijevi kanal se dekodira direktno iz kodiranog toka,
- desni kanal se dekodira na osnovu lijevog kanala:

$$l = d * scale \quad (8.10)$$

$$scale = 0.5^{0.25 \times is_position} \quad (8.11)$$

Lijevi kanal i *is_position* se određuju na strani koda posebno za svaki pojas kvantizatora. Veličina *is_position* određuje percipiranu poziciju stereo slike signala, a proporcionalna je omjeru energija lijevog i desnog kanala.

Da bi se ispoštovale glavne ideje *intenzitet-stereo* kodiranja koje su objašnjene u ovom poglavlju, kodiranje u RAAC koderu je izvedeno na sljedeći način:

- za svaki pojas kvantizatora proračunava se energija lijevog kanala, desnog kanala i ukupna energija,
- *is_position*, veličina koja određuje percipiranu poziciju stereo signala, proračunava se na sljedeći način:

$$is_position = 4 \times \log_2 \sqrt{\left(\frac{E_l}{E_d} \right)} \quad (8.12)$$

- proračunava se konačna vrijednost uzoraka lijevog kanala kao zbroj početnog lijevog i desnog kanala, skalirana sa doprinosom svoje energije prema ukupnoj energiji stereo signala:

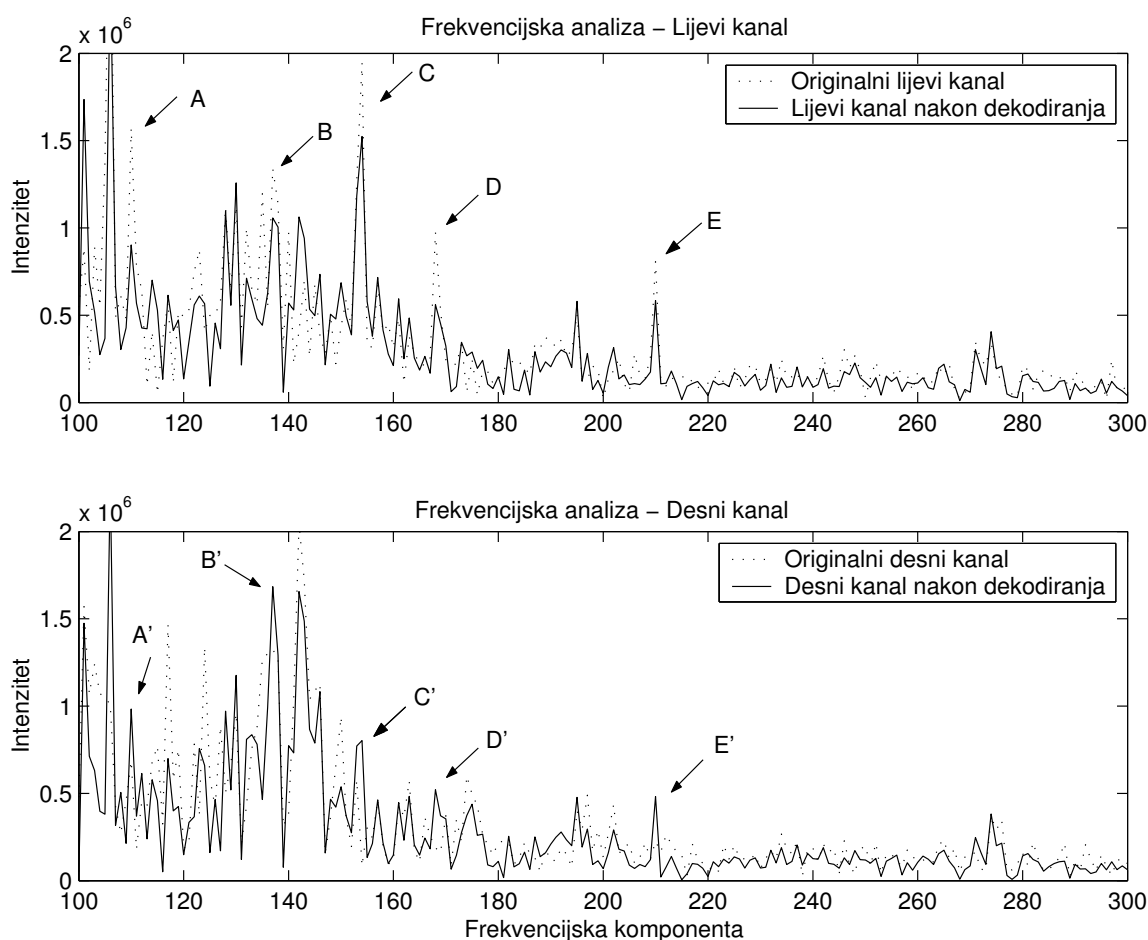
$$l' = (l + d) \times \sqrt{\frac{E_l}{E_{uk}}} \quad (8.13)$$

- u kodiranom toku se prenose sve komponente l' signala, zajedno sa $is_position$ vrijednostima za svaki pojas kvantizatora posebno.

Slika 8.7 prikazuje primjer nepoželjne upotrebe *intenzitet-stereo* kodiranja na niskim frekvencijama (počevši od 1 kHz). Izabran je primjer *pljeskanja* zbog činjenice da su lijevi i desni kanal stereo signala jako nekorelirani, sastoje se od niza nezavisnih audio događaja, pa ih to čini jako nepogodnim za *intenzitet-stereo* kodiranje.

Sa slike je vidljivo sljedeće: većina audio događaja koja se pojavljivala u samo jednom kanalu, sada je preslikana na drugi kanal – javio se tzv. *efekt preslušavanja*. Pošto je u ovom slučaju *intenzitet-stereo* kodiranje upotrijebljeno na niskim frekvencijama, uhu je onemogućena lokalizacija signala. Stereo slika signala je izgubljena, pa se kodirani signal čini kao dual-mono.

Slika 8.7: Primjer *intenzitet-stereo* kodiranja na niskim frekvencijama. Signal je jako nekoreliran, originalni lijevi i desni kanal imaju vrlo malo sličnosti. Nakon IS kodiranja/dekodiranja dobiven je signal sa jako umanjenom stereo slikom. Većina audio događaja (kao primjer su označeni A, B, C, D i E) sa jednog kanala se preslikala na drugi kanal (A', B', C', D' i E') pa se dekodirani signal čini kao da je dual-mono. Stereo doživljaj je izgubljen.



8.2.3.4 *Intenzitet-stereo* i čisti tonalni signali u RAAC koderu

Poznato je da *intenzitet-stereo* kodiranje dodatno narušava kvalitetu jako tonalnih signala. Kako se jako tonalni signali lakše percipiraju, tako je uho osjetljivije i na njihovu distorziranost.

Jedna od mogućnosti rješenja ovog problema, koje je izvedeno u RAAC koderu, je da se nakon detekcije tonaliteta pojedinih dijelova spektra signala, *intenzitet-stereo* kodiranje ne primjenjuje na izraženo tonalne dijelove spektra. Manjkavost ove metode je opasnost od prečestog mijenjanja između običnog i *intenzitet-stereo* kodiranja, što može uvesti dodatni

šum u signal. Decimacija bliskih tonalnih komponenata bi dovela do relativnog grupiranja tonalnih komponenti signala što bi pomoglo u smanjenju ovog efekta.

Unutar RAAC koder *intenzitet-stereo* kodiranje se upotrebljava na frekvencijama iznad 8 kHz do iznad 10 kHz, a prag tonaliteta iznad kojeg se *intenzitet-stereo* kodiranje više ne upotrebljava je eksperimentalno utvrđen i iznosi od 0.15 do 0.25. Ako je tonalitet bilo kojeg od dva kanala iznad ove granice, za spomenuti pojas se koristi standardno *lijevi/desni* kodiranje.

8.3 *Percepcijska zamjena šuma*

8.3.1 *Općenito*

Percepcijska zamjena šuma (engl. *Perceptual Noise Substitution – PNS*) algoritam [40] [101] se ne zasniva na standardnim psihoakustičkim principima koji se baziraju na efektima maskiranja. Posebnost PNS algoritma se ogleda u njegovom iskorištavanju neosjetljivosti ljudskog slušnog sistema na finu strukturu šum signala. Prije pojave PNS algoritma, cijeli signal se uvijek kodirao na isti način, sa pretpostavkom da je značajan za ukupnu percepciju. Tako su čak i najkaotičniji dijelovi signala, kao što je bijeli šum, kodirani na isti način kao i za percepciju najbitniji tonalni signali.

PNS algoritam uvodi novi način kodiranja izrazito netonalnih, šum signala [40], [101]. Umjesto da se nastoji zadržati valni oblik šum signala, pretpostavlja se da je za percepciju šuma bitna samo njegova snaga (kvadrat intenziteta). Tako se za šum signal pamti samo skupna energija unutar određenog frekvencijskog pojasa. Na strani dekodera se generatorom pseudo-slučajnih brojeva generiraju pojedine komponente na način da se unutar pojasa zadrži energija jednaka originalnom signalu, dok se fina struktura signala unutar pojasa gubi. Kako međutim, ljudski slušni sistem ne prepoznaje finu strukturu šum signala, ne gubi se na kvaliteti dekodiranog signala, dok se u isto vrijeme štedi u bitskom protoku jer se za šum-pojas ne prenose pojedine frekvencijske komponente nego se prenosi samo njihova ukupna energija.

Princip PNS kodiranja je dakle sljedeći:

- detektiraju se izrazito šumni pojasevi spektra analiziranog dijela vremenskog signala (šum-pojasevi ne moraju biti međusobno povezani, nego mogu biti ispresijecani sa tonalnim dijelovima spektra),

- za svaki šum-pojas se postavlja PNS zastavica koja dekoderu signalizira da se pojas ne dekodira standardnom metodom nego da je to PNS pojas,
- unutar šum-pojasa zbraja se energija svake komponente, dobiva se ukupna energija svakog pojasa koja se prenosi u kodiranom toku.

Kako je sam algoritam prilično jednostavan, kako za shvaćanje tako i za realizaciju, očito je da je jedini kritični dio cijele ideje o PNS kodiranju upravo sama detekcija šum-pojaseva određenih signala. Na koji način ocijeniti da je određeni pojas netonalan (šuman) pa ga zato kodirati PNS algoritmom i u isto vrijeme biti siguran da taj dio signala ne utječe bitno na sveukupnu percepciju kompleksnog signala koji se kodira?

8.3.2 RAAC izvedba PNS kodiranja

PNS dekodiranje je standardom definirano na sljedeći način:

- za pojas signala koji je označen kao PNS generira se niz slučajnih brojeva $rand[b]$ sa predznakom čija je srednja energija jednaka:

$$MEAN_NRG = \frac{2^{62}}{3} \quad (8.14)$$

- broj slučajnih brojeva je jednak broju frekvencijskih komponenti unutar pojasa,
- od slučajnih brojeva se dobivaju frekvencijske komponente sljedećom formulom:

$$frek[b] = rand[b] \times \frac{1}{broj_komponenti \times \sqrt{MEAN_NRG}} \times 2^{\frac{1}{4} \times noise_nrg[pojas]} \quad (8.15)$$

Na osnovu zadane formule za dekodiranje, kodiranje PNS pojaseva se vrši na sljedeći način:

- zbroje se kvadrati intenziteta svih frekvencijskih komponenti unutar PNS pojasa da bi se dobila ukupna energija pojasa;
- izračunata energija se preslikava na logaritamsku skalu sa korakom od 1.5 dB:

$$noise_nrg = 4 \times \log_2(Energ[pojas]) \quad (8.16)$$

- vrijednost $noise_nrg$ se prenosi u kodiranom toku.

Treba napomenuti da se u RAAC koderu detekcija tonaliteta svakog pojasa provodi unutar psihoakustičkog modula (Poglavlje 7.5.5). Za detaljniji pregled algoritama za detekciju tonaliteta signala pogledati Poglavlje 4.2.2.3.2.

8.3.3 Problemi pri upotrebi PNS kodiranja

8.3.3.1 Niske frekvencije

Kako je rezolucija frekvencijske analize signala frekvencije uzorkovanja od npr. 44.1 kHz (DFT u 1024 točke) otprilike 40 Hz, a kako je rezolucija ljudskog slušnog sistema na niskim frekvencijama jako velika, jasno je da su niske frekvencije nepogodne za upotrebu PNS algoritma. Također, na niskim frekvencijama je bitna i faza signala, koja se gubi pri PNS kodiranju, pa i iz tog razloga PNS kodiranje na niskim frekvencijama nije poželjno. Preporučuje se upotreba PNS kodiranja na frekvencijama višim od 4 do 6 kHz. RAAC koder upotrebljava PNS na frekvencijama iznad 5 kHz.

8.3.3.2 Pre-upotreba

Glavni problem koji se javlja kod PNS kodiranja je detekcija dijelova signala koje možemo nazvati šumnim. Sve dakle ovisi o algoritmu za detekciju tonaliteta signala.

Algoritam koji je izabran u RAAC izvedbi zasniva se na predviđanju na način da se predvidljiviji pojasevi signala smatraju tonalnijim dok se kaotičniji pojasevi signala smatraju više netonalnim. Kako spomenuti algoritam ne razvrstava signale na tonalne i netonalne (šumne), nego samo svakom pojasu pridjeljuje određenu vrijednost tonaliteta, postavlja se pitanje gdje povući granicu i na kojoj vrijednosti tonaliteta početi frekvencijske komponente proglašavati šumom.

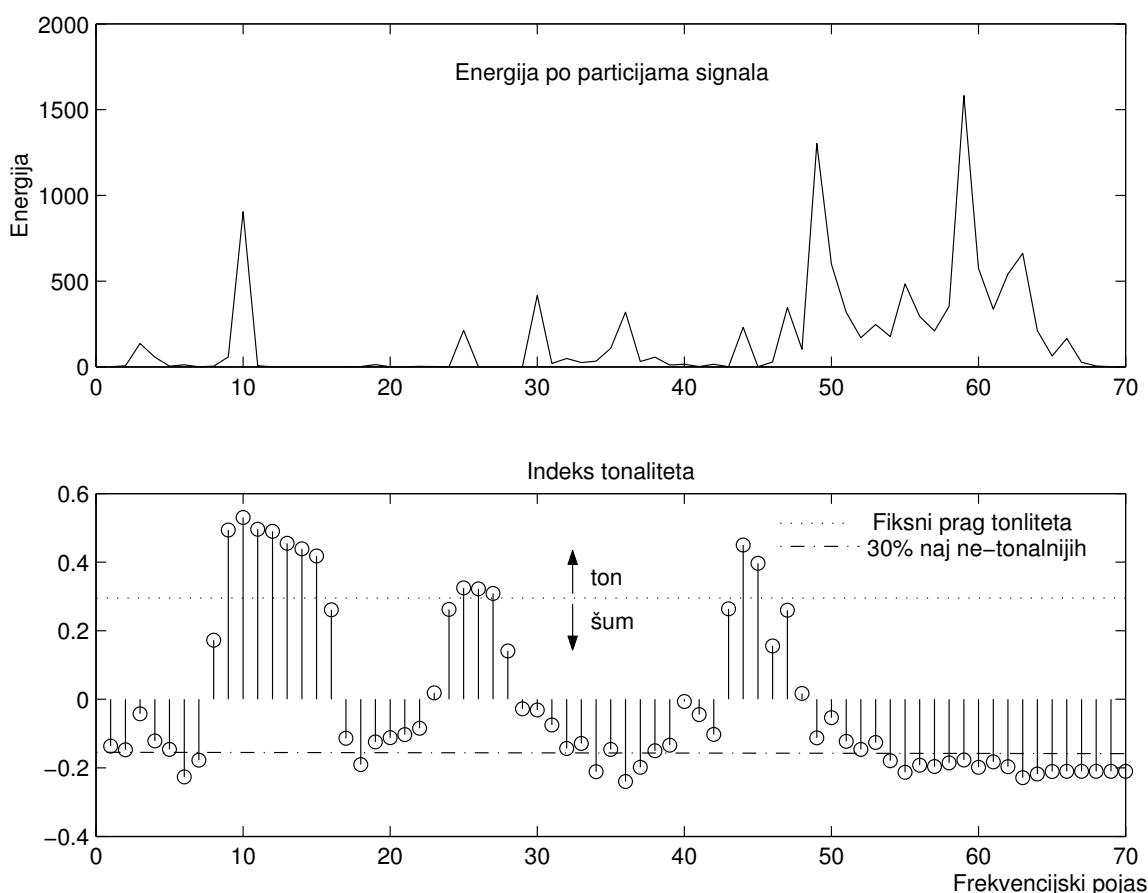
U prvim izvedbama RAAC koder uzimala se fiksna vrijednost tonaliteta od oko 0.3 do 0.5. Pojasevi koji su imali veći tonalitet smatrani su bitnima za percepciju, dok su pojasevi sa tonalitetom manjim od zadane granice smatrani šumnim. U praksi se ovakav pristup pokazao lošim. Kod većine signala, samo mali broj pojaseva je imao iole veći indeks tonaliteta. Pogotovo su problem bili tranzijentni dijelovi signala koji su sami po sebi nepredvidljivi, pa su u takvim slučajevima gotovo sve komponente signala imale nizak tonalitet. Upotrebom PNS algoritma, većina signala je sasvim gubila početnu strukturu, pa je kvaliteta dekodiranog signala bila nedopustivo niska.

Ovaj problem je u RAAC koderu riješen na sljedeći način: PNS kodiranje se ne primjenjuje na dijelove spektra koji imaju tonalitet ispod zadanog, nego se pojasevi signala poredaju

prema indeksu tonaliteta od najvećeg ka najmanjem, a PNS kodiranje se dozvoljava u samo određenom postotku naj-netonalnijih dijelova signala koji još uvijek imaju tonalitet ispod zadanog. Na ovaj način je spriječena *pre-upotreba* PNS kodiranja, a tamo gdje se upotrebljava PNS će zasigurno biti efikasan. Obično se PNS upotreba limitira na oko 20% naj-netonalnijih pojaseva.

Slika 8.8 prikazuje primjer limitiranja upotrebe PNS kodiranja.

Slika 8.8: Limitiranje upotrebe PNS kodiranja. Fiksni prag tonaliteta je označavao previše frekvencijskih pojaseva šumnim pa je upotreba PNS kodiranja značajno narušavala kvalitetu signala. Limitiranjem PNS kodiranja na određeni broj naj-netonalnijih komponenti koje su još uvijek ispod zadanog praga tonaliteta smanjilo je rizik od prevelike upotrebe PNS kodiranja.



8.3.3.3 PNS i čisti tonalni signali u RAAC koderu

Jedan od problema pri upotrebi PNS kodiranja javio se kod kodiranja jako zvukovno *čistih*, ali složenih audio signala. Bez obzira na limitiranje PNS upotrebe na određeni broj naj-

netonalnijih komponenti, kod audio uzoraka koji su sadržavali manji broj čistih, izraženih instrumenata, PNS kodiranje je ipak unosilo značajna pogoršanja u kvalitetu signala.

Ovaj problem je riješen pomoću proračuna percepcijske entropije signala. Percepcijska entropija, koja se računa u psihoakustičkom modulu za svaki blok signala (Poglavlje 7.5.8), na određeni način predstavlja smirenost, odnosno kaotičnost analiziranog bloka. Ukoliko je percepcijska entropija niska, zabranjuje se upotreba PNS kodiranja, jer niska percepcijska entropija znači čist signal. Na ovaj način je uveden je još jedan limit upotrebe PNS kodiranja. Eksperimentalno utvrđen iznos percepcijske entropije (prema Formuli 7.11) kod koje se PNS kodiranje dozvoljava u RAAC koderu je -300.

8.3.3.4 PNS i BMLD efekt

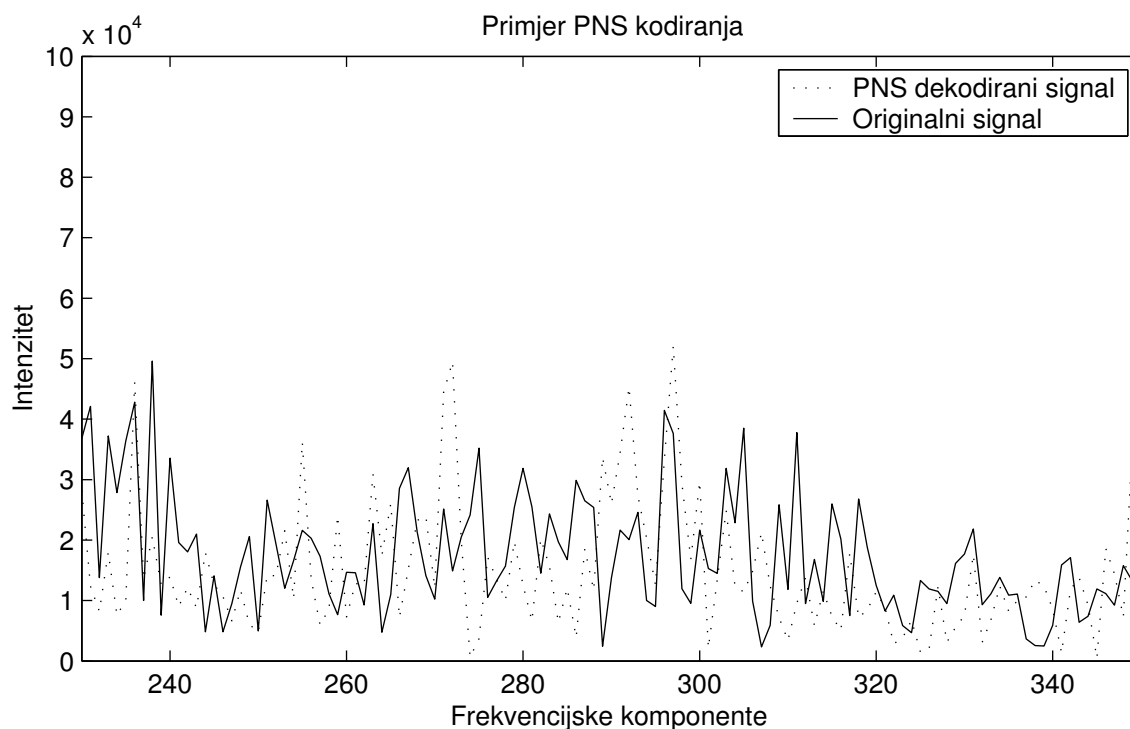
Dodatna pažnja mora biti posvećena kodiranju stereo signala kada se upotrebljava PNS. Poznato je da uvođenje različitog šuma u dva vrlo slična stereo kanala može dovesti do pojave *demaskirajućeg efekta* koji potječe od BMLD efekta (Poglavlje 2.9). Slična pojava se događa u slučajevima kada se oba kanala kodiraju kao PNS šum-signali. Ako se na strani dekodera generiraju različite šum-komponente, a dva originalna kanala su skoro jednaka, lako može doći do demaskirajućeg efekta kada šum postaje čujan.

Ova pojava se može riješiti unutar koda na način da se dekoderu može signalizirati da li je šum koreliran ili nije putem dodatne zastavice koja je u kodiranom toku standardom predviđena. U slučaju da je šum koreliran, jedne te iste slučajno-generirane šum komponente se ubacuju u oba kanala stereo signala, pa se izbjegava pojava demaskirajućeg efekta.

8.3.4 Primjer PNS kodiranja

Na Slici 8.9 je dan primjer pojasa frekvencija koje su kodirane PNS metodom. Iako se dekodirani signal na prvi pogled znatno razlikuje od originalnog, u percepciji dva signala razlika je prihvatljiva.

Slika 8.9: Primjer PNS kodiranja. Na prvi pogled, dekodirani signal se značajno razlikuje od originalnog. Međutim, kako je psihoakustičkom analizom pojas procijenjen šumnim, njegova fina struktura i nije bitna za ukupnu percepciju signala. U svakom pojasu kvantizatora ukupna energija signala je zadržana, dok je fina struktura signala nepovratno izgubljena.



8.3.5 Zaključak

Iako je PNS kodiranje jedna od novijih metoda audio kodiranja, ona je vrlo brzo usvojena kao dio MPEG-4 standarda. PNS kodiranje može uštediti u određenim slučajevima i do 20% bitskog protoka. Međutim, treba biti vrlo pažljiv u izboru dijelova signala u kojima će PNS biti upotrijebljen. RAAC koder ima nekoliko ograničenja upotrebe PNS algoritma:

- 1) tonalitet pojasa mora biti manji od 0.1,
- 2) maksimalan broj PNS pojaseva je ograničen na 30%,
- 3) entropija signala mora biti manja od -300.

Uz gornja ograničenja, RAAC koder sigurno iskorištava prednosti PNS algoritma, bez straha od degradacije kvalitete kodiranog signala.

Poglavlje 9

9 Ocjena RAAC koder

9.1 Uvod

MPEG-1 i MPEG-2 audio standardi revolucionizirali su primjenu audio kodiranja krajem devedesetih godina kroz pojavu *mp3* audio formata. Prihvatljiva kvaliteta signala je postignuta kod bitskog protoka 128 kb/s za stereo signal. Jedan od osnovnih zadataka MPEG-4 GA audio standarda je pomicanje granice od 128 kb/s koju je postigao prethodni *mp3* format. Da bismo utvrdili da li je ovaj cilj postignut RAAC izvedbom, potrebno je na određeni način evaluirati realizirani koder. Kao što ćemo vidjeti na sljedećim stranicama, evaluacija audio koder je vrlo kompleksna i predstavlja jedan od najzahtjevnijih dijelova razvoja audio koder.

Slika 9.1: Označeno područje bitskog protoka je od najvećeg interesa pri testiranju RAAC koder. Kako MPEG-1 nivo 3 (*mp3*) koderi postižu zadovoljavajuću kvalitetu signala na bitskim protocima iznad 128 kb/s za stereo signal, cilj je utvrđivanje na kojem bitskom protoku realizirani MPEG-4 GA koder postiže istu kvalitetu.



Ocjena kvalitete audio koder općenito se provodi na dva osnovna načina: subjektivnom analizom i objektivnom analizom.

9.2 Subjektivna analiza

Subjektivna analiza [19] [104] [105] [110] se provodi usporedbom kodiranih audio signala sa originalima provođenjem intenzivnih slušnih testova. Ova metoda je danas najprihvatljivija,

međutim i vrlo teško provediva. Zahtijeva veliki broj testova u koje moraju biti uključene glazbeno obrazovane i slušnim testovima trenirane osobe. Testiranje je sporo, ali su rezultati, iako vrlo raznoliki, ipak reprezentativni. Iako su testovi provedeni u istim uvjetima svaka osoba u rezultate testiranja unosi svoj subjektivni sud i osjećaj o kvaliteti signala - zato i naziv *subjektivno* testiranje.

Potaknute velikim problemima sa kojima se susreću pri provođenju subjektivnih audio testova, mnoge audio udruge su pokrenule procese standardiziranja metoda subjektivnih testiranja. Među njima prednjače ITU-R (engl. *International Telecommunication Union – Radiocommunication*), EBU (engl. *European Broadcasting Union*) i AES (engl. *Audio Engineering Society*).

Najrašireniji trenutno je ITU-R standard broj BS.1116. Ugrubo, standard definira način provođenja subjektivnih testiranja preporučujući:

- izbor audio uzoraka za testiranje (teški, laki, prosječni),
- karakteristike audio sistema (pojačala, zvučnika, slušalica, itd),
- akustičke karakteristike okoline u kojoj se testiranje vrši (sobna akustika i dr.),
- izbor individualnih slušatelja (stručnjaci i/ili laici),
- skalu za ocjenjivanje,
- metode analize rezultata.

Subjektivna evaluacija RAAC koda nije provedena sasvim sukladno BS.1116 iz više razloga. Međutim, testiranje se u velikom dijelu oslanjalo na preporuke iz spomenutog standarda. Kako RAAC testiranju i nije cilj utvrđivanje malih razlika u audio kvaliteti (što definira BS.1116), nego srednjih i velikih, izabrani način testiranja je posve zadovoljavajući.

RAAC subjektivna evaluacija ima sljedeće karakteristike:

- audio uzorci su prosječno teški i teški, a ne isključivo teški kao što standard preporučuje;
- metoda koja je upotrijebljena za testiranje je preporučena standardom *double-blind triple-stimulus with hidden reference* metoda;
- audio sistem na kojem se test vrši nije reprezentativan kao što standard traži, međutim obzirom na cilj testiranja raspoloživa audio konfiguracija je zadovoljavajuća;
- sobna akustika je zanemarena obzirom da se sva testiranja provode slušalicama (što je i osjetljivija metoda);

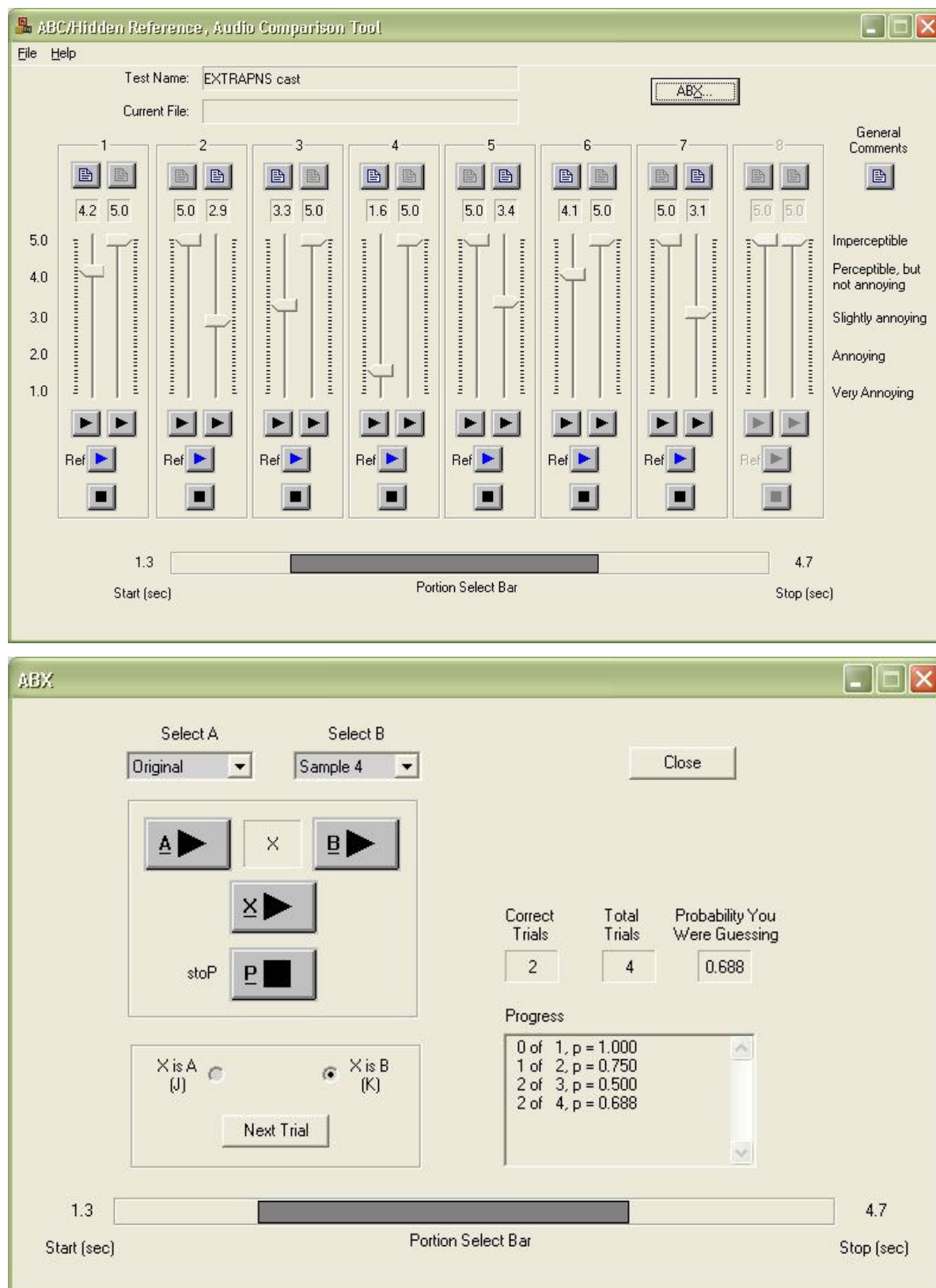
- izbor slušatelja koji su vršili testiranja je sasvim neprikladan standardu jer se cijeli test temelji na autorovu sudu;
- skala za ocjenjivanje je sasvim preuzeta iz standarda.

9.2.1 *Double-blind triple-stimulus with hidden reference test metoda*

Metoda koja je upotrijebljena za testiranje je preporučena *double-blind triple-stimulus with hidden reference* metoda. Ova metoda ima sljedeće karakteristike: ocjenjivač u svakom trenutku ima na raspolaganju tri uzorka: 'A', 'B' i 'C'. 'A' je referentni, originalni audio uzorak koji mu je uvijek na raspolaganju na poznatom mjestu. 'B' i 'C' su dva uzorka od kojih je uvijek jedan original istovjetan sa 'A', dok je drugi kodiran. Na ocjenjivaču je da ocijeni jedan od dva uzorka, 'B' ili 'C', sa ocjenom od 1.0 do 5.0. Neocijenjeni uzorak se onda smatra istovjetnim sa 'A', tj. sa originalom.

Pri testiranju RAAC koda korišten je besplatni *ABCHR* program prikazan na Slici 9.2.

Slika 9.2: *ABCHR* program za ocjenjivanje kodiranih audio signala metodom *double-blind triple-stimulus with hidden reference*. Program je dostupan besplatno na Internet stranici <http://www.fl23.org/>.



9.2.2 Izbor audio uzoraka

Audio uzorci su izabrani sukladno ideji da je cilj testa utvrđivanje srednjih i velikih razlika između kodiranog i originalnog signala. Uočavanje malih razlika nisu bile primarne jer je sam razvoj koder u početnoj ili najviše u srednjoj fazi. Dakle, iako standard predlaže izbor isključivo teških audio uzoraka koji bi *do krajnjih granica* testirali koder, za ovo testiranje su izabrani miješani uzorci sastavljeni i od uobičajenog, složenog audio materijala i od teških audio uzoraka.

Glavne karakteristike audio signala koje su bitne kod izbora testnih uzoraka su:

- tranzijentnost (zbog širenja šuma u vremenskoj domeni),
- tonalnost (zbog visoke osjetljivosti na šum i slabog maskiranja),
- govorni signal (zbog kritične kombinacije tonova i tranzijenata),
- složenost (zbog kombiniranog utjecaja svih vrsta signala),
- frekvencijski raspon (zbog gubitka visokih frekvencija i moduliranja visokofrekvencijskog šuma).

Vodeći računa da sve nabrojane karakteristike audio signala budu ravnopravno zastupljene, izabrano je 10 audio uzoraka koji su nabrojani u Tablici 9.1 dok su njihove glavne karakteristike naznačene u Tablici 9.2.

Frekvencija uzorkovanja svih odabranih audio uzoraka je 44.1 kHz, a dužina je od 6 do 13 sekundi.

Tablica 9.1: Opis audio uzoraka odabranih za testiranje RAAC koda.

	Ime datoteke	Opis
1.	<i>adam.wav</i>	Pop-rock, prosječno težak uzorak.
2.	<i>cast.wav</i>	Kastanje sa gitarom, tranzijentan, testiranje efekta pred-jeke.
3.	<i>gspe.wav</i>	Ksilofon (<i>glockenspiel</i>) instrument, izrazito tonalan, izražene visoke frekvencije.
4.	<i>harp.wav</i>	Čembalo (<i>harpsichord</i>) instrument, tonalan, složen signal, širok frekvencijski raspon.
5.	<i>horn.wav</i>	Truba, tonalan, uzak frekvencijski raspon, pretežno niže frekvencije.
6.	<i>moza.wav</i>	Mozart, klasična glazba, složen signal, bez izraženih visokih frekvencija.
7.	<i>pump.wav</i>	Rock, dosta teška kombinacija glasa i gitare, širokopojasan.
8.	<i>spfe.wav</i>	Govor, ženski, engleski jezik, kritični prijelazi kod slova <i>s</i> .
9.	<i>tehn.wav</i>	Polu-sintetička glazba, tranzijentan, složen signal, jako maskiranje.
10.	<i>zeph.wav</i>	Rock, kritičan visoko frekvencijski (<i>činele</i>) dio.

Tablica 9.2: Osnovne karakteristike audio uzoraka odabranih za testiranje kvalitete RAAC koda.

Ime datoteke		Karakteristike				
		složen	VF	tonal.	tranz.	govor
1.	<i>adam.wav</i>	+				
2.	<i>cast.wav</i>	+			+	
3.	<i>gspe.wav</i>		+	+	+	
4.	<i>harp.wav</i>	+		+	+	
5.	<i>horn.wav</i>			+		
6.	<i>moza.wav</i>	+				
7.	<i>pump.wav</i>	+	+			
8.	<i>spfe.wav</i>					+
9.	<i>tehn.wav</i>	+			+	
10.	<i>zeph.wav</i>	+	+		+	

9.2.3 Karakteristike audio sistema

Testiranja su provedena isključivo slušalicama, tako da sistem ne uključuje zvučnike, a sastoji se od sljedećeg:

- PC zvučna kartica Terratec EWS88 MT (24 bit D/A, 96 kHz, 108 dB dinamika);
- Sony STR-313L audio pojačalo;
- slušalice Beyer Dynamic (20 Hz – 20000 Hz, 200 Ohm).

9.2.4 Akustičke karakteristike okoline

Sobna akustika prilikom testiranja nije bila bitna jer je testiranje vršeno isključivo slušalicama.

9.2.5 Izbor individualnih slušatelja

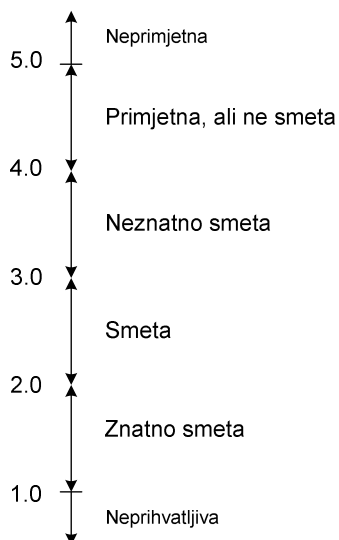
Izbor slušatelja koji će vršiti testiranja je sasvim neprikladan standardu ITU-R BS.1116 – cijeli test se temelji na ocjenama samog autora, dok su, većinom prosječni, netrenirani slušatelji uzeti u obzir samo kao korekcionni faktor. Ipak, autor je višemjesečnim radom stekao znatne sposobnosti u ocjenjivanju kvalitete kodiranih audio signala, pa je opravdano autorovu sudu dati i najveću težinu pri vrednovanju rezultata testiranja.

9.2.6 Skala za ocjenjivanje

Skala za ocjenjivanje kvalitete signala je preuzeta iz standarda ITU-R BS.1116. Prikazana je na Slici 9.3.

Skala se sastoji od četiri područja kvalitete signala koja se brojčano označavaju od 1 do 5. Također, pojedini cjelobrojni intervali imaju i opisne nazive. Tako se signal, za koji je nemoguće razlučiti razliku prema originalnom, označava kao "*neprimjetna razlika*", a ekvivalentna ocjena je 5.0. Ukoliko se razlika između originalnog i kodiranog signala ocijeni između 4.0 i 4.9 onda se razlika označava kao "*primjetna, ali ne smeta*" (engl. *perceptible but not annoying*). "*Neznatno smeta*" (engl. *slightly annoying*) razlika signala koji je dobio ocjenu između 3.0 i 3.9 dok "*smeta*" (engl. *annoying*) signal koji je ocijenjen u intervalu od 2.0 do 2.9. "*Znatno smeta*" (engl. *very annoying*) signal sa ocjenom od 1.0 do 1.9, a sve ispod 1.0 se ocjenjuje kao "*neprihvatljivo*".

Slika 9.3: Skala prema kojoj se ocjenjuje kvaliteta kodiranih audio signala. Brojčane vrijednosti od 1 do 5 imaju svoje ekvivalente u opisnim nazivima koji govore o načinu na koji je rangirana *razlika* kodiranog prema originalnom signalu.



9.2.7 Konfiguracije koda

Testiranje koda je provedeno upotrebom nekoliko konfiguracija koda koje na različite načine eksploatiraju pojedine njegove dijelove. Na ovaj način se najbolje vidi utjecaj pojedinih algoritama na specifične tipove signala.

Izabrano je pet konfiguracija koda koje na neki način objedinjuju sve pojedine algoritme uključene u RAAC koder. Izabrane konfiguracije su nazvane:

- 1) *Normal*
- 2) *Full*
- 3) *Basic*
- 4) *Extra PNS*
- 5) *Extra TNS*

9.2.7.1 *Normal*

Normal konfiguracija (Tablica 9.3) predstavlja tipičnu, osnovnu (engl. *default*) konfiguraciju koder. Obzirom da se ne upotrebljava PNS algoritam koji je i najproblematičniji obzirom na tip signala, ova konfiguracija bi trebala dati najstabilnije rezultate na širokom spektru signala.

9.2.7.2 *Full*

Full konfiguracija (Tablica 9.4) objedinjuje upotrebu svih algoritama izvedenih u koderu, pa stoga predstavlja i ono *najbolje* što bi koder mogao dati. Međutim, za pretpostaviti je da će ova konfiguracija kod određenih signala zakazati.

9.2.7.3 *Basic*

Basic konfiguracija (Tablica 9.5) predstavlja osnovnu ili minimalnu konfiguraciju koder. Korisna je za otkrivanje utjecaja pojedinih algoritama koder na kvalitetu kodiranog signala nakon što bi spomenuti algoritmi bili nadodani na *Basic* konfiguraciju, a kvaliteta signala onda bila uspoređena sa *Basic* konfiguracijom.

9.2.7.4 *Extra PNS*

Extra PNS (Tablica 9.6) se temelji na *Basic* konfiguraciji, a namijenjena je otkrivanju utjecaja PNS algoritma na različite tipove signala.

9.2.7.5 *Extra TNS*

Extra TNS (Tablica 9.7), slično kao i *Extra PNS* konfiguracija, temelji se na *Basic* konfiguraciji, a namijenjena je otkrivanju utjecaja TNS algoritma na različite tipove signala.

Tablica 9.3: Postavke koda za *Normal* konfiguraciju

Konfiguracija Normal	
Zaglavlje	adts
Niski propust	+
<i>Niski propust, frekvencija</i>	15 [kHz]
Intenzitet-stereo	+
<i>Intenzitet-stereo, frekvencija</i>	10 [kHz]
<i>Intenzitet-stereo, tonalitet</i>	0.3
Kratki blokovi	+
<i>Kratki blokovi, entropija</i>	-1000
TNS	-
<i>TNS frekvencija</i>	-
<i>TNS red filtra</i>	-
Zbroj/razlika (MS)	+
PNS	-
<i>PNS frekvencija</i>	-
<i>PNS tonalitet</i>	-
<i>PNS entropija</i>	-
<i>PNS postotak</i>	-

Tablica 9.4: Postavke koda za *Full* konfiguraciju

Konfiguracija Full	
Zaglavlje	adts
Niski propust	+
<i>Niski propust, frekvencija</i>	15 [kHz]
Intenzitet-stereo	+
<i>Intenzitet-stereo, frekvencija</i>	10 [kHz]
<i>Intenzitet-stereo, tonalitet</i>	0.3
Kratki blokovi	+
<i>Kratki blokovi, entropija</i>	-1000

TNS	+
<i>TNS frekvencija</i>	3 [kHz]
<i>TNS red filtra</i>	12
Zbroj/razlika	+
PNS	+
<i>PNS frekvencija</i>	5 [kHz]
<i>PNS tonalitet</i>	0.05
<i>PNS entropija</i>	-200
<i>PNS postotak</i>	35 %

Tablica 9.5: Postavke kodera za *Basic* konfiguraciju

Konfiguracija Basic	
Zaglavlje	adts
Niski propust	+
<i>Niski propust, frekvencija</i>	15 [kHz]
Intenzitet-stereo	-
<i>Intenzitet-stereo, frekvencija</i>	-
<i>Intenzitet-stereo, tonalitet</i>	-
Kratki blokovi	+
<i>Kratki blokovi, entropija</i>	-1000
TNS	-
<i>TNS frekvencija</i>	-
<i>TNS red filtra</i>	-
Zbroj/razlika	-
PNS	-
<i>PNS frekvencija</i>	-
<i>PNS tonalitet</i>	-
<i>PNS entropija</i>	-
<i>PNS postotak</i>	-

Tablica 9.6: Postavke koda za *Extra PNS* konfiguraciju

Konfiguracija Extra PNS	
Zaglavlje	adts
Niski propust	+
<i>Niski propust, frekvencija</i>	15 [kHz]
Intenzitet-stereo	-
<i>Intenzitet-stereo, frekvencija</i>	-
<i>Intenzitet-stereo, tonalitet</i>	-
Kratki blokovi	+
<i>Kratki blokovi, entropija</i>	-1000
TNS	-
<i>TNS frekvencija</i>	-
<i>TNS red filtra</i>	-
Zbroj/razlika	-
PNS	+
<i>PNS frekvencija</i>	5 [kHz]
<i>PNS tonalitet</i>	0.05
<i>PNS entropija</i>	-2000
<i>PNS postotak</i>	70 %

Tablica 9.7: Postavke koda za *Extra TNS* konfiguraciju

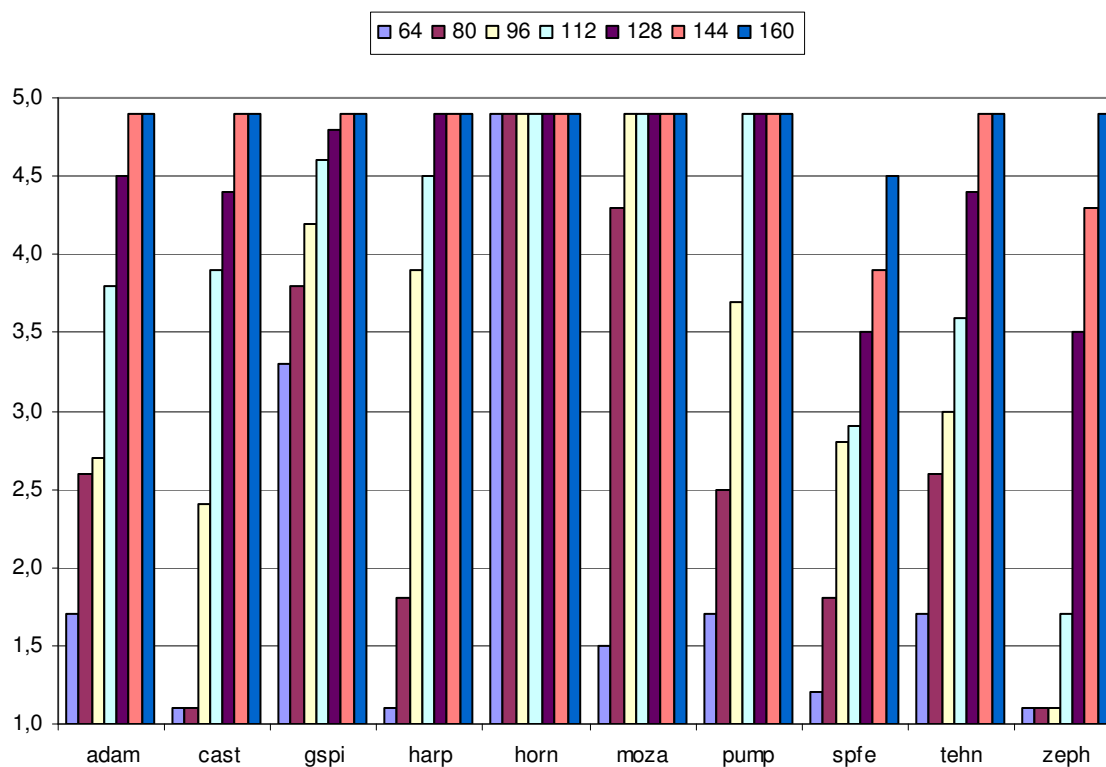
Konfiguracija Extra TNS	
Zaglavlje	adts
Niski propust	+
<i>Niski propust, frekvencija</i>	15 [kHz]
Intenzitet-stereo	-
<i>Intenzitet-stereo, frekvencija</i>	-
<i>Intenzitet-stereo, tonalitet</i>	-
Kratki blokovi	-
<i>Kratki blokovi, entropija</i>	-
TNS	+

<i>TNS frekvencija</i>	2 [kHz]
<i>TNS red filtra</i>	18
Zbroj/razlika	-
PNS	-
<i>PNS frekvencija</i>	-
<i>PNS tonalitet</i>	-
<i>PNS entropija</i>	-
<i>PNS postotak</i>	-

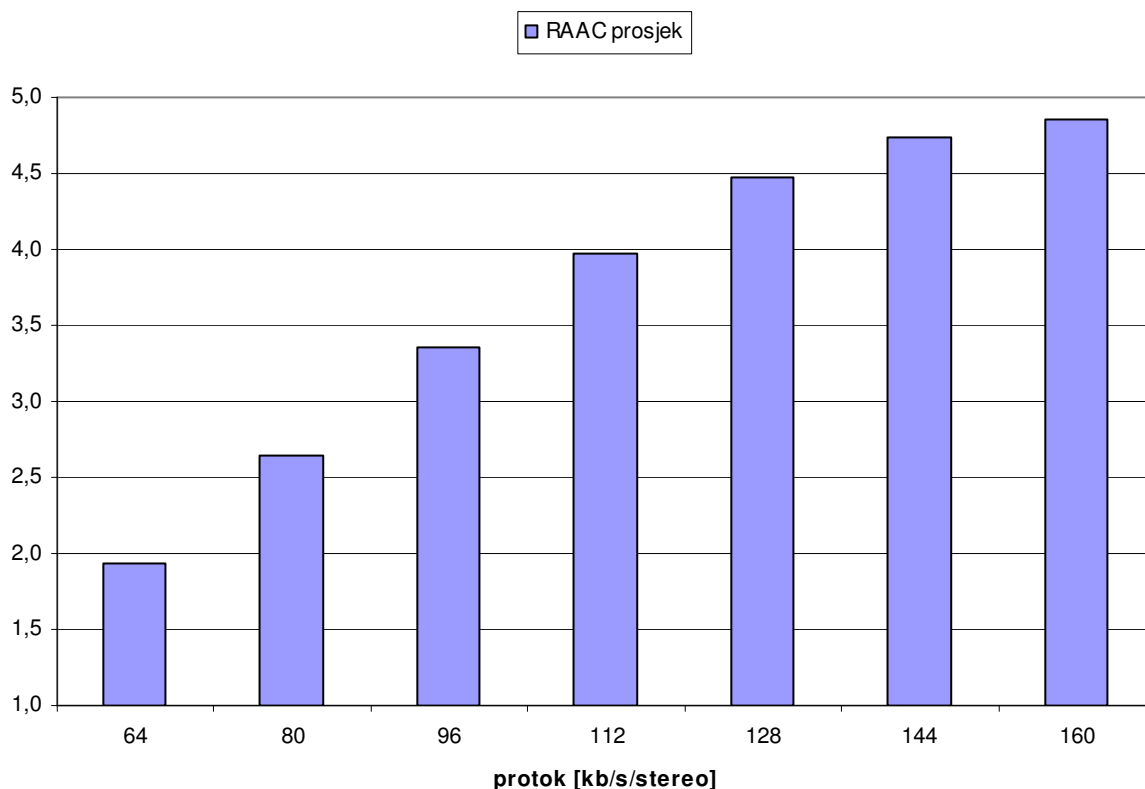
9.2.8 Rezultati testiranja

Sažetak rezultata testiranja svih 10 uzorka konfiguracijom *Normal* prikazan je na Slici 9.4. Prosječna ocjena prikazana je na Slici 9.5.

Slika 9.4: Rezultati subjektivnog testiranja konfiguracijom *Normal* na bitskim protocima 64, 80, 96, 112, 128, 144 i 160 kb/s stereo.



Slika 9.5: Prosječna ocjena subjektivnog testiranja konfiguracijom *Normal* na bitskim protocima 64, 80, 96, 112, 128, 144 i 160 kb/s stereo.



9.2.8.1 Ocjene pojedinačnih audio uzorka

Prikaz rezultata subjektivnog testiranja za svaki od 10 izabranih uzorka je dan pomoću dvije tablice i jedne slike.

Prva tablica sadrži ocjene za konfiguraciju *Normal* na 7 testiranih bitskih protoka: 64, 80, 96, 112, 128, 144 i 160 kb/s stereo.

Druga tablica sadrži informaciju o *prijelomnom* bitskom protoku (kod kojeg kvaliteta kodiranog signala počinje značajno i naglo padati) i ocjeni za konfiguraciju *Normal* na prijelomnom bitskom protoku. Također, druga tablica sadrži i usporedbu ostalih konfiguracija koda (*Full*, *Basic*, *Extra pns* i *Extra tns*) sa konfiguracijom *Normal*. Ukoliko je primijećeno da neka od ovih konfiguracija značajno utječe na kvalitetu signala u odnosu na konfiguraciju *Normal* to je i ocjenjeno – ako je ocjena pozitivna znači da je konfiguracija nadmašila *Normal* konfiguraciju, a ako je ocjena negativna znači da je konfiguracija lošija od konfiguracije *Normal*.

Na slici koja je dana nakon tablica uz svaki ocjenjeni uzorak prikazana je usporedba frekvencijskog odziva kodiranih signala na tri bitskog protoka (64, 96 i 128 kb/s) sa originalnim, iz čega se dobiva pregledan uvid u frekvencijsku karakteristiku kodiranih signala i eventualni gubitak određenih, pretežito VF frekvencija na nižim bitskim protocima (imati u vidu da je granična frekvencija RAAC koda uvijek 15 kHz).

9.2.8.1.1 Adam

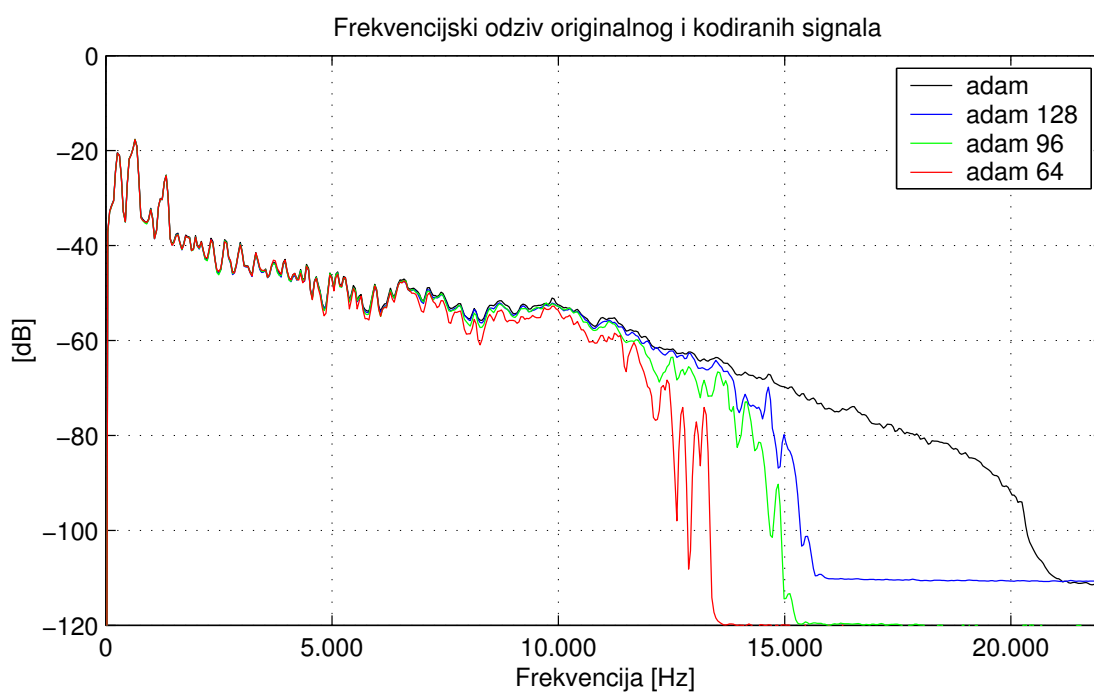
Tablica 9.8: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *adam*.

<i>adam</i>	
b. protok	ocjena
64	1,7
80	2,6
96	2,7
112	3,8
128	4,5
144	5,0
160	5,0

Tablica 9.9: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *adam*.

<i>adam</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
112	3,8	full	-
		basic	-
		extrapns	+0,5
		extratns	-

Slika 9.6: Usporedba frekvencijskog odziva originalnog signala *adam* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.2 *Cast*

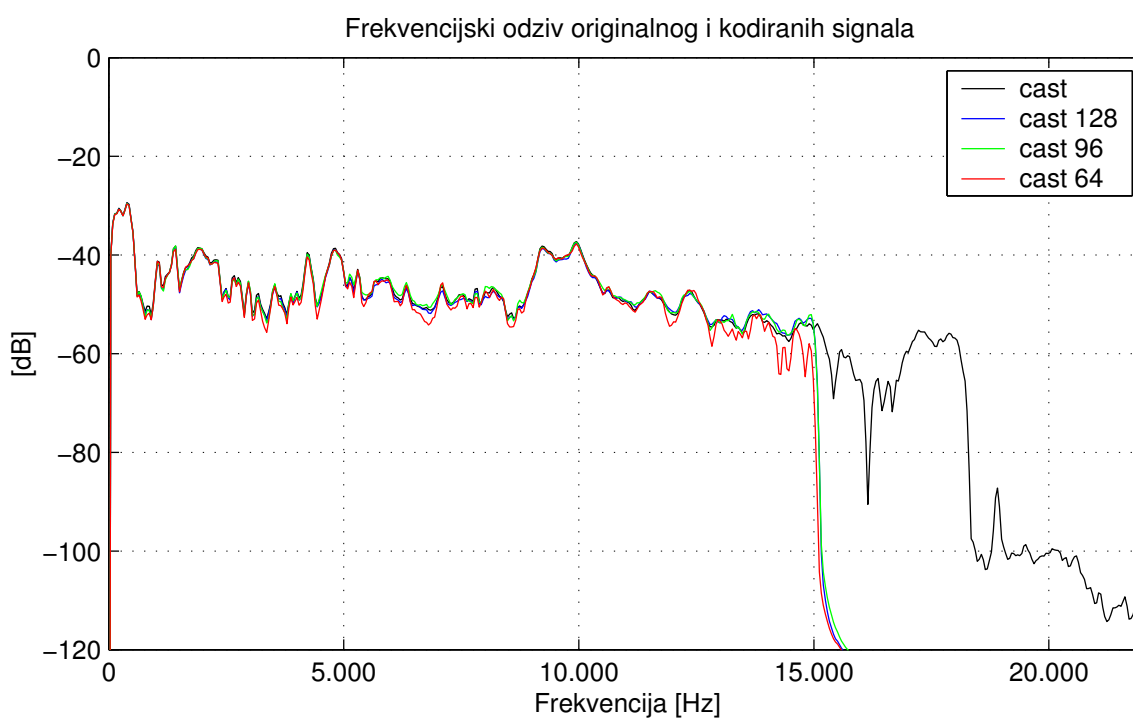
Tablica 9.10: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *cast*.

<i>cast</i>	
b. protok	ocjena
64	1,1
80	1,1
96	2,4
112	3,9
128	4,4
144	4,9
160	4,9

Tablica 9.11: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *cast*.

<i>cast</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
112	3,9	full	-
		basic	-
		extrapns	-0,6
		extratns	+0,5

Slika 9.7: Usporedba frekvencijskog odziva originalnog signala *cast* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.3 *Gspi*

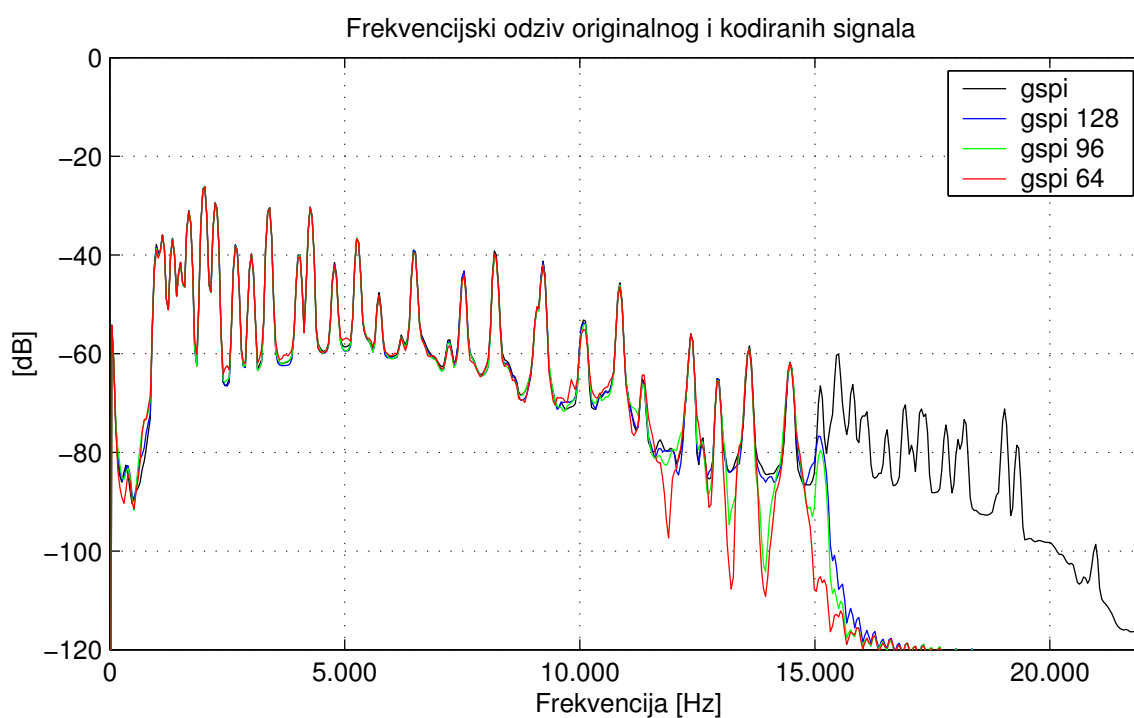
Tablica 9.12: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *gspi*.

<i>gspi</i>	
b. protok	ocjena
64	3,3
80	3,8
96	4,2
112	4,6
128	4,8
144	4,9
160	4,9

Tablica 9.13: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *gspi*.

<i>gspi</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
80	3,8	full	-
		basic	-
		extrapns	-
		extratns	+0,8

Slika 9.8: Usporedba frekvencijskog odziva originalnog signala *gspi* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.4 Harp

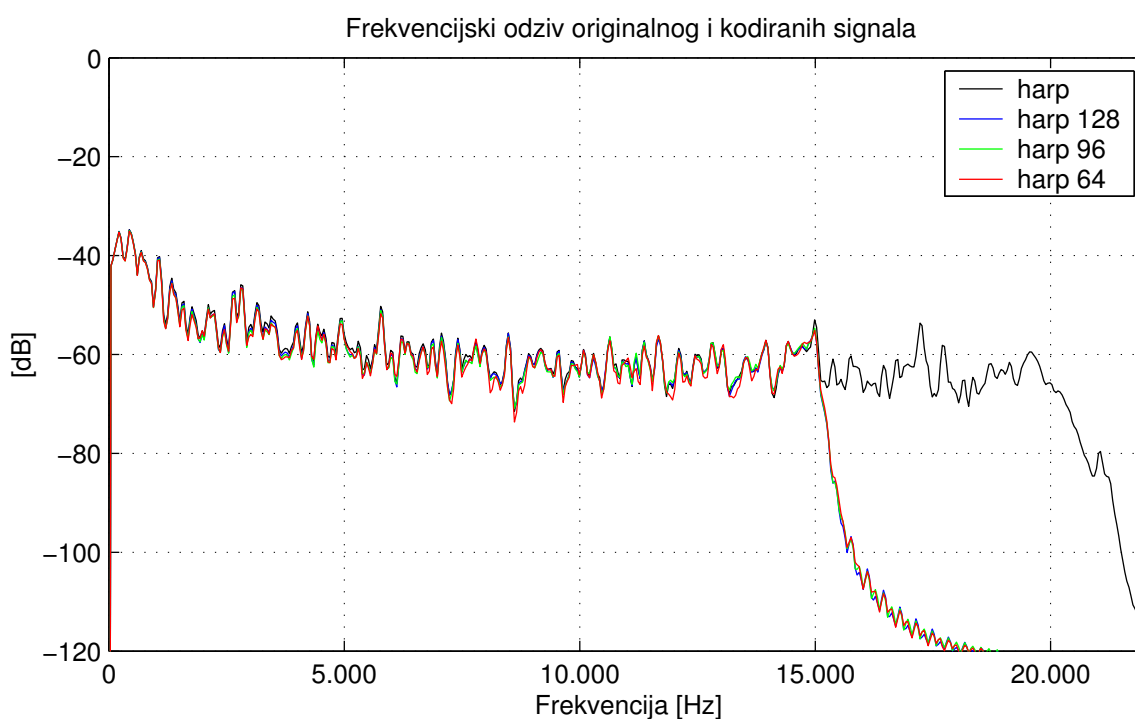
Tablica 9.14: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *harp*.

<i>harp</i>	
b. protok	ocjena
64	1,1
80	1,8
96	3,9
112	4,5
128	4,9
144	4,9
160	4,9

Tablica 9.15: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *harp*.

<i>harp</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
96	3,9	full	-
		basic	-
		extrapns	-
		extratns	-

Slika 9.9: Usporedba frekvencijskog odziva originalnog signala *harp* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.5 *Horn*

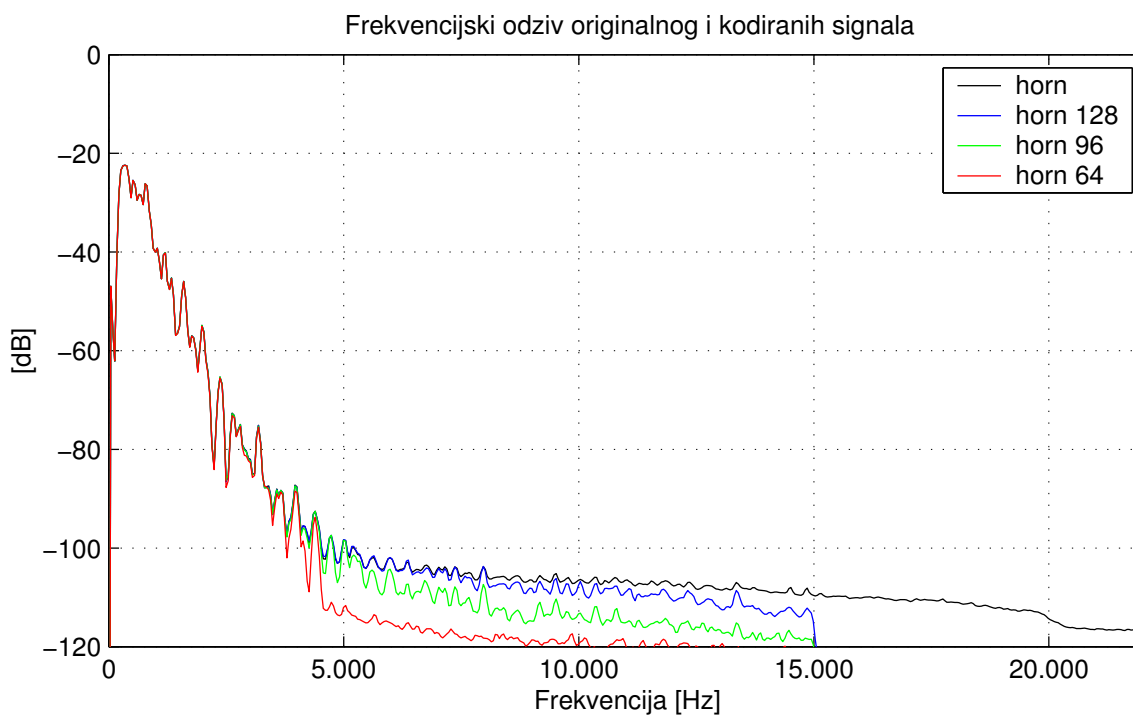
Tablica 9.16: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *horn*.

<i>horn</i>	
b. protok	ocjena
64	4,9
80	4,9
96	4,9
112	4,9
128	4,9
144	4,9
160	4,9

Tablica 9.17: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *horn*.

<i>horn</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
-	(4,9)	full	-
		basic	-
		extrapns	-
		extratns	-

Slika 9.10: Usporedba frekvencijskog odziva originalnog signala *horn* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.6 Moza

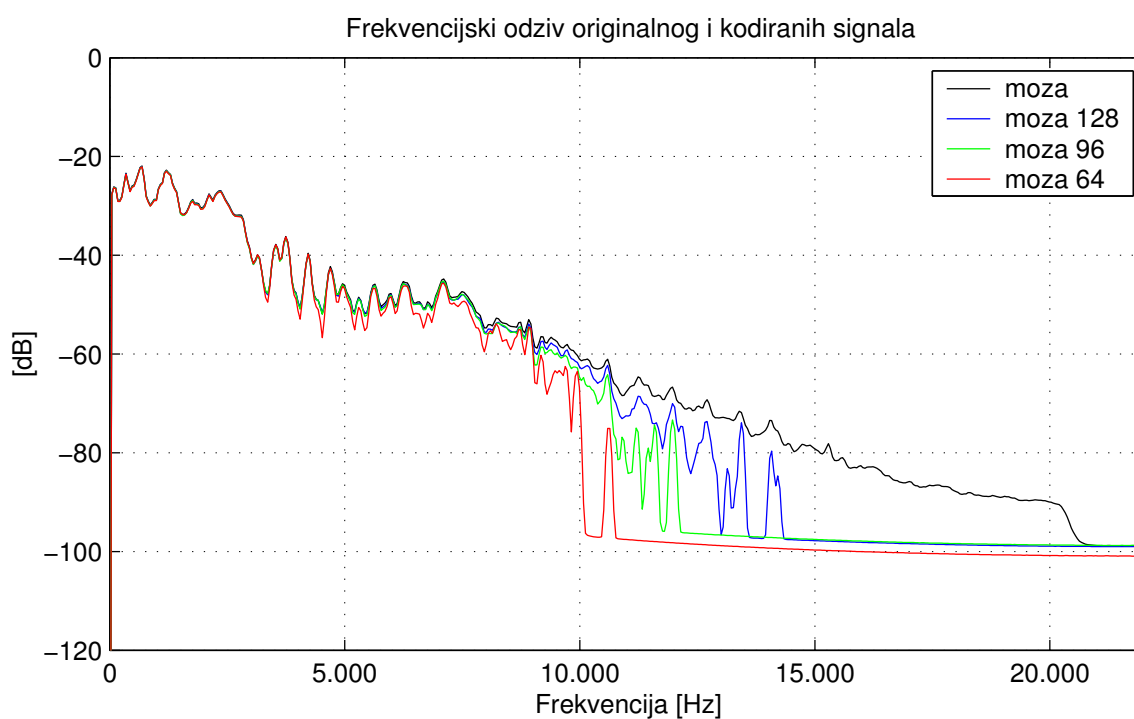
Tablica 9.18: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *moza*.

<i>moza</i>	
b. protok	ocjena
64	1,5
80	4,3
96	4,9
112	4,9
128	4,9
144	4,9
160	4,9

Tablica 9.19: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *moza*.

<i>moza</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
80	4,3	full	-
		basic	-
		extrapns	+0,8
		extratns	-

Slika 9.11: Usporedba frekvencijskog odziva originalnog signala *moza* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.7 Pump

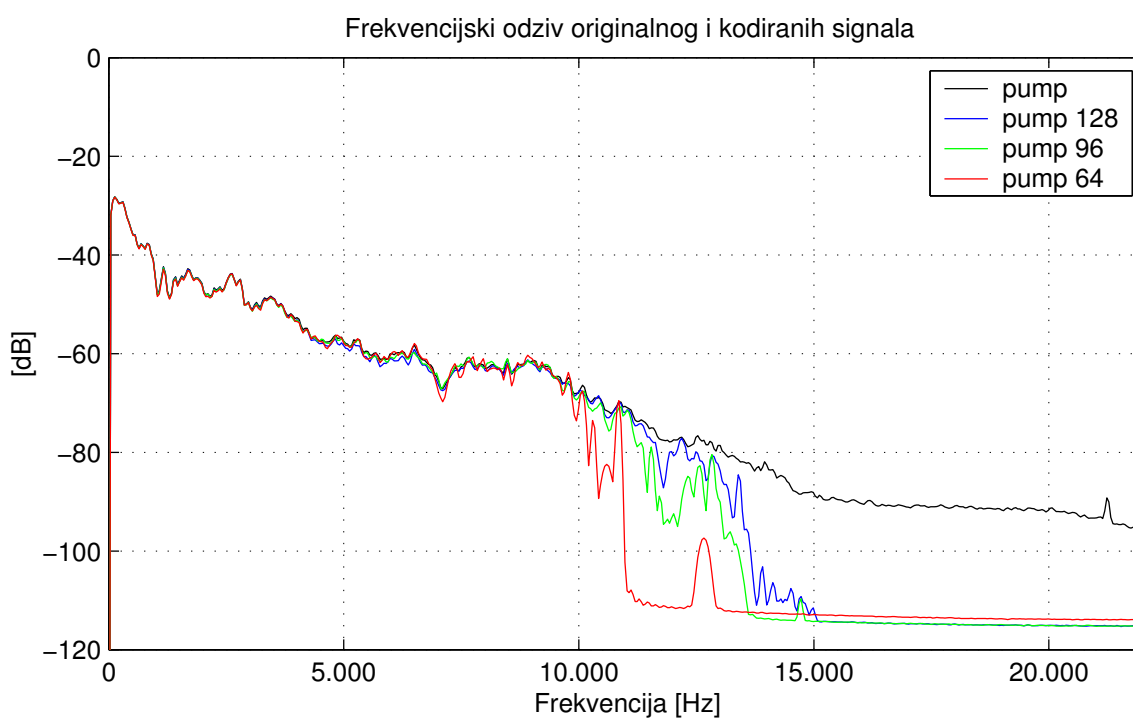
Tablica 9.20: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *pump*.

<i>pump</i>	
b. protok	ocjena
64	1,7
80	2,5
96	3,7
112	4,9
128	4,9
144	4,9
160	4,9

Tablica 9.21: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *pump*.

<i>pump</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
112	4,9	full	-
		basic	-
		extrapns	-
		extratns	-

Slika 9.12: Usporedba frekvencijskog odziva originalnog signala *pump* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.8 *Spfe*

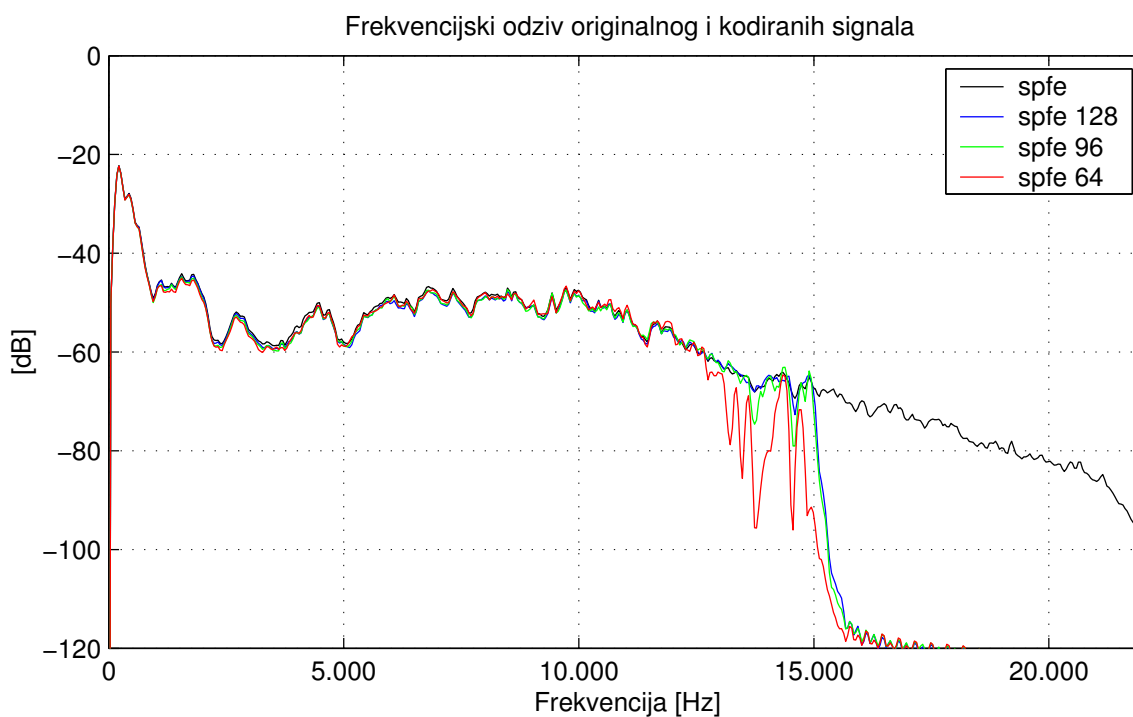
Tablica 9.22: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *spfe*.

<i>spfe</i>	
b. protok	ocjena
64	1,2
80	1,8
96	2,8
112	2,9
128	3,5
144	3,9
160	4,5

Tablica 9.23: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *spfe*.

<i>spfe</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
144	3,9	full	-
		basic	-
		extrapns	+1,2
		extratns	+0,5

Slika 9.13: Usporedba frekvencijskog odziva originalnog signala *spfe* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.9 Tehn

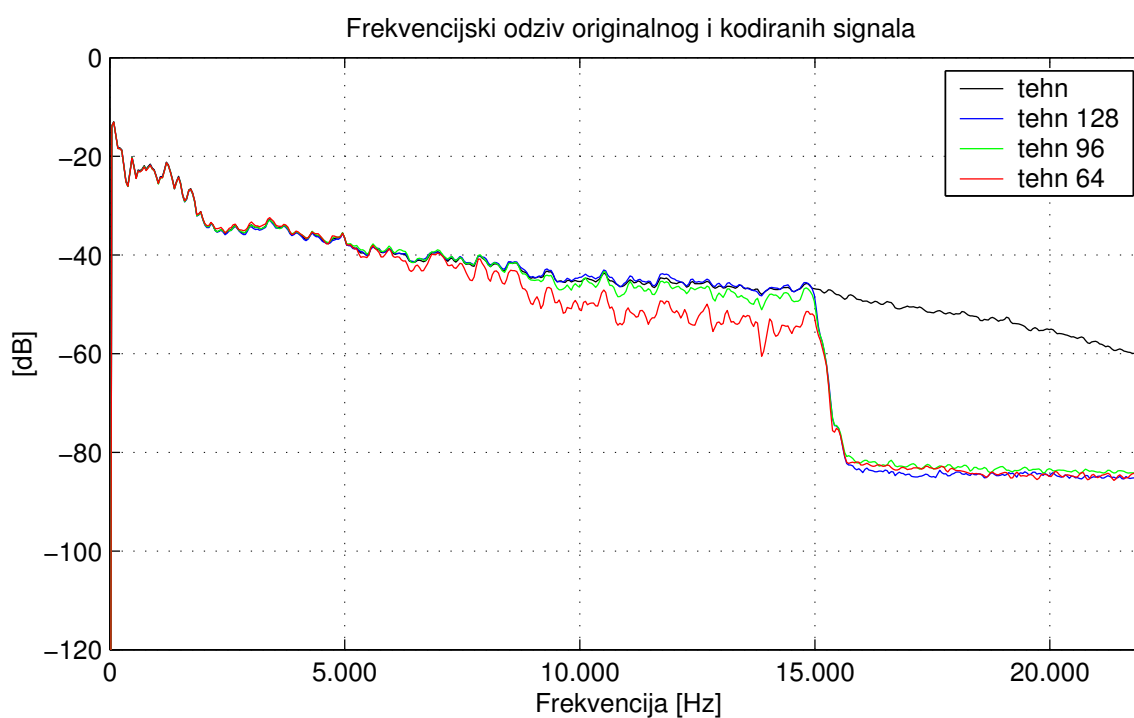
Tablica 9.24: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *tehn*.

<i>tehn</i>	
b. protok	ocjena
64	1,7
80	2,6
96	3,0
112	3,6
128	4,4
144	4,9
160	4,9

Tablica 9.25: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *tehn*.

<i>tehn</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
128	4,4	full	-
		basic	-
		extrapns	+0,4
		extratns	-

Slika 9.14: Usporedba frekvencijskog odziva originalnog signala *tehn* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.1.10 Zeph

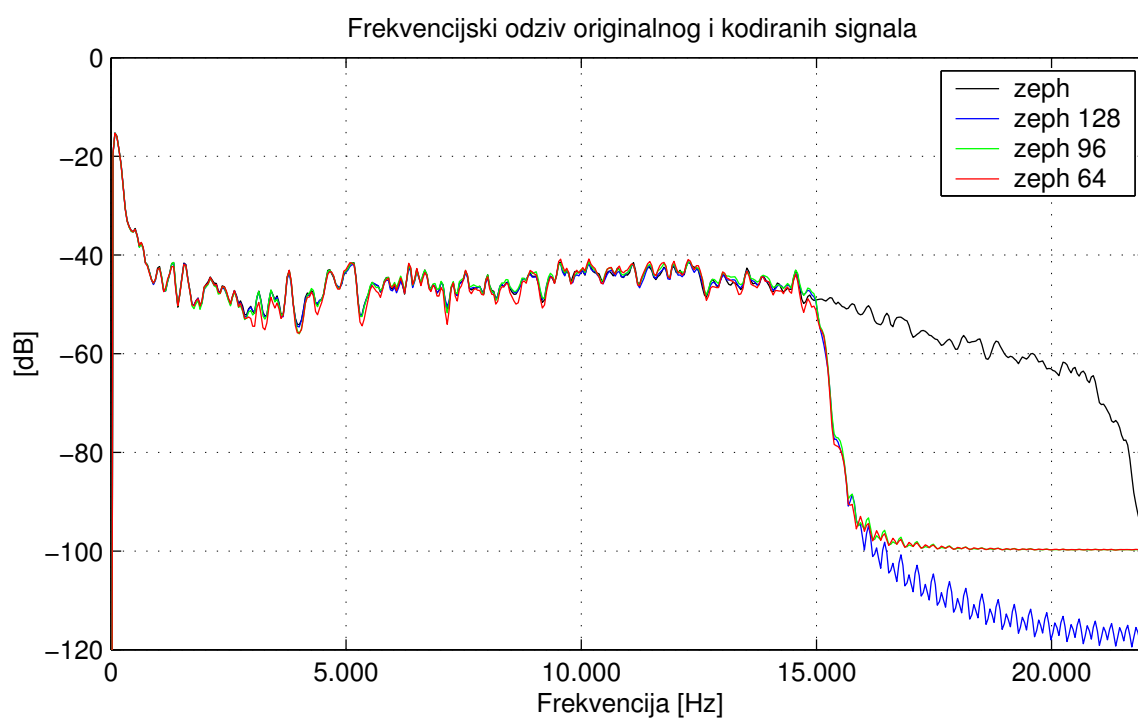
Tablica 9.26: Rezultat subjektivnog testiranja konfiguracije *Normal* za uzorak *zeph*.

<i>zeph</i>	
b. protok	ocjena
64	1,1
80	1,1
96	1,1
112	1,7
128	3,5
144	4,3
160	4,9

Tablica 9.27: Usporedba *Normal* konfiguracije sa ostalim konfiguracijama koda za uzorak *zeph*.

<i>zeph</i>			
prijelomni b. protok	ocjena normal	koder konfig.	relativno normal
144	4,3	full	-
		basic	-
		extrapns	+0,8
		extratns	-

Slika 9.15: Usporedba frekvencijskog odziva originalnog signala *zeph* sa kodiranim signalom na bitskim protocima 128, 96 i 64 kb/s.



9.2.8.2 Analiza uspješnosti

Nakon provedenih subjektivnih testova nametnula se podjela izabranih audio uzoraka u tri skupine, obzirom na uspješnost kodiranja sa RAAC koderom. Prvu skupinu čine uzorci koje je RAAC koder vrlo uspješno savladavao, drugu grupu čine uzorci srednje težine, dok se u treću grupu svrstani teži uzorci.

Koder postiže najbolje rezultate sa kodiranjem uzoraka *gspe*, *harp*, *horn*, *moza* i *pump*. Uzorak *horn*, iako tonalan, ima vrlo uzak frekvencijski raspon (oko 5 kHz), pa je daleko najlakši za kodiranje – razlika prema originalu je skoro *neprimjetna* čak na niskom bitskom protoku od 64 kb/s. Uzorak *gspe* je najtonalniji uzorak, ipak RAAC koder sa njim nema problema. Uzorci *harp*, *moza* i *pump* su složeni, ne izrazito tranzijentni signali, iako *harp* izraženo tonalan. Prosječnog se frekvencijskog raspona, a postižu nominalni bitski protok od 96 kb/s do 112 kb/s.

Srednje teški uzorci su *adam*, *cast* i *tehn*. Iako su *cast* i *tehn* natprosječno tranzijentni signali, nominalni bitski protok kod ovih uzoraka je od odličnih 112 kb/s do dobrih 128 kb/s.

Kao najteži uzorci su se pokazali *spfe* (govor) i *zeph* (jako izražene visoke frekvencije). Kod ovih uzoraka nominalni bitski protok je iznad 128 kb/s, ali je uz upotrebu PNS algoritma i on snižen na zadovoljavajuće vrijednosti.

9.2.8.3 Nominalni bitski protok

Koder u prosjeku postiže prihvatljivu kvalitetu ocijenjenu sa 3.5 i više (razlika '*primjetna*, *ali ne smeta*' ili '*neznatno smeta*') u velikoj većini testiranih uzoraka na bitskom protoku od 112 kb/s. Kod lakših uzoraka, kojih je pola od testiranih, ovaj, nazovimo ga nominalni bitski protok, još je i niži, i iznosi 96 kb/s.

9.2.8.4 Frekvencijski spektar

Frekvencijski spektar je jedna od ključnih karakteristika svakog audio signala kada je u pitanju audio kodiranje. Od izabranih uzoraka najizraženije visoke frekvencije ima *zeph* uzorak koji se i pokazao najteži za kodiranje. Jako izražene visoke frekvencije imaju i *gspe* i *harp*, ali oni nisu previše složeni signali pa RAAC sa njima nema problema. Najuži frekvencijski raspon ima *horn* uzorak, koji je dobio najbolje ocjene.

9.2.8.5 Detekcija tonalnosti

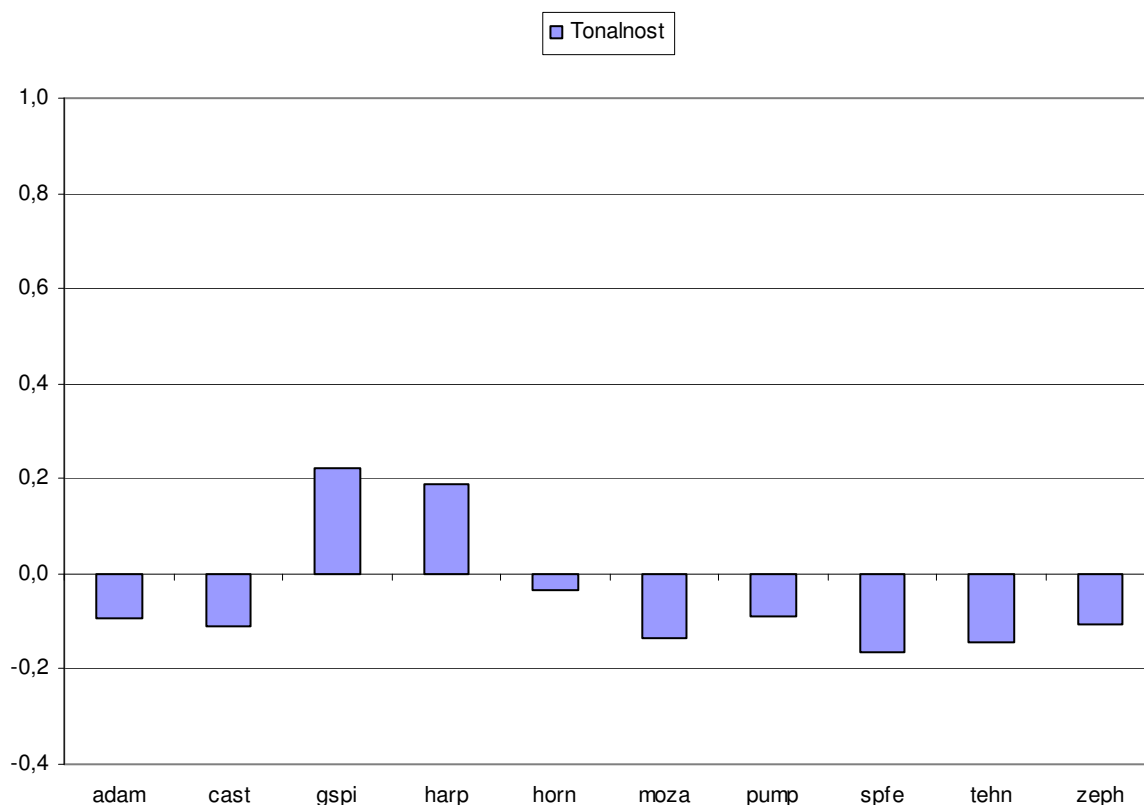
Na Slici 9.16 prikazan je prosječan iznos tonaliteta svakog testiranog audio uzorka, izračunat prema RAAC algoritmu za procjenu tonaliteta (Poglavlje 7.5.5).

Daleko najtonalniji su se pokazali uzorci *gspe* i *harp*, dok je donekle visok tonalitet pokazao i *horn* uzorak. Iz frekvencijskog prikaza na Slikama 9.8, 9.9 i 9.10 također je očito da su *gspe*, *harp* i *horn* izrazitije tonalni signali, pa zaključujemo da RAAC algoritam dobro ocjenjuje tonalitet raznovrsnih tipova signala.

Ostali uzorci ne pokazuju visok prosječni tonalitet, što ipak ne znači da njihovi određeni dijelovi nisu visoko tonalni (npr. *moza*).

Kod tonalnih uzoraka *harp*, *horn* i *gspe*, vrlo je osjetljiva upotreba PNS algoritma na bitskim protocima iznad nominalnog jer dolazi do očitog narušavanja kvalitete, pa je u RAAC koderu upotreba PNS algoritma dodatno ograničena obzirom na entropiju signala. Od upotrebe PNS algoritma najviše profitiraju uzorci *moza*, *spfe*, *tehn* i *zeph* koji su najviše netonalni, a imaju i relativno visoku entropiju.

Slika 9.16: Prosječna tonalnost testiranih audio uzoraka prema RAAC algoritmu. Tonalni uzorci su osjetljivi na PNS algoritam, dok netonalni uzorci značajno dižu kvalitetu upotrebom PNS algoritma.



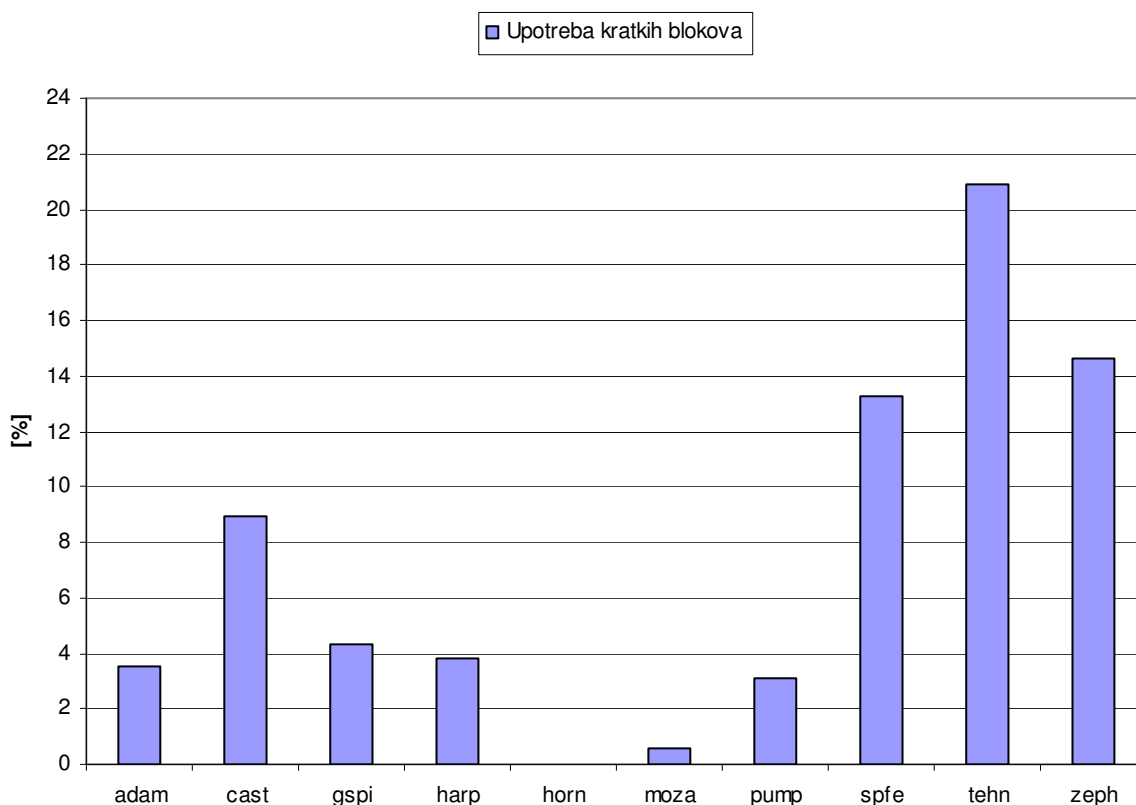
9.2.8.6 Detekcija tranzijentnosti

Tranzijentnost je jedna od bitnijih karakteristika signala u audio kodiranju, pa su i algoritmi za detekciju tranzijentnosti vrlo značajni. Postotak svakog uzorka detektiran kao tranzijent prema RAAC algoritmu za detekciju tranzijenta je prikazan na Slici 9.17.

Kao najtranzijentniji uzorak pokazao se *tehn* uzorak koji ima više od 20% kratkih blokova. Nakon njega u grupu jako tranzijentnih ubrajaju se *zeph* (izražene činele) i *spfe* govorni uzorak. Uzorak *cast* ima oko 9% kratkih blokova. Ostali uzorci su ocijenjeni više (*gspe*, *harp*), manje (*adam*, *pump*) ili sasvim (*horn*) stacionarni.

Ovakvi rezultati detekcije tranzijentnosti su u potpunosti u skladu sa očekivanjima, pa zaključujemo da je algoritam za detekciju tranzijenata u RAAC koderu vrlo pouzdan.

Slika 9.17: Postotak kratkih blokova detektiranih RAAC koderom pomoću porasta u energiji visoko-frekvencijskih komponenti signala i proračuna entropije signala.

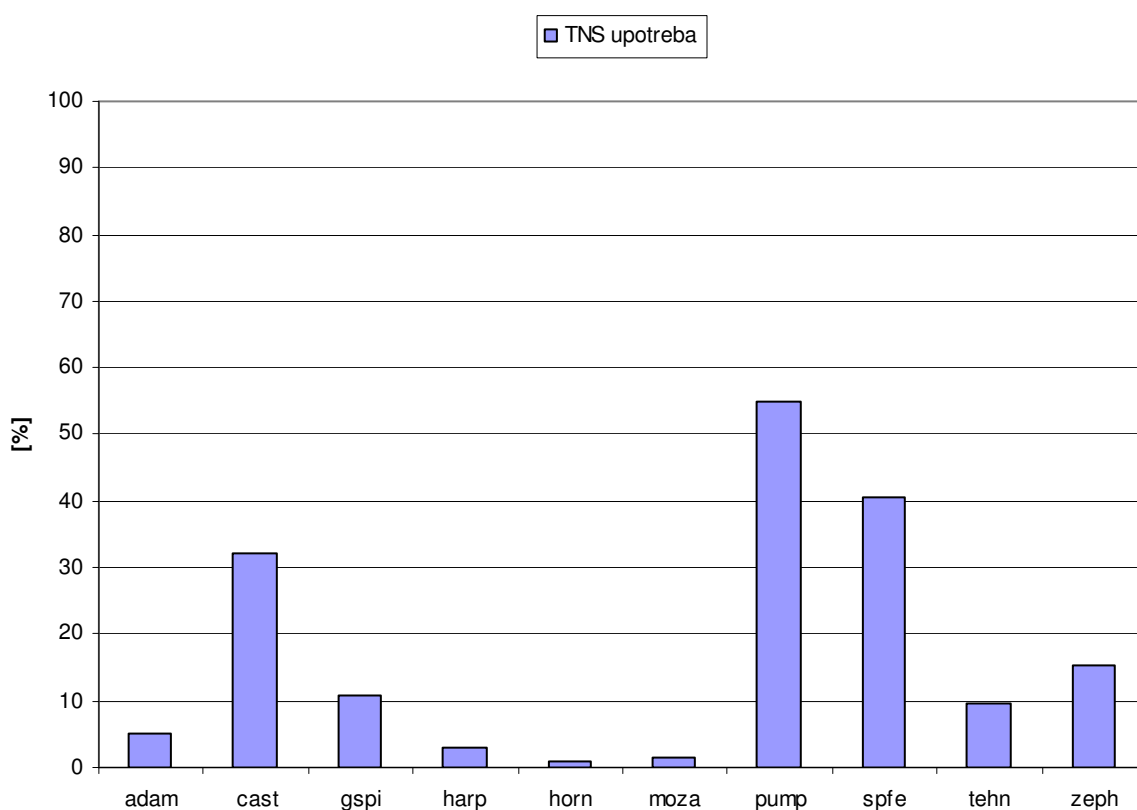


9.2.8.7 TNS upotreba

TNS algoritam za sprječavanje efekta pred-jeke najviše je korišten kod kodiranja uzoraka *pump*, *spfe*, *cast* i *zeph*. Kako tri od nabrojana četiri uzorka spadaju u najtranzijentnije uzorke,

može se reći da je algoritam uspješan u detekciji signala koji su tranzijentni i kod kojih je upotreba TNS algoritma potrebna. Kod *tehn* uzorka koji jeste izrazito tranzijentan, upotreba TNS algoritma nije tako intenzivna (~10%), međutim ovu pojavu možemo pripisati karakteristikama samog uzorka – *tehn* uzorak je vrlo specifičan obzirom da je većim dijelom kompjutorski sintetiziran. Ostalih 5 uzoraka imaju relativno rijetku upotrebu TNS algoritma. Intenzivna upotreba TNS filtriranja kod *pump* uzorka može se pripisati specifičnim osobinama samog audio uzorka.

Slika 9.18: Postotak upotrebe TNS algoritma kod RAAC koda.



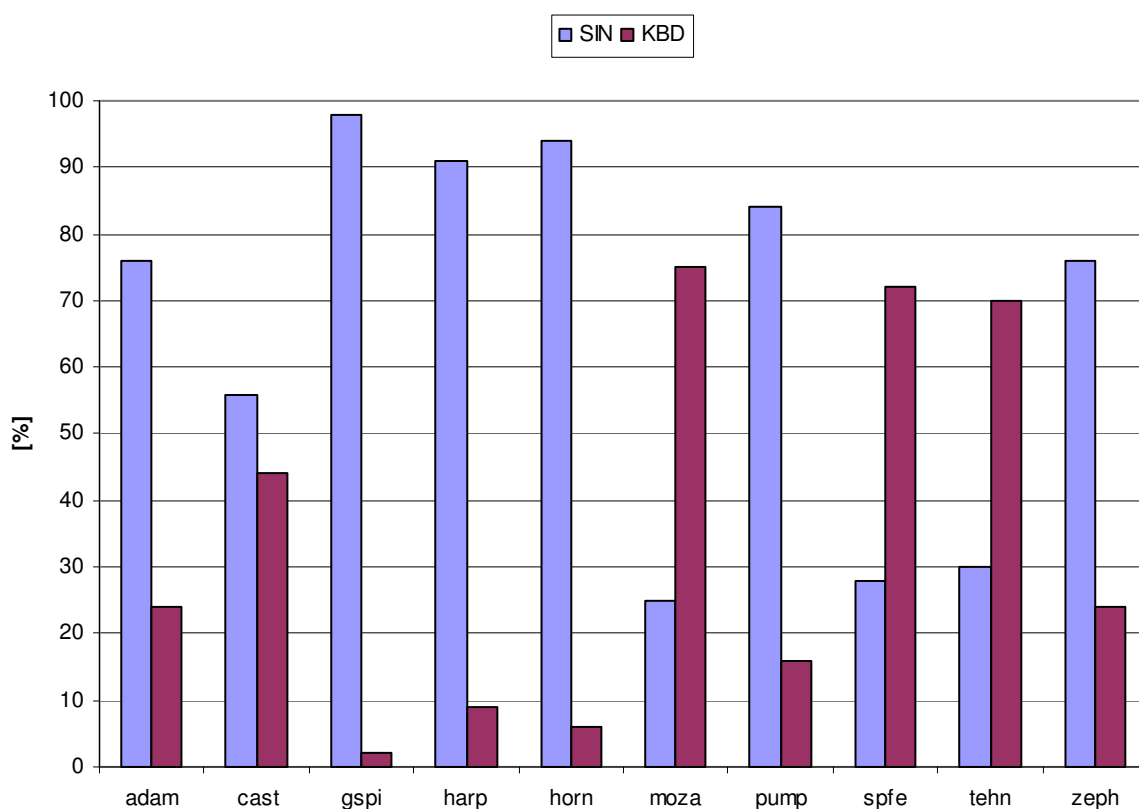
9.2.8.8 Izbor oblika prozora

Na Slici 9.19 prikazan je postotak izbora *Sinus* i *KBD* oblika prozora kod testiranih audio uzoraka. Kako se RAAC algoritam izbora oblika prozora bazira na proračunu zbijenosti tonalnih komponenti, očito je da kod jače tonalnih signala koji imaju zbijenije tonalne komponente, prevladava sinusni oblik prozora. Tri najtonalnija uzorka *gspi*, *harp* i *horn* imaju 98%, 91% i 94% sinus prozora. Tri naj-netonalnija uzorka *moza*, *spfe* i *tehn* od svih uzoraka

imaju najizraženiju upotrebu KBD prozora: 75%, 72% i 70%. Rezultati su u potpunosti u skladu sa očekivanjem.

Treba napomenuti da je prag tonaliteta kod kojeg su signali proglašavani tonalnim u ovom slučaju imao iznos 0.4, što je eksperimentalno potvrđeno kao prihvatljiva granica. Veći prag bi uzrokovao smanjenje tonalnih komponenti, a samim tim i smanjenje udjela sinus prozora. Ipak, predlaže se dodatno eksperimentiranje sa granicom tonaliteta.

Slika 9.19: Postotak upotrebe *sinus* i *KBD* prozora kod RAAC koda. Očito je da zbog bolje frekvencijske selektivnosti u tonalnim signalima prevladava upotreba *sinus* prozora. Iz obrnutog razloga kod kaotičnih signala prevladava *KBD* oblik prozora.

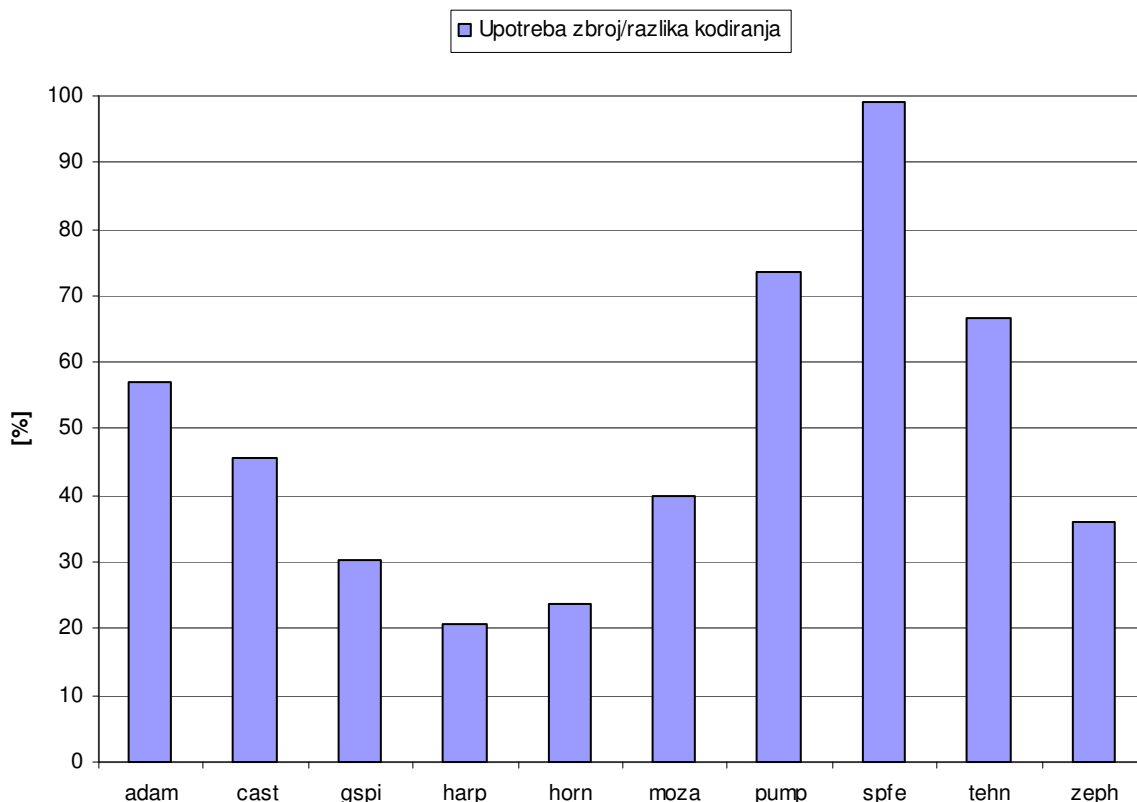


9.2.8.9 Upotreba *zbroy/razlika* kodiranja

Kako upotreba *zbroy/razlika* kodiranja ovisi samo o sličnosti dva kanala audio uzorka, razumljivo je da RAAC koder najviše upotrebljava *zbroy/razlika* kodiranje kod uzorka *spfe* jer je to govorni uzorak kod kojeg se jedan glas snimao istovremeno na oba kanala. Ostali audio uzorci su složeni pa ocjena *zbroy/razlika* algoritma nije jednostavna jer je sličnost dva kanala

prilično nepoznata, iako se preslušavanjem kanali čine vrlo sličnima kod svih uzoraka. Preostaje samo konstatacija da se *zbroy/razlika* kodiranje najviše upotrebljava kod uzoraka *pump*, *tehn* i *adam* i to u preko 50 % slučajeva. Slijede uzorci *cast*, *moza* i *zeph* sa 30 do 50 % *zbroy/razlika* pojaseva. Najmanji udio *zbroy/razlika* kodiranja imaju uzorci *gspe*, *harp* i *horn*, od 20 do 30%.

Slika 9.20: Postotak upotrebe *zbroy/razlika* kodiranja kod RAAC koder.



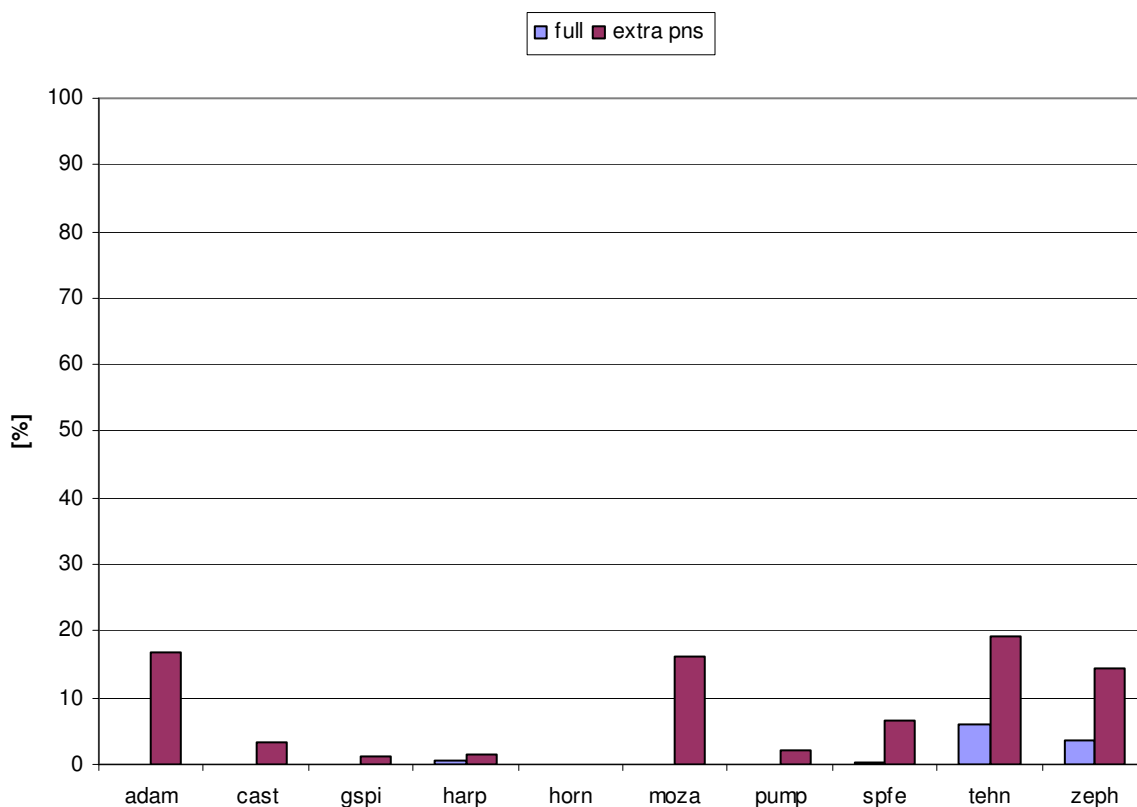
9.2.8.10 PNS upotreba

Upotreba PNS algoritma u RAAC koderu je ilustrirana na Slici 9.21 sa dvije konfiguracije koder. Prva je *Full* konfiguracija koja objedinjuje sve algoritme izvedene u koderu (Tablica 9.4), a upotreba PNS algoritma je strogo limitirana preko tonaliteta i posebno preko percepcijske entropije (vidi Poglavlje 8.3 za objašnjenje izvedbe PNS algoritma u RAAC koderu). Na Slici 9.21 je vidljivo da je po ovom postroženom kriteriju PNS algoritam upotrebljavan dosta rijetko, i to samo kod uzoraka *tehn* i *zeph* (4% i 6% pojaseva).

Kod druge konfiguracije *Extra pns* (Tablica 9.5) kod koje je limit entropije značajno podignut (sa -200 na -2000), upotreba PNS algoritma je puno veća, a osim *tehn* i *zeph* (19% i 14%) uzoraka, ističu se i *adam*, *moza* i *spfe* (17%, 16% i 7%).

Zaključujemo da je u RAAC koderu moguće vrlo precizno kontrolirati upotrebu PNS algoritma, što je neophodno zbog činjenice da je sam algoritam vrlo nepredvidiv – kod nekih signala jako podiže kvalitetu signala dok druge vrlo lako degradira. Npr. tranzijentni, složeni uzorci *zeph* i *spfe* jako profitiraju od upotrebe PNS algoritma, dok stacionarni, tonalni uzorci kao što su *horn* i *gspi* forsiranom upotrebom PNS algoritma gube na kvaliteti. Limitiranjem upotrebe PNS algoritma s jedne strane je izbjegnuta degradacija spomenutih uzoraka, dok je s druge strane PNS ipak upotrijebljen kod signala kojima podiže kvalitetu.

Slika 9.21: Postotak upotrebe PNS algoritma kod dvije konfiguracije RAAC koda: *Full* i *Extra pns*.



9.2.8.11 Upotreba *intenzitet-stereo* kodiranja

Upotreba *intenzitet-stereo* algoritma je analizirana u 6 slučajeva. Izabrane su dvije frekvencije iznad kojih se IS algoritam upotrebljava, a to su 10 kHz i 6 kHz. Za svaku frekvenciju analizirana je upotreba IS algoritma u tri slučaja:

- 1) bez limitiranja IS obzirom na tonalitet signala,
- 2) sa limitiranjem IS ako je tonalitet pojasa iznos 0.3,
- 3) sa limitiranjem IS ako je tonalitet pojasa iznos 0.1.

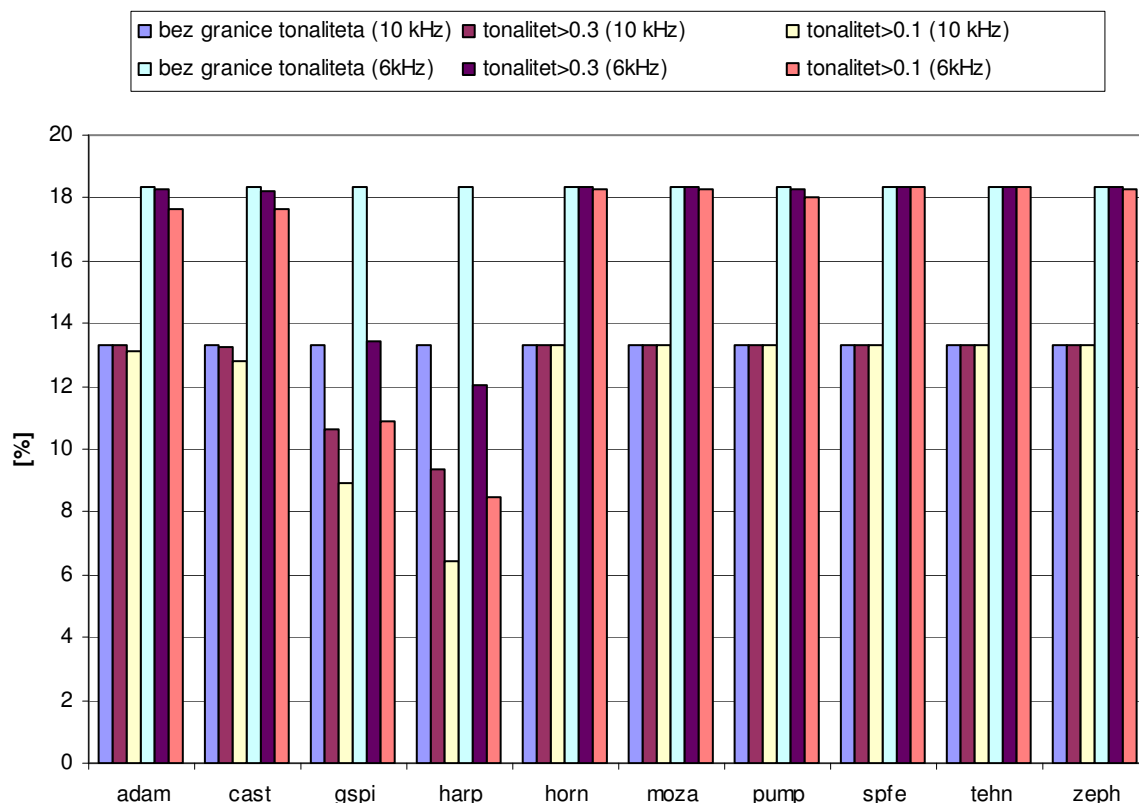
Slučaj u kome je IS frekvencija 10 kHz, a tonalitet 0.3 odgovara konfiguracijama koder *Normal* i *Full*.

Rezultati testiranja su prikazani na Slici 9.22. Očito je da tonalitet limitira upotrebu IS algoritma samo kod *gspi* i *harp* uzoraka koji jedini imaju izraženu tonalnost (Slika 9.16). Kod uzorka *gspi* upotreba IS algoritma na 10 kHz podrazumijeva 18.33% pojaseva u slučaju da ne postoji limit obzirom na tonalitet signala. U slučaju kada je upotreba IS algoritma limitirana tonalitetom iznosa 0.3, postotak upotrebe IS algoritma je pao za 2.7%, a u drugom slučaju kada je limit tonaliteta 0.1, IS upotreba je pala za 4.4%. Iako je uzorak *harp* općenito manje tonalan od uzorka *gspi*, limitiranje IS algoritma tonalitetom je više izraženo (upotreba IS pada za 3.95% za tonalitet 0.3 i za 6.90% za tonalitet 0.1). Razlog je činjenica da uzorak *harp* ima više tonalnih komponenti na višim frekvencijama gdje se IS i upotrebljava.

Na drugoj frekvenciji od 6 kHz primjećuje se, relativno prvom slučaju, veći utjecaj tonaliteta na limitiranje IS upotrebe. Tako je kod *gspi* uzorka postotak upotrebe pao za 4.9% (tonalitet 0.3) i za 7.5% (tonalitet 0.1), a kod *harp* uzorka za 6.3% i za 9.9%.

Druga grupa uzoraka koja vrlo malo reagira na limitiranje upotrebe IS algoritma uključuje uzorke *adam*, *cast* i *pump*. Uzorci *horn*, *moza*, *spfe*, *tehn* i *zeph* uopće ne reagiraju na limitiranje upotrebe IS algoritma tonalitetom, jer je njihov tonalitet izrazito nizak.

Slika 9.22: Upotreba IS algoritma kod RAAC koda. Samo izrazito tonalni signali imaju smanjenu upotrebu IS algoritma kada se IS limitira obzirom na tonalitet.



9.3 Objektivna analiza

Objektivna analiza kvalitete kodiranih audio signala vrši se posebno razvijenim programskim alatima koji se mogu smatrati nekom vrstom *matematičkih simulatora* ljudskog slušnog sistema. Ovakvi alati su mnogo pogodniji za analizu jer puno brže i jednostavnije dovode do konkretnih rezultata. Ipak treba napomenuti da su metode objektivne analize dovedene u pitanje od strane mnogih priznatih znanstvenika na području audio kodiranja, pa ih zato treba uzeti sa rezervom [54]. Sigurno je međutim da ovakvi alati mogu pouzdano odrediti odnose među različitim izvedbama koda, a također pomažu i kod usporedbe različitih izvedbi jednog te istog koda.

9.3.1 Standard ITU-R BS. 1387

Razvoj alata za objektivnu analizu je započet krajem 1980-ih godina, dok je 1994. godine ITU-R organizacija započela proces standardizacije. Prikupljeno je i ocijenjeno šest dotadašnjih izvedbi. Zaključak je bio da niti jedna izvedba nije superiorna ostalima u svim

segmentima, pa je odlučeno da se od svake izvedbe uzme pojedini dio i tako donese novi standard. Rezultat usklađivanja je bio standard ITU-R BS. 1387 nazvan "Method for Objective Measurements of Perceived Audio Quality" donesen 1999. godine. Sinonim za ovaj standard je skraćenica PEAQ (engl. *Perceptual Evaluation of Audio Quality*), ili *percepcijska ocjena audio kvalitete*. PEAQ je trenutno jedini kompletni sistem za objektivnu evaluaciju kodiranih audio signala koji je danas u svijetu priznat.

Ulaz u PEAQ sistem čine dva signala, originalni i kodirani. Nakon procesiranja oba signala, kao najvažniji izlaz iz PEAQ sistema dobije se općenita ocjena kvalitete predstavljena kao ODG (engl. *Objective Difference Grade*) broj. ODG je broj u rasponu od 0 do -4 i lako se uspoređuje sa subjektivnom ocjenom čiji je raspon vrijednosti od 1 do 5 (Slika 9.3). Osim ODG broja, kao izlaz iz PEAQ sistema dobije se i niz MOV varijabli (engl. *Model Output Variables*) iz kojih je ODG proračunat, a pomoću kojih se dobije bolji uvid u kvalitetu audio koda.

PEAQ standard definira dva modela za analizu, osnovni (engl. *Basic*) i napredni (engl. *Advanced*). Dva modela se razlikuju po složenosti – osnovni model je jednostavniji i namijenjen je za izvedbe u stvarnom vremenu, dok je napredni model složeniji i bolji ali neprikladan za izvedbe u stvarnom vremenu. Jedna od bitnijih razlika je i činjenica da osnovni model simulira ljudski slušni sistem na bazi DFT transformacije, dok napredni model upotrebljava osim DFT transformacije i filtarski slog. Kao rezultat analize dva modela daju različite MOV varijable.

Napredni model daje sljedećih 5 izlaznih MOV varijabli:

- *RmsModDiff*,
- *RmsNoiseLoudAsym*,
- *AvgLinDist*,
- *SegmentalNoiseMaskRatio*,
- *ErrHarmonicStructure*.

RmsModDiff predstavlja prosjek kvadriranih razlika u modulaciji vremenskih envelope dva signala.

RmsNoiseLoudAsym predstavlja zbroj kvadriranog prosjeka glasnoće šuma i glasnoće izgubljenih frekvencijskih komponenti.

AvgLinDist je mjera glasnoće komponenti signala koje su izgubljene.

SegmentalNoiseMaskRatio je linearno uprosječen lokalni omjer praga maskiranja i šuma.

ErrHarmonicStructure predstavlja mjeru harmoničnosti šuma (pod određenim uvjetima šum može poprimiti sličan frekvencijski spektar kao i sam signal i tako postati izraženo čujan).

9.3.2 OPERA program

Kvaliteta RAAC koda ocijenjena je programskom izvedbom standarda ITU-R BS.1387 tvrtke Opticom (Erlangen, Njemačka) pod nazivom OPERA (verzija 3.5). Testiranje je izvedeno na probnoj verziji programa ograničenoj na 100 testova, tako da su provedeni samo osnovni testovi.

9.3.3 Rezultati

Izvršeno je nekoliko vrsta testiranja. Prvo je testirano i međusobno uspoređeno 5 konfiguracija RAAC koda preko 10 uzoraka koji su odabrani kod subjektivnog testiranja u prvom dijelu ovog poglavlja. Ovo testiranje je provedeno samo na bitskom protoku od 128 kb/s/stereo. Nakon toga je konfiguracija *Normal* testirana na tri bitska protoka: 96, 128 i 160 kb/s/stereo. Na kraju su rezultati konfiguracije *Normal* uspoređeni sa dvije alternativne trenutno dostupne izvedbe MPEG-4 GA koda.

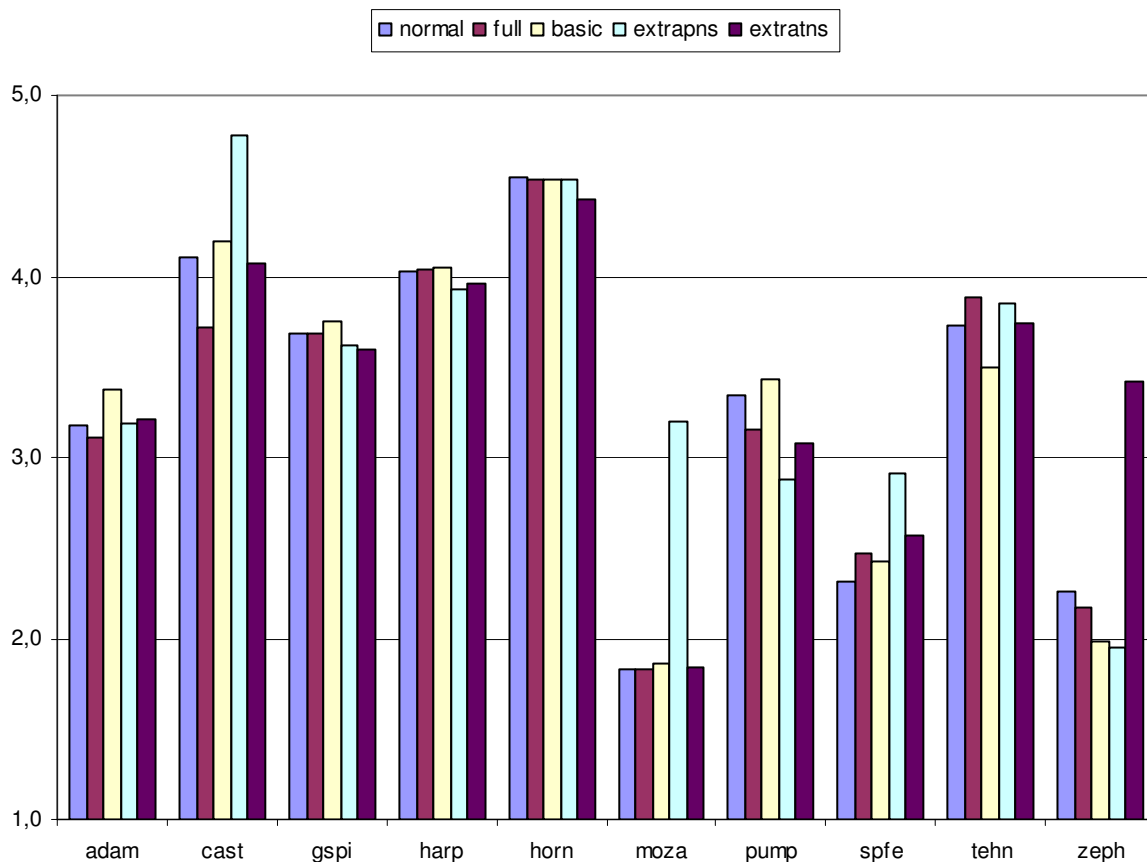
Objektivna usporedba različitih konfiguracija RAAC koda na 128 kb/s/stereo

Dobiveni rezultati ovog testa su prikazani na Slici 9.23. Prvi zaključak je da koder kao i kod subjektivnih testiranja daje najbolje rezultate kod vrlo tonalnih signala i onih uskog frekvencijskog raspona. Tako je ubjedljivo najbolji rezultat dao *horn* uzorak čiji rezultat je (u prosjeku) odličnih 4.52. Osim ovog uzorka zadovoljavajući rezultat daju *cast* (4.17), *gspe* (3.67), *harp* (4.00) i *tehn* (3.74). Srednje dobre rezultate daju uzorci *adam* (3.21) i *pump* (3.18). Najlošije rezultate daju *moza* (2.11), *spfe* (2.54) i *zeph* (2.36).

Pažljivijom analizom može se uočiti da RAAC koderu najveće probleme prave uzorci sa dosta velikim frekvencijskim rasponom, a koji su istovremeno složeni, a to je ponajviše *zeph* uzorak. Uzorak *harp* je također širokopojasan ali on nije suviše složen pa je njegov rezultat znatno bolji, što potvrđuje gornje zaključke.

Najveće negativno iznenađenje objektivne analize je uzorak *moza* koji daje vrlo slabe rezultate, za razliku od subjektivne analize kod koje ovaj uzorak dobiva odlične ocjene. Ostaje nejasno zašto je ovo slučaj iako je očito da *AvgLinDist* MOV izlazna varijabla ima izrazito visoke vrijednosti.

Slika 9.23: Rezultati objektivnog testiranja 10 odabranih uzoraka sa 5 različitih konfiguracija RAAC koda na bitskom protoku 128 kb/s.



Uočljivo je i da su različite konfiguracije koda prilično slične i da daju neznatno različite rezultate. Nekoliko je iznimaka ovom pravilu.

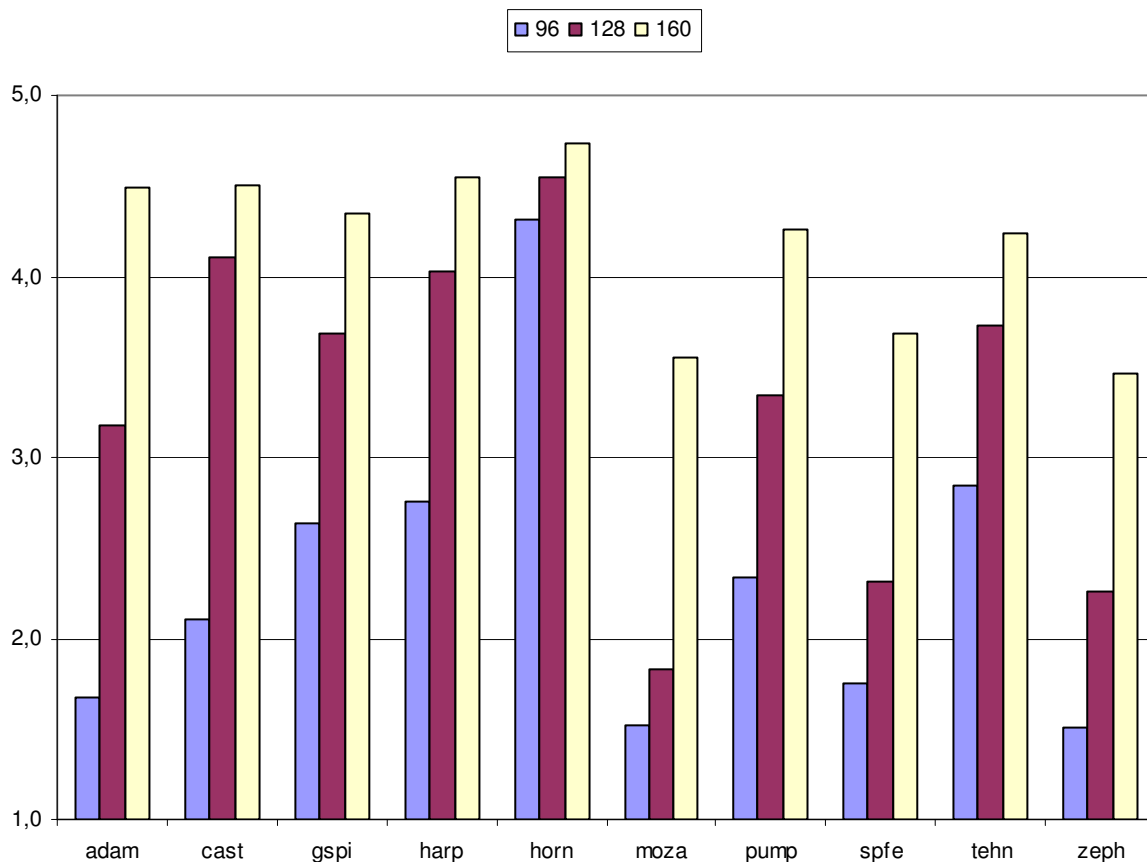
Kod uzorka *cast*, *moza* i donekle *spfe* upotreba PNS algoritma znatno povećava ODG ocjenu. Subjektivnom provjerom ova pojava je potvrđena kod *moza* i *spfe* uzoraka, dok naprotiv kod *cast* uzorka upotreba PNS algoritma pogoršava subjektivnu kvalitetu signala.

Uočljivo je i da TNS algoritam kod uzorka *zeph* znatno podiže ODG ocjenu. Ovo međutim nije u skladu sa rezultatima subjektivnog testiranja.

Objektivna usporedba konfiguracije Normal na različitim bitskim protocima

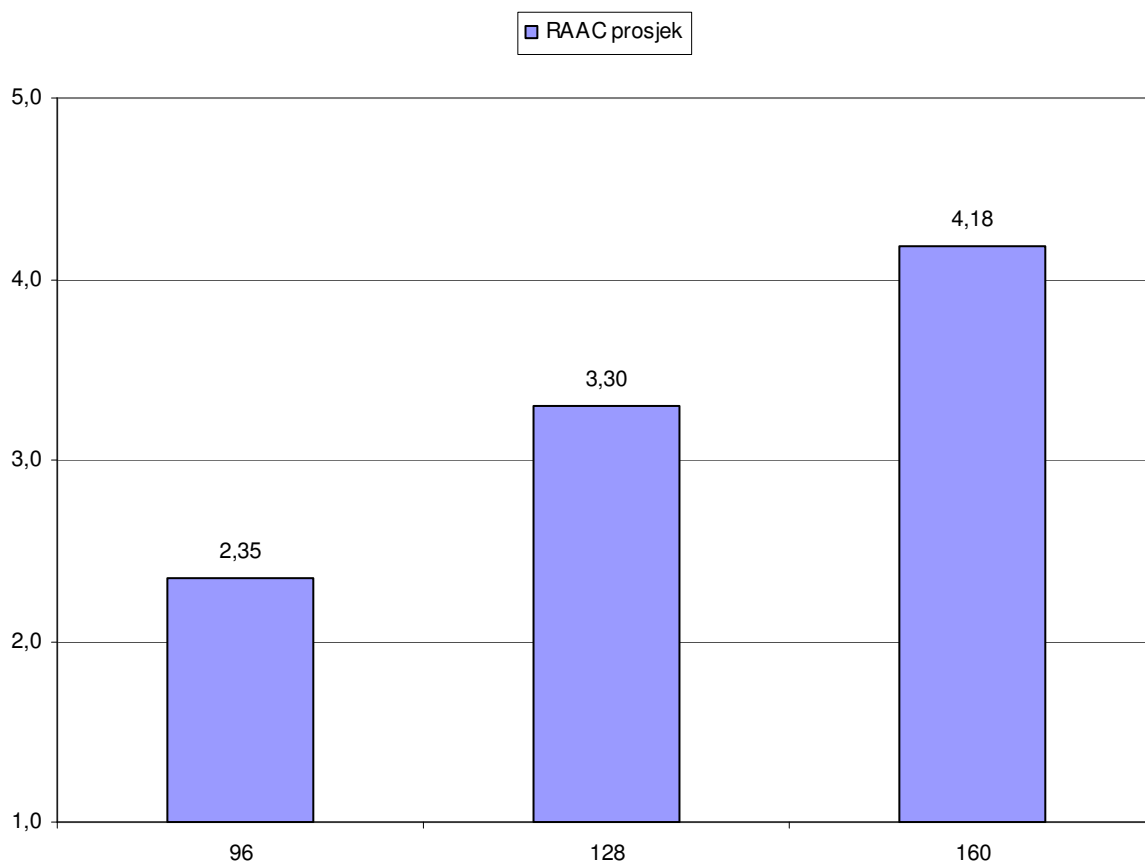
RAAC koder je testiran na tri bitska protoka: 96, 128 i 160 kb/s/stereo, svaki bitski protok samo na konfiguraciji *Normal*. Dobiveni rezultati ovog testa su prikazani na Slici 9.24.

Slika 9.24: Rezultati objektivnog testiranja 10 odabranih uzoraka na 3 različita bitska protoka konfiguracijom *Normal*.



Značajno je primijetiti da većina uzoraka nema linearan rast kvalitete sa linearnim porastom bitskog protoka. Iznimka je uzorak *horn*. Ova pojava dokazuje da uzorci imaju veći skok u kvaliteti, ukoliko između spomenutih bitskih protoka postignu određenu optimalnu kvalitetu. Tako bolje ocijenjeni uzorci imaju skok između 96 i 128 kb/s što znači da su već prije 128 kb/s postigli optimalnu kvalitetu i da većih skokovitih poboljšanja više ne može biti. U radikalne predstavnike ovakvih uzoraka spadaju *cast* i *harp* uzorci koji imaju jako velik skok u kvaliteti između 96 i 128 kb/s (1.99 kod *cast* i 1.27 kod *harp*), i puno manji skok između 128 i 160 kb/s (0.40 kod *cast* i 0.52 kod *harp*). U drugu skupinu uzoraka spadaju uzorci koji imaju veći skok između bitskih protoka 128 i 160 kb/s. U ovu skupinu spadaju uzorci *moza* i *spfe* koji očito nisu kod bitskog protoka od 128 dostigli optimalnu kvalitetu.

Na Slici 9.25 je prikazana prosječna ocjena svih uzorka na pojedinim bitskim protocima. Zaključujemo da je kvaliteta kod 160 kb/s/stereo ocjenjena kao "*razlika primjetna ali ne smeta*", kod 128 "*malo smeta*", a kod 96 "*smeta*".

Slika 9.25: Prosječna ocjena RAAC konfiguracije *Normal* na tri bitska protoka.

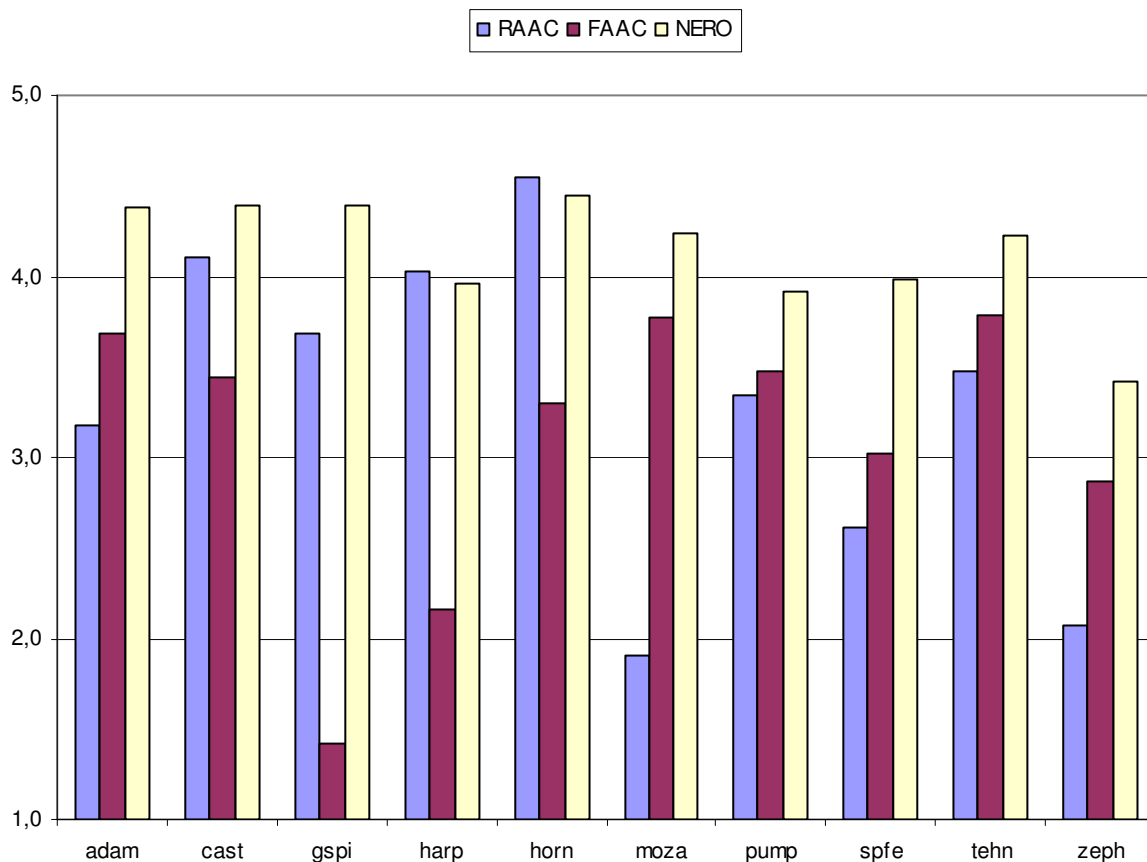
Objektivna usporedba RAAC koda sa drugim izvedbama

Objektivna usporedba RAAC koda je obavljena sa dvije trenutno dostupne izvedbe MPEG-4 audio koda koje se nazivaju FAAC i NERO.

FAAC (engl. *Free Advanced Audio Codec*) je javni projekt (<http://faac.sourceforge.net>) u čijem razvoju je učestvovalo i još uvijek učestvuje više stručnjaka i programera. Projekt je star nekoliko godina, a koder je konstantno bio izvrnut javnom testiranju i ocjenjivanju, dakako i poboljšavanju i usavršavanju. Trenutno je jedini javno dostupni MPEG-4 audio koder.

NERO koder je komercijalni proizvod tvrtke Ahead (Karlsbad, Njemačka) i trenutno je jedan od najkvalitetnijih MPEG-4 GA audio koda na tržištu. Koder se isporučuje kao dodatak (engl. *plugin*) za software za snimanje CD medija, pa tako omogućuje direktno snimanje glazbenih CD-a u MPEG-4 audio format.

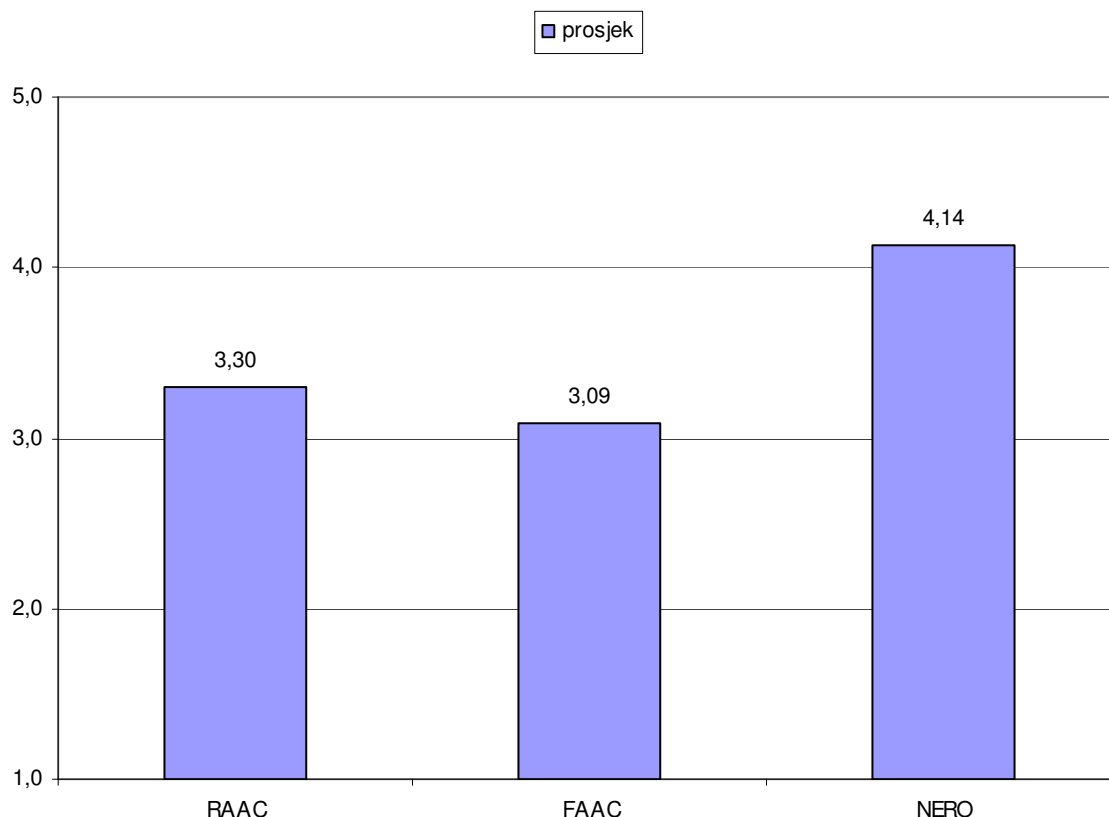
Slika 9.26: Usporedba RAAC koda na bitskom protoku 128 kb/s sa dvije trenutno dostupne izvedbe: FAAC i NERO.



Rezultati usporedbe su prikazani na Slici 9.26. Na prvi pogled je očito da je najbolje rezultate (kod 8 od 10 uzoraka) dala NERO izvedba. Kod *harp* i *horn* uzorka pobjednik je RAAC, dok FAAC niti u jednom slučaju nije najbolji. Usporedimo li FAAC i RAAC kodere zaključujemo da je RAAC koder bolji kod izrazitije tonalnih signala (*gspe*, *harp*, *horn*), dok je FAAC bolji kod širokopolasnih, složenih uzoraka kod kojih RAAC ima većih problema. FAAC sa druge strane postiže loše rezultate kod tonalnih uzoraka što je potvrđeno i subjektivnim provjerama.

Na Slici 9.27 prikazane su prosječne ocjene tri koda na bitskom protoku od 128 kb/s. Najbolji rezultat je postigao NERO koder koji je ukupno ocijenjen sa "*razlika primjetna ali ne smeta*", dok su FAAC i RAAC ocijenjeni sa "*neznatno smeta*", s tim da je RAAC za nijansu bolje ocijenjen od FAAC koda.

Slika 9.27: Prosječna objektivna ocjena RAAC, FAAC i NERO koda na bitskom protoku od 128 kb/s/stereo.



9.3.4 Usporedba rezultata objektivnog sa subjektivnim testiranjem

Podijelimo li testirane uzorke u tri skupine – lagani, srednje teški i vrlo teški – lako možemo doći do zaključka o sličnosti/različitosti rezultata subjektivnog i objektivnog testiranja.

Uzorci koji su objema metodama svrstani u istu skupinu su: *adam* (srednje težak), *gspe* (lagan), *harp* (lagan), *horn* (lagan), *spfe* (vrlo težak) i *zeph* (vrlo težak). Uzorci koji se razlikuju za jednu kategoriju – što je prihvatljivo – su *pump* (sub. lagan, obj. srednje težak), *cast* (sub. srednje težak, obj. lagan) i *tehn* (sub. srednje težak, obj. lagan). Jedini uzorak koji se sasvim suprotno ocijenjen je *moza* (sub. lagan, obj. vrlo težak).

Zaključujemo da je 60% uzoraka svrstano u istu skupinu, 30% ih je u susjednim skupinama, dok je 10% ocijenjeno sasvim suprotno. Ukupno gledajući, rezultati se uglavnom (90%) podudaraju ili neznatno razlikuju pa ih možemo smatrati vjerodostojnima.

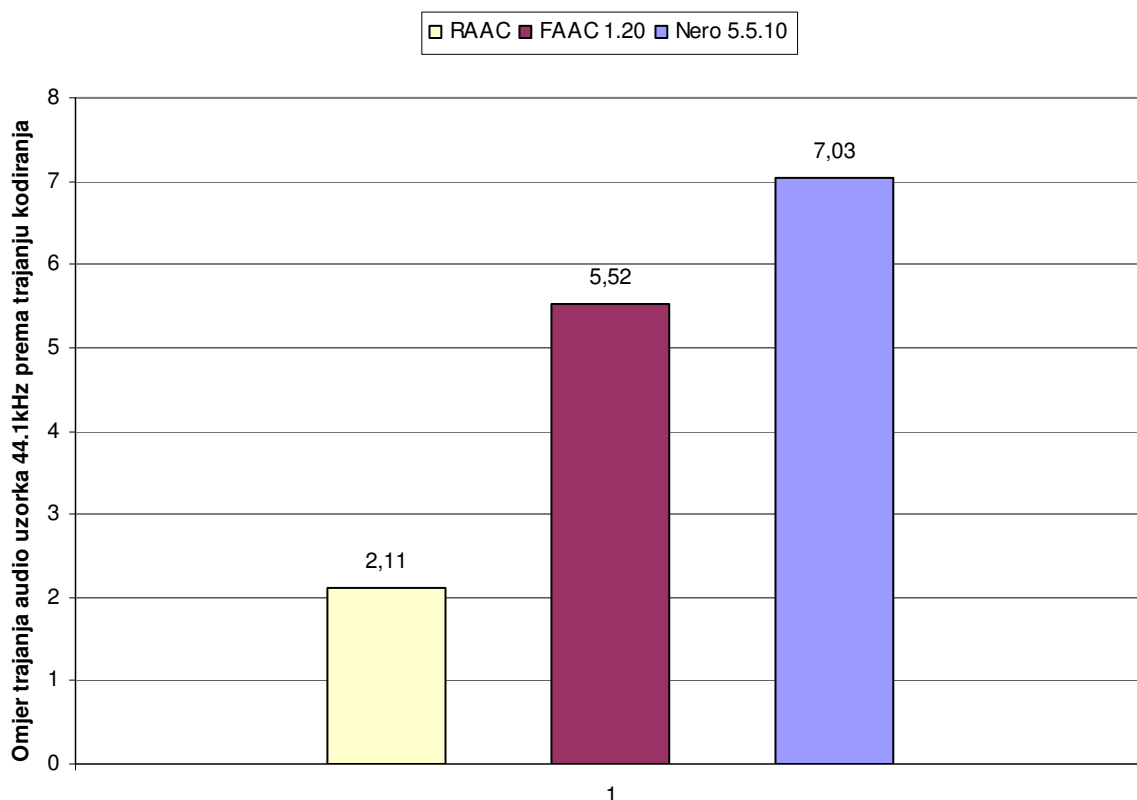
9.3.5 Zaključak

Iako su obavljeni samo na ograničenom broju uzoraka, objektivni testovi pokazali su neke zanimljive rezultate. Iako su objektivne ocjene vrlo upitne ako se gledaju pojedinačni uzorci i porede sa subjektivnim testovima, njihovom analizom može se dobiti dobar općeniti uvid u cjelokupnu vrijednost enkodera. Tako je jasno da RAAC iznimno dobro kodira čiste tonalne signale, a lošije širokopojasne i visoko-frekvencijske. Također, može se reći sa popriličnom sigurnošću da RAAC na bitskom protoku od 128 kb/s spada u kategoriju "*neznatno smeta*".

9.4 Brzina RAAC koda

Iako brzina nije bila od primarnog interesa pri izvedbi RAAC koda, ipak je određenim optimiranjem unutar koda postignuta prihvatljiva brzina koda. Tako RAAC, u prosjeku, kodira audio uzorak duplo brže od njegovog trajanja, što u poređenju sa danas pristupačnim komercijalnim izvedbama MPEG-4 audio koda, nije loš rezultat.

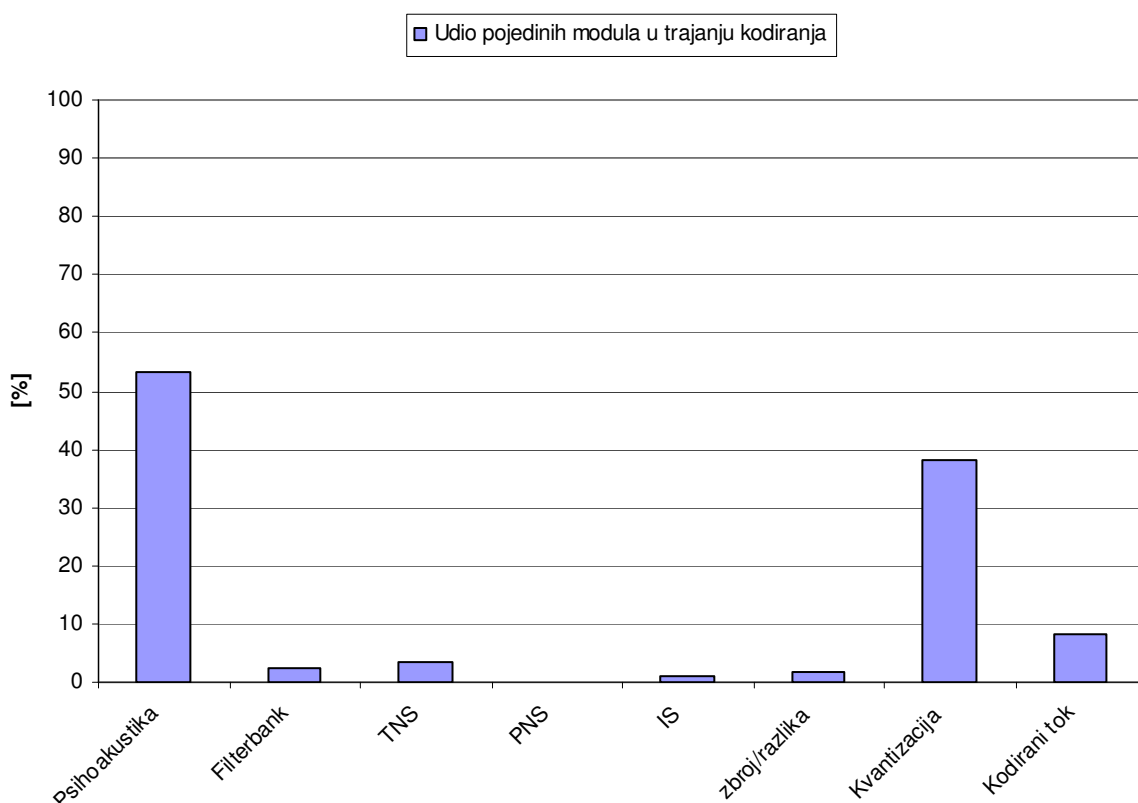
Slika 9.28: Rezultat usporedbe brzine RAAC koda sa FAAC i NERO izvedbama.



Za ilustraciju, brzina RAAC koda je uspoređena sa FAAC i NERO – visoko optimiranim izvedbama MPEG-4 audio koda. Rezultat usporedbe je prikazan na Slici 9.28.

Udio pojedinih dijelova koda u potrošnji vremena je prikazan na Slici 9.29. Psihoakustički modul i modul za kvantizaciju sami troše 90% vremena, pa je, u budućim poboljšanjima, unutar ovih modula potrebno tražiti prostor za ubrzanje rada koda.

Slika 9.29: Utrošak vremena pojedinih dijelova RAAC koda u ukupnom trajanju kodiranja.



9.5 Zaključak

Nakon provedene subjektivne i objektivne analize signala kodiranih sa RAAC koderom, može se zaključiti da RAAC sa uspjehom postiže ocjenu "*razlika primjetna ali ne smeta*" (ocjena veća od 4.0) kod velike većine izabranih audio uzoraka, kod bitskog protoka 112 kb/s za stereo signal. Teži audio uzorci zahtijevaju nešto veći bitski protok od 128 kb/s, dok uzorci sa užim frekvencijskim rasponom istu ocjenu postižu čak na niskom bitskom protoku od 96 kb/s. Konfiguracija RAAC koda koja na zadovoljavajući način kodira većinu signala je *Normal*, s tim da je potrebno dodati restriktivnu upotrebu PNS algoritma koji može biti jako koristan, ali

može u nekim slučajevima i pogubno utjecati na konačnu kvalitetu signala, posebno na bitskim protocima višim od 128 kb/s.

Glavni problem koderu predstavlja širok frekvencijski raspon signala i VF tranzijenti, na što treba obratiti posebnu pažnju kod eventualnog usavršavanja koder.

Poglavlje 10

10 Zaključak

10.1 Cilj razvoja RAAC koda i postignuto

Ugrubo postavljen, početni cilj rada bio je razvoj MPEG-4 audio koda. Nakon početne analize MPEG-4 standarda uočena je osnovna struktura MPEG-4 audia – MPEG-4 audio je podijeljen na tri dijela: kodiranje govora, kodiranje širokopojasnog audio signala i kodiranje sintetičkog audio signala. Fokus rada je tada stavljen na kodiranje širokopojasnog, složenog audio signala. Dio MPEG-4 standarda koji se bavi kodiranjem složenog audia naziva se MPEG-4 General Audio. MPEG-4 General Audio je dalje podijeljen na *Profile* obzirom na ciljane primjene koda, dok je unutar *Profila* definiran veći broj *Audio Objekata*. Nakon analize *Profila* i *Audio Objekata* zaključeno je da fokus rada bude izvedba *Low Complexity* audio objekta koji je dio 4 profila (*Main*, *Scalable*, *High Quality* i *Natural Audio*). *Low Complexity* je najsloženiji audio objekt unutar GA dijela standarda, jer u sebi sadrži gotovo sve bitnije dijelove percepcijskog audio koda – zato je njegova izvedba sasvim dovoljna, a i namijenjena, za postizanje transparentne kvalitete složenog audio signala. Cilj je dakle bio razvoj MPEG-4 General Audio koda fokusiranog na izvedbu *Low Complexity* audio objekta. Koder je nazvan RAAC prema *RASIP Advanced Audio Coding*.

Nakon početne analize psihoakustičkih svojstava koja se danas koriste u percepcijskom audio kodiranju, a također i ljudskog slušnog sistema na koji je psihoakustika velikim dijelom oslonjena, započeto je sa realizacijama jednog po jednog bloka koda. Prvo je realizirana osnovna, *kostur* struktura, koja je sadržavala samo module *LC* audio objekta koji su neophodni za generiranje osnovnog kodiranog audio toka: MDCT filtarski slog, modul za kvantizaciju, Huffmanovu kompresiju i pakiranje u kodirani tok. Zbog jednostavnosti, kodiran je samo jedan kanal, a nisu bili podržani kratki blokovi.

Nakon početne konfiguracije, realiziran je psihoakustički modul, nakon čega su dobiveni prvi kodirani audio uzorci sa donekle dobrom kvalitetom. Nakon osnovne strukture audio koda, pristupljeno je realizaciji naprednijih blokova, redom: intenzitet-stereo, percepcijska zamjena šuma, zbroj/razlika i vremensko uobličavanje šuma. U međuvremenu je koder proširen na kodiranje stereo signala.

Nakon izvedbe nabrojanih blokova, RAAC koder je postizao solidnu kvalitetu kodiranog audio signal – kod 160 kb/s većina uzoraka je ocijenjena sa ocjenom 3.5 do 4.0.

Nakon što je zaokružena realizacija svih modula *Low Complexity* audio objekta, razvoj koda je usmjeren na poboljšanja izvedbi pojedinih modula. Tako su najprije izvršene značajne izmjene na psihoakustičkom modelu. Zatim su usavršeni PNS, IS i MS moduli. Realizirano je i napredno upravljanje sa *bit-rezervoarom* i *bitovima za popunjavanje*. Sa ovako usavršenim koderom, na kraju je pristupljeno subjektivnoj i objektivnoj evaluaciji.

Paralelno sa usavršavanjem pojedinih modula koda, izvršena je i njihovo optimiranje obzirom na brzinu izvršavanja. Tako je koder višestruko ubrzan u odnosu na početnu izvedbu. Pri spomenutom optimiranju nije vođeno računa o potrošnji memorije, tako da na ovu činjenicu treba obratiti pažnju kod budućih usavršavanja koda.

Glavni zaključak koji se nameće nakon evaluacije RAAC koda, je sljedeći: RAAC koder ni po čemu ne zaostaje za sličnim, akademskim izvedbama *mp3* koda. Čak i više od toga: RAAC koder je sa uspjehom pomaknuo granicu prihvatljivog *mp3* audio kodiranja sa 128 kb/s (stereo) na najmanje 112 kb/s, dok je u većini srednje složenih audio uzoraka kodiranje sasvim zadovoljavajuće za prosječnog PC, *discman* ili Internet korisnika, i na bitskom protoku od 96 kb/s.

Ostali značajni rezultati RAAC izvedbe manifestiraju se kroz novine uvedene u izvedbe pojedinih modula – posebno psihoakustičkog modula, zatim algoritma za detekciju tonaliteta i tranzijentnosti, PNS modula, MS modula i IS modula. Svaka od ovih izvedbi daje svoj doprinos posebnosti RAAC izvedbe i ističe njegov doprinos razrađenosti i savršenosti pojedinih algoritama koji su dio MPEG-4 audio standarda.

10.2 Nadogradnje RAAC koda

Iako RAAC koder predstavlja akademsku izvedbu MPEG-4 audio koda, čiji je prvenstveni cilj istraživanje različitih izvedbi i unapređivanje MPEG-4 audio algoritama, nastavak razvoja RAAC koda može biti usmjeren i prema profesionalnom, komercijalnom plasmanu.

Ponajprije, trenutnu izvedbu RAAC koda trebalo bi nastaviti u dva smjera. Prvi bi bio dodatno optimiranje i ubrzanje algoritama, što bi otvorilo mogućnosti plasmana RAAC koda u vidu zasebnog (engl. *stand-alone*) PC koda ili koda ugrađenog u složenije programe. Drugi, poželjan smjer bila bi izvedba u aritmetici nepomičnog zarezaja koja bi otvorila vrata uklapanju RAAC koda u ugrađenim i mobilnim DSP sustavima.

Dakako, RAAC koder je moguće nastaviti usavršavati i akademskim istraživanjima, a poželjno je i njegovo uklapanje u postojeće akademske MPEG-4 video sustave.

10.3 Budućnost MPEG-4 standarda

Gleda li se popularnost upotrebe, unatoč očitim prednostima, MPEG-4 još uvijek značajno zaostaje za, već 5 godina planetarno popularnim, *mp3* audio formatom. Glavni razlog je činjenica da MPEG-4 standard sadrži snažno zaštićena autorska prava na većinu algoritama koji ga sačinjavaju, za razliku od *mp3* formata koji je besplatan.

Ipak, u zadnje vrijeme dolazi do snažnog proboja MPEG-4 video sadržaja, a samim tim i MPEG-4 audia. Trenutne profesionalne programske izvedbe MPEG-4 GA uključuju *QuickTime 6.2*, *Nero 5.5*, *iTunes 4*, *Envivio TV* (dodatka za *Microsoft Windows Media Player* i *RealOne* player), itd. MPEG-4 audio je podržan i u sve više prijenosnih uređaja kao npr. *Apple iPod*, *Compaq iPaq*, *Philips Expanium*, *Nokia N-Gage* (Slika 10.1), *Panasonic SV-SD80*, itd.

Slika 10.1: Budućnost MPEG-4 standarda: *Nokia N-Gage* – igraća konzola, mobitel, radio, audio player – sve u jednom uređaju. Moguća reprodukcija MPEG-4 audio datoteka, a također i snimanje sa ugrađenog radio prijemnika direktno u MPEG-4 audio format (uređaj će biti komercijalno dostupan u jesen 2003.).



Uzme li se u obzir povijest multimedije, od pojave Interneta do danas, nesumnjivo je da MPEG-4 standard predstavlja budućnost, kako video, tako i audio kodiranja.

Dodatak A

A Sadržaj priloženog CD-a

- 1) Izvršne datoteke RAAC izvedbe za MS Windows okruženje:
 - RAACcon.exe – *console* aplikacija – debug i razvoj
 - RAACdev.exe – GUI aplikacija – kodiranje jedne datoteke uz precizno postavljenje parametara koda
 - RAACgo.exe – GUI aplikacija – uzastopno kodiranje više datoteka
 - RAACcdex.exe – GUI aplikacija – integriranje sa CDex programom, RAAC kao eksterni koder
 - RAACpublic.exe – GUI aplikacija – uzastopno kodiranje više datoteka.
- 2) Izvorni kod RAAC izvedbe (uključuje Microsoft Visual Studio 6.0 *project*).
- 3) Audio uzorci provedenih testiranja RAAC koda:
 - originalni uzorci (wav),
 - kodirani uzorci (aac) i
 - dekodirani uzorci (wav).

Dodatak B

B Popis korištenih alata

Alat korišten u razvoju RAAC audio koda	Opis
Microsoft Visual C++ 6.0® www.microsoft.com	Radna okolina korištena za programiranje u C i C++ programskim jezicima.
WaveLab® www.steinberg.com	Program korišten za editiranje i dekodiranje .wav datoteka i dinamičku spektralnu analizu.
CoolEdit Pro® www.sintylum.com	Program korišten za editiranje i playback .wav datoteka i dinamičku spektralnu analizu.
MPEG4IP www.mpeg4ip.org	Izvorni kod korišten za pretvaranje .aac u .mp4 datoteke.
MatLab® R12 www.mathworks.com	Pseudo-programerski alat korišten za DSP analizu.
ISO 14496-3 source www.iso.ch	ISO izvorni kod osnovne izvedbe audio koda prema normi 14496-3.
Nero® www.ahead.de	Program korišten za kodiranje .aac datoteka za usporedbu.
FAAD (Free AAC Decoder) www.audiocoding.com	Winamp dodatak za dekodiranje .aac ili .mp4 datoteka i konverziju .aac u .wav.
FAAC (Free AAC Coder) www.audiocoding.com	Winamp dodatak za kodiranje .aac ili .mp4 datoteka za usporedbu.
Winamp® www.winamp.com	Player korišten zajedno sa FAAD dodatkom za reprodukciju .aac i .mp4 datoteka.
PHILIPS Expanium® www.expanium.philips.com	Discman korišten za reprodukciju .aac (MPEG-2 kompatibilnih) datoteka.
QuickTime® www.apple.com	Player korišten za reprodukciju i generiranje .mp4 audio datoteka za usporedbu.
OPERA® www.opticom.de	Alat za objektivnu evaluaciju kodiranog signala prema PEAQ modelu (ITU-R BS.1387).
ABCHR www.ff123.net	Program korišten za testiranje i usporedbu kodiranih .aac datoteka sa originalnim datotekama.
ABX www.pcabx.com	Program korišten za testiranje i usporedbu kodiranih .aac datoteka sa originalnim datotekama.

Reference i bibliografija

- [1] Aggarwal A., Regunathan S., Rose K., Near-Optimal Selection of Encoding Parameters for Audio Coding, *Proc. ICASSP 2001*, Salt Lake City, Utah, 2001.
- [2] Aggarwal A., Regunathan S., Rose K., Compander Domain Approach to Scalable AAC, *AES 110th Convention*, Amsterdam, NL, 2001.
- [3] Algazi, R., Divenyi P., Duda R. O., Subject Dependent Transfer Functions in Spatial Hearing, *Proc. IEEE MWSCAS Conference*, 1997.
- [4] Allamanche E., Geiger R., Herre J., Sporer T., MPEG-4 Low Delay Audio Coding Based on the AAC Codec, *AES 106th Convention*, 1999.
- [5] Baumgarte F., A Physiological Ear Model for Auditory Masking Applicable to Perceptual Coding, *AES 103th Convention*, 1997.
- [6] Baumgarte F., Application of a Physiological Ear Model To Irrelevance Reduction in Audio Coding, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [7] Baumgarte F., Evaluation of a Physiological Ear Model Considering Masking Effects to Audio Coding, *AES 105th Convention*, San Francisco, CA, 1998.
- [8] Baumgarte F., Faller C., Why Binaural Cue Coding is better than Intensity Stereo Coding, *AES 112th Convention*, Munich, Germany, 2002.
- [9] Baumgarte F., Ferekidis C., Fuchs H., A Nonlinear Psychoacoustic Model Applied to the ISO MPEG Layer 3 Coder, *AES 99th Convention*, New York, NY, 1995.
- [10] Blauert J., *Spatial Hearing*, MIT Press, 1983.
- [11] Bosi M., Multichannel Audio Coding and Its Application in DAB and DVB, *Journal of Audio Engineering Society*, vol. 46, no. 6, 2000.
- [12] Bosi M., Davidson G., High Quality, Low-Rate Audio Transform Coding for Transmission and Multimedia Application, *AES 93th Convention*, San Francisco, CA, 1992.
- [13] Bradbury J., Linear Predictive Coding, http://www.cs.queensu.ca/~bradbury/pdf/lpc_slides.pdf, 2002.
- [14] Brandenburg K., Introduction to Perceptual Coding, *Collected Papers on Digital Audio Bit-Rate Reduction*, Gilchrist and Grewin, 1996.

- [15] Brandenburg K., MP3 and AAC Explained, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [16] Brandenburg K., Popp H., An Introduction to MPEG Layer-3, *EBU Technical Review*, 2000.
- [17] Brandenburg K., Bosi M., "Overview of MPEG-audio: Current and future standards for low bit-rate audio coding", *Journal of the Audio Engineering Society*, vol. 45, pp. 4-21, 1997.
- [18] Brandenburg K., Kunz O., Sugiyama A., MPEG-4 Natural Audio Coding, *Signal Processing: Image communication*, vol. 15, pp. 423-444, 2000.
- [19] Colomes C., Schmidmer C., Thiede T., Treurniet W. C., Perceptual Quality Assessment for Digital Audio: PEAQ – the new ITU Standard for Subjective Measurement of the Perceived Audio Quality, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [20] Czyzewski A., Krolikowski R., Noise Reduction in Audio Employing Auditory Masking Approach, *AES 106th Convention*, Munich, Germany, 1999.
- [21] Decker E., Cole A., Jasper K., Fernandez A., Proposal For Hardware Processing of MPEG I Layer III,
- [22] Dimković I., Improved ISO AAC Coder, <http://www.psytel-research.co.yu/papers/>, 2002.
- [23] Dimković I., Milovanović D., MPEG Audio Implementacija i Optimizacija, <http://www.psytel-research.co.yu/papers/>, 2002.
- [24] Duenas A. D., Perez R., Rivas B., Alexandre E., Pena A. S., Realtime Implementation of MPEG-2 and MPEG-4 Natural Audio Coders, *AES 110th Convention*, Amsterdam, NL, 2001.
- [25] Fuchs H., Improving Joint Stereo Audio Coding by Adaptive Inter-Channel Prediction, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, 1993.
- [26] Fastl H., The Psychoacoustics of Sound-Quality Evaluation, *ACUSTICA*, vol. 83, no. 5, pp. 754-764, 1997.
- [27] Ferreira A. J. de S., *Spectral Coding and Post-Processing of High Quality Audio*, PhD Thesis, Universidade de Porto, Porto, 1998.
- [28] Garcia R. A., Digital Watermarking of Audio Signals Using a Psychoacoustic Auditory Model and Spread Spectrum Theory, *AES 107th Convention*, New York, NY, 1999.
- [29] Gluth R., Regular FFT-Related Transform Kernels for DCT/DST-Based Polyphase Filter Banks, *ICASSP 91*, pp. 2205-2208, 1991.

- [30] Gray R. M., Neuhoff D. L., Quantization, *IEEE Trans. on Information Theory*, vol. 44, no. 6, 1998.
- [31] Grill B., The MPEG-4 General Audio Coder, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [32] Hardman V., Sasse M. A., Handley M., Watson A., Reliable Audio for Use Over the Internet, *Proc. INET '95*, 1995.
- [33] Harma A., Vaalgamaa M., Laine U. K., A Warped Linear Predictive Stereo Codec Using Temporal Noise Shaping, *Proc. Nordic Signal Proc. Symposium, NORSIG'98*, pp. 229-232, Denmark, 1998.
- [34] Herre J., Temporal Noise Shaping, Quantization and Coding Methods in Perceptual Audio Coding: A Tutorial Introduction, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [35] Herre J., Brandenburg K., Eberlein E., Combined Stereo Coding, *AES 93th Convention*, San Francisco, CA, 1992.
- [36] Herre, J., Brandenburg K., Lederer D., Intensity Stereo Coding, *AES 96th Convention*, Amsterdam, NL, 1994.
- [37] Herre J., Johnston J. D., Enhancing the Performance of Perceptual Audio Coders by Using Temporal Noise Shaping (TNS), *AES 101st Convention*, Los Angeles, 1996.
- [38] Herre J., Johnston J. D., Exploiting Both Time and Frequency Structure in a System that Uses an Analysis/Synthesis Filterbank with High Frequency Resolution, *AES 103th Convention*, New York, NY, 1997.
- [39] Herre J., Johnston J. D., A Continuously Signal Adaptive Filterbank for High-Quality Perceptual Audio Coding, *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, Mohonk, 1997.
- [40] Herre J., Schulz D., Extending the MPEG-4 AAC Codec by Perceptual Noise Substitution, *AES 104th Convention*, Amsterdam, NL, 1998.
- [41] Hicks C. M., *Modeling of Multi-Channel Audio Signals*, PhD Thesis, University of Cambridge, Cambridge, UK, 1999.
- [42] Ingle V. K., Proakis J. G., *Digital Signal Processing Using MATLAB*, Brooks/Cole, 2000.
- [43] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 13818-7 Information Technology – Generic Coding of Moving Pictures and Associated Audio, Part 7: Advanced Audio Coding, 1997.
- [44] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Information Technology – Coding of Audio-Visual Objects, Part 3: Audio, 1999.

- [45] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Information Technology – Coding of Audio-Visual Objects, Part 3: Audio, AMENDMENT 1, 2001.
- [46] ISO/IEC JTC1/SC29/WG11 MPEG. International Standard IS 14496-3 Information Technology – Coding of Audio-Visual Objects, Part 3: Audio, TECHNICAL CORRIGENDUM 1, 2002.
- [47] ITU-R RECOMMENDATION BS.1387-1, *Methods for Objective Measurements of Perceived Audio Quality*, 1998.-2001.
- [48] ITU-R RECOMMENDATION BS.1116-1, *Methods for the Subjective Assessment of Small Impairments in Audio Systems Including Multichannel Sound Systems*, 1994.-1997.
- [49] Jelaković T., *Zvuk, Sluh, Arhitektonska Akustika*, Školska knjiga, Zagreb, 1978.
- [50] Johnson J. D., Transform coding of audio signals using perceptual noise criteria, *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 2, pp. 314-323, 1988.
- [51] Johnston J., Estimation of perceptual entropy using noise masking criteria. *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 2524-2527., 1992.
- [52] Johnston J., Transform coding of audio signals using perceptual noise criteria. *IEEE Journal on Selected Areas in Communications*, 1988.
- [53] Johnston J., Ferreira A., Sum-difference stereo transform coding. *Proc. ICAASP*, San Francisco, CA, pp. 569-571., 1992.
- [54] Kabal P., An Examination and Interpretation of ITU-R BS.1387: Perceptual Evaluation of Audio Quality, *TSP Lab Technical Report*, Dept. Electrical & Computer Engineering, McGill University, 2002.
- [55] Kahrs M., Brandenburg K., *Applications of Digital Signal Processing to Audio and Acoustics*, Kluwer Academic, 2001.
- [56] Karp T., Fliege N. J., Modified DFT Filter Banks with Perfect Reconstruction, *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 46, no. 11, 1999.
- [57] Kliewer J., On the Relationship between Pseudo-QMF Designs and Perfect-Reconstruction Solutions for Modulated Filter Banks, *Proc. 33rd ASILOMAR Conference on Signals, Systems and Computers*, pp. 1622-1626, Pacific Grove, CA, 1999.
- [58] Lame, <http://www.mp3dev.org/mp3>, 2003.
- [59] Lanciani C. A., Schafer R. W., Subband-Domain Filtering of MPEG Audio Signals, *IEEE ICASSP*, Arizona, 1999.

- [60] Lee K., Jeong N., Bang K., Youn D., A VLSI Implementation of MPEG-2 AAC Decoder System, *IEEE Trans. Consumer Electronics*, vol. 44, no. 3, pp. 872-877, 1998.
- [61] Levine S., *Audio Representation for Data Compression and Compressed Domain Processing*, PhD Thesis, Stanford University, 1998.
- [62] Levine S. N., Effects Processing on Audio Subband Data, *ICMC Proceedings*, 1996.
- [63] Levine S., Smith J.O., A sines+transients+noise audio representation for data compression and time/pitch-scale modifications, *AES 105th Convention*, San Francisco, CA, 1998.
- [64] Lie C. M., Lee W. J., Hong R. S., A New Criterion and Associated Bit Allocation Method for Current Audio Coding Standards, *Proc. of the 5th Int. Conference on DAFX*, 2002.
- [65] Liebchen T., Lossless Audio Coding using Adaptive Multichannel Prediction, *AES 113th Convention*, Los Angeles, CA, 2002.
- [66] Liebchen T., Purat M., Noll P., Lossless Transform Coding of Audio Signals, *AES 102th Convention*, Munich, Germany, 1997.
- [67] Lin Y., Vaidyanathan P. P., A Kaiser Window Approach for the Design of Prototype Filters of Cosine Modulated Filterbanks, *IEEE Signal Processing Letters*, vol. 5, no. 6, 1998.
- [68] Lincoln B., An Experimental High Fidelity Perceptual Audio Coder, Stanford University, <http://www-ccrma.stanford.edu/~jos/bosse/>, 1998.
- [69] Lyons R. G., *Understanding Digital Signal Processing*, Addison Wesley Longman, 1997.
- [70] Makhoul J., Linear Prediction: A Tutorial Review, *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561-580, 1975.
- [71] Malvar H., Lapped transforms for efficient transform/subband coding, *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 38, pp. 960-978, 1990.
- [72] Malvar H. S., Extended Lapped Transforms: Fast Algorithms and Applications, *IEEE Transactions on Signal Processing*, vol. 40, pp. 2703-2714, 1992.
- [73] Malvar H., A Modulated Complex Lapped Transform and Its Applications to Audio Processing, *IEEE ICASSP99*, 1999.
- [74] Malvar H. S., Biorthogonal and Nonuniform Lapped Transforms for Transform Coding with Reduced Blocking and Ringing Artifacts, *IEEE Transactions on Signal Processing*, pp. 1043-1053, 1998.

- [75] Masri P., *Computer Modelling of Sound Transformation and Synthesis of Musical Signals*, PhD Thesis, University of Bristol, Bristol, 1996.
- [76] Masri P., Bateman A., Improved Modelling of Attack Transients in Musical Analysis-Resynthesis, *Proc. 1996 Int. Computer Music Conference*, Hong Kong, 1996.
- [77] Masri P., Bateman A., Identification of Nonstationary Audio Signals Using the FFT, with Application to Analysis-based Synthesis of Sound, *Proc. IEE Colloquium on Audio Engineering*, pp.11/1 - 11/6, 1995.
- [78] Nikolajevič V., Fettweis G., New Recursive Algorithms for the Forward and Inverse MDCT, *Proc. IEEE Workshop on Signal Processing Systems (SiPS)*, Antwerp, Belgium, 2001.
- [79] Noll P., Liebchen T., Digital Audio: From Lossless to Transparent Coding, *Proceedings IEEE Signal Processing Workshop*, Poznan, pp. 53-60, 1999.
- [80] Oppenheim A. V., Schaffer R. W., *Discrete-Time Signal Processing*, Prentice-Hall, 1989.
- [81] Painter T., Spanias A., A Review of Algorithms for Perceptual Coding of Digital Audio Signals, <http://www.eas.asu.edu/~speech/ndct/dsp97.ps>, 1997.
- [82] Painter T., Spanias A., Perceptual Coding of Digital Audio, *Proc. IEEE*, vol. 88, no. 4, pp. 451-513, 2000.
- [83] Pan D. Y., Digital Audio Compression, *Digital Technical Journal*, vol. 5, no. 2, 1993.
- [84] Pan D., A tutorial on MPEG/audio compression, *IEEE Multimedia*, vol. 2, no. 2, pp. 60-74, 1995.
- [85] Pereira W., *Modifying LPC Parameter Dynamics to Improve Speech Coder Efficiency*, Master's thesis, McGill University, Montreal, Canada, 2001.
- [86] Pohlmann K. C., *Principles of Digital Audio*, 4th ed., McGraw-Hill, 2000.
- [87] Princen J., Bradley A., Analysis/synthesis filter bank design based on time domain aliasing cancellation, *IEEE Transaction on Acoustic Speech and Signal Processing*, vol. 34, no. 5, pp. 1153-1161, 1986.
- [88] Princen J., Johnson A. W., Bradley A. B., Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation, *Proc. IEEE ICASSP*, pp. 2161-2164, 1987.
- [89] Pulkki V., Evaluating Spatial Sound with Binaural Auditory Model, *Proc. International Computer Music Conference ICMC 2001*, Havana, Cuba, 2001.
- [90] Purnhagen H., An Overview of MPEG-4 Audio Version 2, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.

- [91] Purnhagen H., Advances in Parametric Audio Coding, *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 31-34, 1999.
- [92] Robinson D. J., Hawksford M., Time-Domain Auditory Model for the Assessment of High-Quality Coded Audio, *AES 107th Convention*, New York, NY, 1999.
- [93] Rothweiler J. H., Polyphase quadrature filters – a new subband coding technique, *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Boston, pp. 1280-1283, 1983.
- [94] Salomonsen K., Sogaard S., Larsen E. P., *Design and Implementation of an MPEG/Audio layer III Bitstream Processor*, Master's Thesis, Aalborg University, Aalborg, 1997.
- [95] Scheirer E. D., The MPEG-4 Structured Audio Standard, *Proc. IEEE ICASSP*, 1998.
- [96] Scheirer E. D., Ray L., Algorithmic and Wavetable Synthesis in the MPEG-4 Multimedia Standard, *AES 105th Convention*, San Francisco, CA, 1998.
- [97] Scheirer E. D., Structured Audio and Effects Processing in the MPEG-4 Multimedia Standard, *Multimedia Systems*, Springer-Verlag, 1999.
- [98] Scheirer E. D., Kim Y. E., Generalized Audio Coding with MPEG-4 Structured Audio, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.
- [99] Scheirer E. D., Vercoe B. L., SAOL: The MPEG-4 Structured Audio Orchestra Language, *Computer Music Journal*, 23:2, pp. 31-51, 1999.
- [100] Schroeder M. R., Models of Hearing, *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1332-1350, 1975.
- [101] Schulz D., Improving Audio Codecs by Noise Substitution, *Journal of the Audio Engineering Society*, vol. 44, no. 7/8, pp. 593-598, 1996.
- [102] Serra X., *A System For Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*, PhD Thesis, Stanford University, CA, 1989.
- [103] Serra X., Musical Sound Modeling with Sinusoids plus Noise, *Musical Signal Processing*, Swets&Zeitlinger, 1997.
- [104] Soulodre G., Grusec T., Lavoine M., Thibault L., Subjective evaluation of state-of-the-art two-channel audio coders, *Journal of Audio Eng. Soc.* 46(3):164-177, 1998.
- [105] Soulodre G. A., Lavoie M. C., Subjective Evaluation of Large and Small Impairments in Audio Codecs, *AES 17th International Conference on High Quality Audio Coding*, Florence, Italy, 1999.

- [106] Spanias A., Multimedia Signal Processing, Perceptual Audio Coding, Audio Lecture 5, <http://www.eas.asu.edu/~spanias/>, 2002.
- [107] Spatial Hearing, *NCTU/CSIE/DSP Lab.*, C. M. Liu, 1996.
- [108] Sporer T., Brandenburg K., Edler B., The Use of Multirate Filter Banks for Coding of High Quality Digital Audio, *EUSIPCO 6th Conference*, vol. 1, pp. 211-214, 1992.
- [109] Srinivasan P., Jamieson L. H., High Quality Audio Compression Using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling, *IEEE Transactions on Signal Processing*, 1999.
- [110] Stoll G., Kozamernik F., EBU Listening Tests on Internet Audio Codecs, *EBU Technical Review*, 2000.
- [111] Stuart J. R., Perceptual Issues in Multi-Channel Environments, *AES 105th Convention*, San Francisco, CA, 1994.
- [112] Touimi A. B., A Generic Framework for Filtering in Subband-Domain, *DSP2000*, Hunt, Texas, 2000.
- [113] Vanhouche V., Block Artifacts Cancellation in DCT Based Image Compression, <http://www.stanford.edu/~nouk/mdct/>, 2001.
- [114] Varho S., Predictive Coding, <http://www.physics.utu.fi>, 2002.
- [115] Verma T. S., *A Perceptually Based Audio Signal Model with Application to Scalable Audio Compression*, PhD thesis, Stanford University, Stanford, 1999.
- [116] Verma T. S., Meng T. H. Y., Extending Spectral modeling Synthesis with Transient Modeling Synthesis, *Computer Music Journal*, vol. 24, no. 2, pp. 47-59, 2000.
- [117] Verma T. S., Levine S. N., Meng T. H. Y., Transient Modeling Synthesis: a flexible analysis/synthesis tool for transient signals, *Proc. ICMC*, pp. 164-167, 1997.
- [118] Verma T. S., Meng T. H. Y., Time Scale Modifications Using a Sines + Transients + Noise Signal Model, *Proceedings of the Digital Audio Effects Workshop DAFX'98*, pp. 49-52, Barcelona, Spain, 1998.
- [119] Waal R. G. van der, Veldhuis R. N. J., Subband Coding of Stereophonic Digital Audio Signals, *IEEE ICAASP*, pp. 3601-3604, 1991.
- [120] Wang Y., Vilermo M., An Excitation Level Based Psychoacoustic Model for Audio Compression, *Nokia Research Center*, Tampere, Finland, 1999.
- [121] Wang Y., Yaroslavsky L., Vilermo M., Vaananen M., Some Peculiar Properties of MDCT, *Proceedings of ISCP2000*, 2000.
- [122] West J. R., *Five-Channel Panning Laws: An Analytical and Experimental Comparison*, Master's Thesis, University of Miami, Miami, FL, 1998.

- [123] Yang D., Ai H., Kyriakakis C., Kuo J., Design of Progressive Syntax-Rich Multichannel Audio Codec, *Proc. SPIE*, vol. 4674, pp. 121-123, San Jose, CA, 2002.
- [124] Yin L., Suonio M., Vaananen M., A New Backward Predictor for MPEG Audio Coding, *AES 103th Convention*, New York, NY, 1997.
- [125] Domazet D., Kovač M., "Advanced Software Implementation of MPEG-4 AAC Audio Encoder", 4th EURASIP Conference, July, 2003.

Sažetak

Osnovni cilj magistarskog rada je programska izvedba MPEG-4 audio koda. Osim same izvedbe koda, rad nastoji dati i zaokruženi uvod u cjelokupno područje percepcijskog audio kodiranja, sa naglaskom na MPEG-4 standardu.

Magistarski rad se sastoji od deset poglavlja, a ugrubo se može podijeliti na tri cjeline. Prva cjelina daje općeniti uvod u audio kodiranje, druga cjelina opisuje MPEG-4 audio standard, dok treća cjelina opisuje konkretnu izvedbu audio koda.

Prvo poglavlje daje uvod u stanje multimedijalnih tehnologija danas, sa naglaskom na audio kodiranju. Drugo poglavlje ukratko opisuje ljudski slušni sistem i služi kao uvod u treće poglavlje koje opisuje psihoakustičke principe na kojima audio kodiranje počiva. Nakon upoznavanja sa glavnim principima psihoakustike, u četvrtom poglavlju je opisana tipična arhitektura audio koda koju slijede gotovo svi audio kodi danas. U istom poglavlju je dan i sažetak povijesnog razvoja audio koda. Peto poglavlje daje općeniti opis danas najsofisticiranijeg multimedijalnog standarda ISO 14496 (MPEG-4). U šestom poglavlju počinje opis konkretne programske izvedbe audio koda prema dijelu MPEG-4 standarda koji se odnosi na kodiranje širokopojasnog, složenog audio signala (MPEG-4 General Audio). Audio koder je nazvan RAAC prema *RASIP Advanced Audio Coding*. Sedmo poglavlje opisuje psihoakustički model izveden u RAAC koderu, dok osmo poglavlje opisuje posebne blokove za obradu spektra izvedene u RAAC koderu koji su presudni za kvalitetu koda. Deveto poglavlje je posvećeno subjektivnoj i objektivnoj analizi i evaluaciji izvedenog koda, dok je u zadnjem desetom poglavlju dan zaključak.

Ključne riječi:

MPEG-4, AAC, mp3, mp4, audio kodiranje, audio kompresija, multimedija, psihoakustika.

Abstract

"Software Implementation of MPEG-4 General Audio Encoder"

Master's thesis main goal is MPEG-4 audio encoder software implementation. However, the thesis also aims to give an overall introduction to perceptual audio encoding, with an emphasis on MPEG-4 standard.

The thesis consists of ten chapters, generally divided into three groups. First group gives overall introduction to audio coding, second group describes MPEG-4 audio standard, and the third group describes actual audio encoder implementation.

Chapter 1 gives a short introduction to multimedia technologies, with an emphasis on audio coding. Chapter 2 briefly describes human auditory system and serves as an introduction to Chapter 3, which describes psychoacoustic principles that, in fact, represent fundamentals of audio coding itself. After explaining the basic psychoacoustic principles, Chapter 4 describes a typical audio encoder architecture that is followed by nearly all audio encoders now days. Same chapter gives a concise description of historical audio encoder development. Chapter 5 gives overall introduction to the most sophisticated multimedia standard today – ISO 14496 (MPEG-4). Chapter 6 starts the description of the concrete software implementation of the audio encoder according to MPEG-4 General Audio – a part of MPEG-4 standard that relates to high bandwidth, complex audio signal encoding. Audio encoder is called RAAC according to *RASIP Advanced Audio Coding*. Chapter 7 describes psychoacoustic model implemented in RAAC encoder, while Chapter 8 describes main MPEG-4 audio blocks also implemented in RAAC encoder. Chapter 9 is devoted to subjective and objective analysis and evaluation of RAAC implementation, while Chapter 10 gives the overall conclusion.

Key words:

MPEG-4, AAC, mp3, mp4, advanced audio coding, audio encoding, audio compression, multimedia, psychoacoustics.

Životopis

Rođen sam 26.02.1973. godine u Banjaluci u Bosni i Hercegovini, gdje sam završio osnovnu i prva tri razreda srednje škole (Elektro Škola, Banjaluka). Četvrti razred srednje škole završio sam 1992. godine u Sjedinjenim Američkim Državama (East Wake High School, North Carolina). Nakon srednje škole upisao sam Fakultet Elektrotehnike i Računarstva na Sveučilištu u Zagrebu (smjer Industrijska Elektronika), koji sam završio 1998. godine diplomiravši na Zavodu za Automatiku i Procesno Računarstvo (grupa Računarski Sustavi i Procesi). Nakon završetka studija, paralelno sa zaposlenjem (RIZ Transmitters, Zagreb), nastavio sam rad na Fakultetu Elektrotehnike i Računarstva gdje u sklopu magistarskog studija radim na razvoju i implementaciji multimedijalnih tehnologija pod vodstvom Prof. dr. sc. Maria Kovača.

