



**UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA**



**ANALIZA I METODE ZA POBOLJŠANJE TAČNOSTI U
TRANSFORMACIONOM KODOVANJU**

- Magistarski rad -

Mentor:
Prof. dr Vladimir Kovačević

Kandidat:
Dipl. ing. Željko Lukač

Novi Sad, Januar 2004.

UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KLJUČNA DOKUMENTACIJSKA INFORMACIJA

Redni broj:

RBR

Identifikacioni broj:

IBR

Tip dokumentacije:

TD

Monografski rad

Tip zapisa:

TZ

Štampa

Vrsta rada:

VR

Magistarski rad

Autor:

AU

Željko Lukač

Mentor/Komentor

MN

prof. dr Vladimir Kovačević

Naslov rada:

NR

Analiza i metode za poboljšanje tačnosti u transformacionom kodovanju

Jezik publikacije:

JP

srpski

Jezik izvoda:

JI

sprski

Zemlja publikovanja:

ZP

Srbija i Crna Gora

Uže geografsko područje:

UGP

Vojvodina

Godina:

GO

2004.

Izdavač:

IZ

Fakultet Tehničkih Nauka

Mesto i adresa:

MA

21000 Novi Sad, Trg Dositeja Obradovića 6

Fizički opis rada: 7 / 70 / 17 / 30 / 41 / 0 / 0
(poglavlja/strana/citata/tabela
/slika/grafika/priloga)
FO

Naučna oblast: Elektrotehnika
NO

Naučna disciplina: Računarska tehnika
ND

Predmetna odrednica/ključne reči: Računarska tehnika / transformaciono kodovanje,
tačnost izračunavanja, konačna dužina riječi,
digitalni signal procesori
PO
UDK 681.32 (043.2)

Čuva se: U biblioteci Fakulteta Tehničkih Nauka
ČU

Važna napomena:
VN

Izvod: U okviru rada izvršena je analiza grešaka nastalih
izračunavanjem transformacija poznatim metodama
i predložena poboljšana metoda

Datum prihvatanja teme
Od strane NN veća: 11. april 2002. godine
DP

Datum odbrane: 10. mart 2004. godine
DO

Članovi komisije:
KO

Predsednik: prof. dr Borivoj Lazić
Član: prof. dr Vladimir Kovačević
Član: prof. dr Miodrag Temerinac
Član: prof. dr Miroslav Popović
Član: docent dr Nikola Teslić

UNIVERSITY OF NOVI SAD
FACULTY OF TECHNICAL SCIENCES
KEY WORDS DOCUMENTATION

Accession number:
ANO

Identification number:
INO

Document type *Monographic*
DT

Type of record: *Printed*
TR

Contents code: *Master thesis*
CC

Author: *Željko Lukač*
AU

Mentor /Co mentor *Vladimir Kovačević, Ph D*
MN

Title: *Analysis and methods for improvement of accuracy
in transform coding*
TI

Language of text: *Serbian*
LT

Language of abstract: *Serbian*
LA

Country of publication: *Serbia and Montenegro*
CP

Locality of publication: *Vojvodina*
LP

Publication year: *2004.*
PY

Publisher: *Faculty of Technical Sciences*
PU

Publication place: *21000 Novi Sad, Trg Dositeja Obradovića 5*
PP

Physical description: 7 / 70 / 17 / 30 / 41 / 0 / 0
(chapters/pages/ref./tables/
pictures/graphs/appendixes)
PD

Scientific field: Electrical Engineering
SF

Scientific discipline: Computer engineering
SD

Subject/Key words: Computer engineering / transform coding, accuracy,
fixed word lenght, digital signal processor
SKW

UC

Holding data: The Library of the Faculty of Technical Sciences
HD

Note:
N

Abstract: Analysis of accuracy of published methods for
transform calculation and proposal for their
improvement.
A

Accepted by the Scientific Board on: 11. April 2002.
ASB

Defended on: 10. March 2004.
DE

Thesis defends board:
DB

President: Borivoj Lazić, Ph D
Member: Vladimir Kovačević, Ph D
Member: Miodrag Temerinac, Ph D
Member: Miroslav Popović, Ph D
Member: Nikola Teslić, Ph D

SKRAĆENICE

AAC	- standard za kodovanje zvuka
AC3	- standard za kodovanje zvuka
ATRAC	- standard za kodovanje zvuka
CD	- kompakt disk (eng. <i>Compact Disc</i>)
DAB	- standard za digitalni radio (eng. <i>Digital Audio Broadcasting</i>)
DAT	- digitalna audio kaset (eng. <i>Digital Audio Tape</i>)
DCT	- diskretna kosinusna transformacija (eng. <i>Discrete Cosine Transform</i>)
DIT	- prorjeđivanje u vremenskom domenu (eng. <i>Decimation in Time</i>)
DIF	- prorjeđivanje u frekventnom domenu (eng. <i>Decimation in Frequency</i>)
DRM	- standard za digitalni radio (eng. <i>Digital Radio Mondiale</i>)
DVB	- standard za digitalnu televiziju (eng. <i>Digital Video Broadcasting</i>)
DVD	- digitalni disk opšte namijene (eng. <i>Digital Versatile Disc</i>)
DSP	- digitalni signal procesor (eng. <i>Digital Signal Procesor</i>)
FFT	- brza Furijeova transformacija (eng. <i>Fast Fourier Transform</i>)
ISO	- internacionalna organizacija za standarde (International Standard Organization)
KBD	- vrsta prozorske funkcije (eng. <i>Kaiser-Bessel Derived</i>)
MAS	- oznaka familije digitalnih signal procesora firme Micronas
MDCT	- modifikovana diskretna kosinusna transformacija
MMC	- vrsta memorijske kartice (eng. <i>Multimedia Memory Card</i>)
MPEG	- standard za kodovanje zvuka i slike (eng. <i>Moving Pictures Experts Group</i>)
OBT	- ortogonalna blok transformacija (eng. <i>Orthogonal Block Transform</i>)
OFDM	- multipleksiranje sa ortogonalnom podjelom u frekventnom domenu (eng. <i>Orthogonal Frequency Division Multipleks</i>)
PAC	- standard za kodovanje zvuka
PCM	- impulsno kodna modulacija (eng. <i>Pulse Code Modulation</i>)
PR	- potpuna rekonstrukcija (eng. <i>Perfect Reconstruction</i>)
SD	- vrsta memorijske kartice (eng. <i>Smart Digital</i>)
SMC	- vrsta memorijske kartice (eng. <i>Smart Memory Card</i>)

1. UVOD	1
1.1 DEFINICIJA TRANSFORMACIJA SA PREKLAPANJEM I NJIHOVA PRIMJENA U OBRADI AUDIO SIGNALA	2
1.1.1 Kratak prikaz	2
1.1.2 Direktna i inverzna transformacija	4
1.1.3 Primjeri prozorskih funkcija	5
1.1.4 Algoritam za izračunavanje	7
1.2 STANDARDI ZA KOMPRESIJU AUDIO SIGNALA ZASNOVANI NA ORTOGONALNIM TRANSFORMACIJAMA SA PREKLAPANJEM	8
1.3 IZRAČUNAVANJE TRANSFORMACIJA SA PREKLAPANJEM KORIŠTENJEM BRZIH ALGORITAMA BAZIRANIH NA BRZOJ FURIJEVOJ TRANSFORMACIJI	12
1.4 PROBLEMATIKA PRIMJENE TRANSFORMACIJA SA PREKLAPANJEM KORIŠTENJEM DIGITALNIH SIGNALNIH PROCESORA	14
2. TAČNOST IZRAČUNAVANJA BRZIH ALGORITAMA NA DSP	17
2.1. GREŠKE KVANTIZACIJE	17
2.2 GREŠKA KVANTIZACIJE KOD DIREKTOG RAČUNANJA DISKRETNE FURIJEVE TRANSFORMACIJE (DFT)	19
2.3 GREŠKA KVANTIZACIJE KOD RAČUNANJA BRZE FURIJEVE TRANSFORMACIJE	21
2.4 PROSTIRANJE GREŠKE KVANTIZACIJE	27
3. POSTOJEĆE METODE ZA SMANJENJE GREŠKE UNESENE PRILIKOM IZRAČUNAVANJA BRZE FURIJEVE TRANSFORMACIJE KORIŠTENJEM ARITMETIKE NEPOKRETNOG ZAREZA I ANALIZA NJIHOVE TAČNOSTI	28
3.1 KRATAK PRIKAZ POSTOJEĆIH METODA	28
3.1.1 Implementacija FFT algoritma automatskim skaliranjem	28
3.1.2. Metode zasnovane na blok pokretnom zarezu	28
3.1.2.1 Dijeljenje sa dva ako postoji mogućnost prekoračenja u narednom stepenu FFT-a	29
3.1.2.2 Dijeljenje sa dva ako postoji mogućnost prekoračenja u narednom stepenu FFT-a i izbjegavanje prekoračenja kod kompleksnog množenja skaliranjem unutar baterflaja	30
3.1.2.3 Automatsko dijeljenje sa dva izlaznih vrijednosti i množenje ulaznih vrijednosti sa dva ako ne postoji mogućnost prekoračenja u narednom stepenu FFT-a	30
3.1.2.3 Automatsko dijeljenje sa dva izlaznih vrijednosti i množenje ulaznih vrijednosti sa dva ako ne postoji mogućnost prekoračenja u narednom stepenu FFT-a uz izbjegavanje prekoračenja dijeljenjem sa dva rezultata množenja ukoliko postoji mogućnost prekoračenja kod izvođenja kompleksnog množenja	31
3.2 KRITERIJUMI TAČNOSTI IZRAČUNAVANJA	31
3.3 ANALIZA TAČNOSTI POSTOJEĆIH METODA NA PRIMJERIMA NEKIH KARAKTERISTIČNIH SIGNALA	32
4. PRIJEDLOG NOVE METODE ZA SMANJENJE GREŠKE UNESENE PRILIKOM IZRAČUNAVANJA BRZE FURIJEVE TRANSFORMACIJE KORIŠTENJEM ARITMETIKE NEPOKRETNOG ZAREZA	39
4.1 OPIS METODE	39
4.1.1 Izbjegavanje prekoračenja kod sabiranja	39
4.1.2 Izbjegavanje prekoračenja kod kompleksnog množenja	40
4.1.2.1 Izbjegavanje prekoračenja kod kompleksnog množenja dodatnim upisom u memoriju	40
4.1.2.2 Izbjegavanje prekoračenja kod kompleksnog množenja skaliranjem unutar baterflaja	40
4.2 ANALIZA GREŠKE KVANTIZACIJE U DSP APLIKACIJI	40
4.3 POREĐENJE SA DRUGIM METODAMA	46

5. DSP PRIMJENA NOVE METODE	52
5.1 PREPORUKE I MOGUĆI PROBLEMI U ZAVISNOSTI OD RASPOLOŽIVIH RESURSA	52
5.2 PRIMER KONKRETNE PRIMENE – AC3 DEKODER	54
5.2.1 <i>Opis problema</i>	54
5.2.1.1 <i>AC3 algoritam</i>	54
5.2.1.1.1 <i>Filter banka</i>	56
5.2.1.1.2 <i>Spektralna anvelopa</i>	57
5.2.1.1.3 <i>Dodela bita</i>	58
5.2.1.1.4 <i>Sprezanje kanala</i>	58
5.2.1.1.5 <i>Kontrola nivoa glasnosti</i>	59
5.2.1.2 <i>Način provere AC3 dekodera</i>	60
5.2.2 <i>Realizovano rješenje</i>	61
5.2.3 <i>Dobijeni rezultati</i>	62
6. ZAKLJUČAK	68
7. LITERATURA	69

1.Uvod

Osnovni cilj istraživanja opisanog u ovom radu je pronalaženje efikasnog algoritma za brzo izračunavanje transformacije (FFT, DCT, MDCT) uz zadovoljenje slijedećih zahtjeva:

- kvalitet rezultata izračunavanja transformacionih koeficijenata koji je dat ili kroz prosečno odstupanje u odnosu na proračun sa neograničenom tačnošću (odnos signal-kvantizacioni šum) ili kroz maksimalno dozvoljeno odstupanje (broj tačnih bita).
- zadati DSP resursi (dužina riječi, raspoloživa memorija i brzina obrade)

Algoritam je potrebno verifikovati kroz računarsku simulaciju i implementaciju na MAS procesoru.

Osnovni koncept se zasniva na detaljnoj analizi postojećih metoda i rezultira u sledećim tezama:

- potrebno je kroz analizu prostiranja greške ustanoviti minimalno potrebno povećanje dužine reči u svakom stepenu
- rezultati operacija množenja i sabiranja u svakom stepenu moraju se izmjeniti tako da omogućuju minimalnu grešku i sprečavaju eventualno prekoračenje opsega
- broj dodatnih kontrolnih operacija potrebnih za definisane izmjene mora se minimizirati
- kompromis između zahtevane brzine obrade i memorijskih resursa mora biti prilagodiv ciljnoj aplikaciji i DSP platformi

Kompresija audio signala je trend koji se brzo širi zahvaljujući novim medijima (Internet, memorijske karte (MMC, SD, SMC), DVD, DVB, DAB, DRM, multimedijalne aplikacije u 3G mobilnoj telefoniji). Ona se zasniva na spektralnoj analizi visoke rezolucije audio signala i primeni psihoakustičkih modela.

Spektralnu analizu visoke rezolucije je moguće ostvariti samo primenom filter banki (transformaciono kodovanje) sa velikom dužinom filtera. Sličan zahtev se javlja i u nekim sistemima za prenos digitalnih signala sa više nosilaca (OFDM). Ove metode se moraju efikasno implementirati na DSP platformama, koje imaju vrlo ograničene resurse (memorija, brzina obrade, dužina reči). Efikasna implementacija filter banki je moguća samo sa brzim algoritmima koji uzastopno izračunavaju zahtevanu rezoluciju ali se pri tome javlja problem tačnosti. Pri uzastopnom računanju se javlja potreba za dužom reči u svakom sledećem stepenu da bi se očuvala tačnost. Algoritamska dekompozicija i analiza prostiranja greške su jedini put da se raspoloživi DSP resursi efikasno iskoriste.

1.1 Definicija transformacija sa preklapanjem i njihova primjena u obradi audio signala

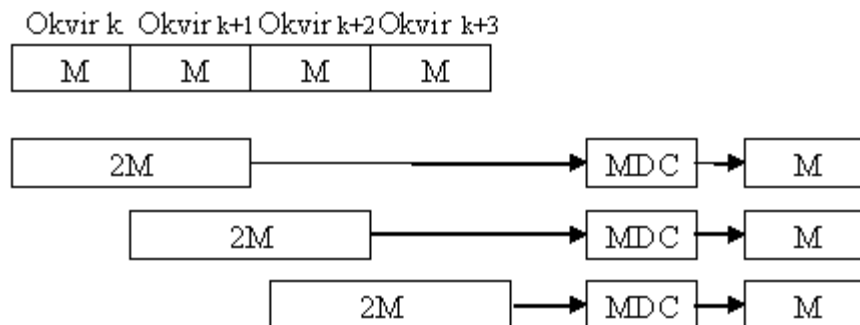
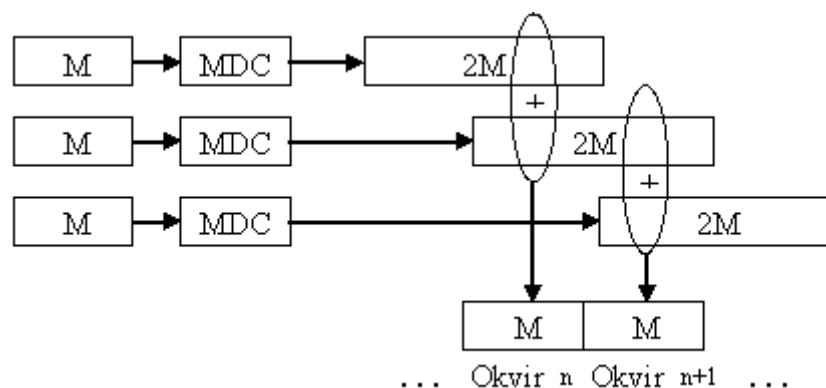
1.1.1 Kratak prikaz

Većina savremenih algoritama za kompresiju audio signala je zasnovana na transformacionom kodovanju (TF,TC) i/ili kodovanju sa podjelom na podopsege (engl. subband coding) koji predstavljaju efikasne metode za kompresiju podataka. Kod oba metoda se koristi korelacija među odbircima ulaznog signala u cilju smanjenja bitske brzine. Oba metoda sadrže blok za analizu (transformacija ili filter banka), kvantizaciju i kodovanje na strani predajnika, i dekodovanje i blok za sintezu (inverzna transformacija ili filter banka) na strani prijemnika. U bloku za analizu obavlja se dekompozicija ulaznog signala na komponente (koeficijenti transformacije ili signali po podopsezima) koje treba dekorelisati. Dekorelacija obezbjeđuje dobro sabijanje energije što je uslov za efikasno smanjenje bitske brzine.

Tehnike transformacionog kodovanja su blokovski orijentisane, blok od L ulaznih odbiraka se transformiše u izlazni blok od M koeficijenata. Raniji metodi su se zasnivali na kvadratnim ($L=M$) ortogonalnim transformacijama koje mogu da izazovu izobličenja koja se nazivaju blok efekti (engl. “blocking effects”). U cilju smanjenja posljedica preklapanja spektra koriste se ortogonalne transformacije sa preklapanjem (engl. “Lapped orthogonal transforms” - LOT) koje se ponekad nazivaju i ortogonalne blok transformacije (engl. “orthogonal block transforms” - OBT) kod kojih je $L>M$ [1][2][3].

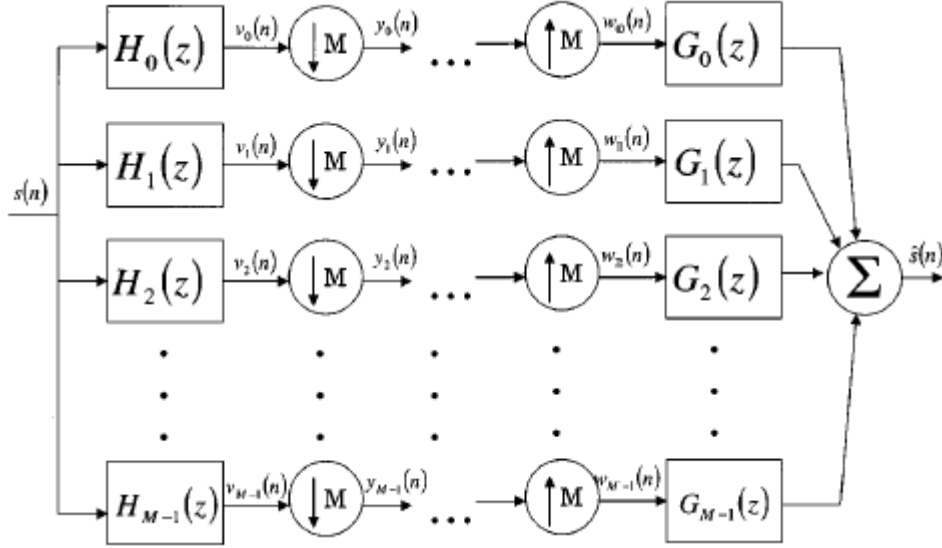
Tehnike kodovanja podjelom na podopsege se zasnivaju na dekompoziciji signala korištenjem filter banke sa M filtera propusnika opsega dužine L i decimaciji [4]. Uglavnom se koriste uniformne filter banke sa kritičnim odabiranjem.

Primjer ortogonalne transformacije sa preklapanjem sa kritičnim odabiranjem je prikazan na slici 1.1. Na slici je prikazan specijalan slučaj ortogonalne transformacije sa preklapanjem kod kojeg je $L = 2M$ koja se naziva modifikovana diskretna kosinusna transformacija. U procesu analize (slika 1.1. a) transformiše se ulazni signal korištenjem uzastopnih operacija transformacije. U svakoj operaciji transformacije korištena transformacija (MDCT u ovom slučaju) mapira blok od L ulaznih odbiraka u M koeficijanata transformacije. S obzirom da se u svakoj transformaciji generiše M novih koeficijenata od L ulaznih koeficijenata brzina generisanja transformacionih koeficijenata je jednaka brzini odabiranja ulaznih odbiraka. Pomenuta osobina se naziva kritično odabiranje [9]. Inverzna ortogonalna transformacija sa preklapanjem (u ovom slučaju IMDCT) mapira M koeficijenata na blok od L vrijednosti korištenjem operacije “preklopi i saberi” (engl. “overlap and add”).

Slika 1.1a Proces analize ortogonalne filter banke sa preklapanjem za $L=2$ Slika 1.1b Proces sinteze ortogonalne filter banke sa preklapanjem za $L=2$

Na slici 1.2 je prikazana filter banka za kodovanje podjelom na podopsege sa kritičnim odabiranjem. Na lijevoj strani slike je prikazan blok za analizu u kom se filtrira ulazni signal $s(n)$ korištenjem M filtera za analizu $H_i(z)$ i decimacija za faktor M dajući signal podopsega $y_i(n)$, tako da se ukupna brzina odabiranja ne mijenja ovim procesom, a na desnoj strani slike je prikazan blok za sintezu u kojem se obavlja interpolacija za faktor M i filtriranje sa M filtera za sintezu $G_i(z)$ čime se obavlja rekonstrukcija početna brzina odabiranja i pun propusni opseg. Sabiranjem izlaznih signala svih grana dobija se izlazni signal $\hat{s}(n)$.

Dualnost transformacija sa preklapanjem i filter banaka je pokazana u [1],[2],[9]. Pokazano je da su direktna transformacija, inverzna transformacija, filter banka za analizu i filter banka za sintezu samo različiti oblici za opisivanje iste operacije kodovanja i dekodovanja (L odbiraka u M koeficijenata).



Slika 1.2 Filter banka za kodovanje podjelom na podopsege

1.1.2 Direktna i inverzna transformacija

Ako se definiše impulsni odziv filtera za analizu $h_n(k)$ sa

$$h_n(k) = w(k) \sqrt{\frac{2}{M}} \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M+1}{2}\right)\right) \quad (1)$$

tada se transformaciju sa preklapanjem koja mapira 2LM ulaznih odbiraka $x(k)$ u M izlaznih koeficijenata $y(n)$ može definisati sa:

$$y(n) = \sum_{k=0}^{2LM-1} x(k) h_n(k) \quad (2)$$

što je ekvivalentno sa

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{2LM-1} x(k) \times w(k) \times \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M+1}{2}\right)\right) \quad n=0, \dots, M-1 \quad (3)$$

gdje je $w(k)$ vremenski prozor dužine 2LM.

Na sličan način se može definisati inverzna transformaciju:

$$x'(k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} y(n) \times w(k) \times \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M+1}{2}\right)\right) \quad k=0, \dots, 2LM-1 \quad (4)$$

Inverzna transformacija (4) ne omogućava rekonstrukciju ulaznih odbiraka za bilo koju prozorsku funkciju $w(k)$ jer transformacija veličine $M \times 2LM$ i njoj inverzna veličine $2LM \times M$ nisu kvadratne i ne mogu biti ortogonalne. Primjenom operacije “preklopi i saberi” moguće je postići potpunu rekonstrukciju ako prozorska funkcija $w(k)$ ispunjava uslov za potpunu rekonstrukciju.

Primjenom direktne transformacije na preklapljenje blokove ulaznih odbiraka $s(k)$:

$$X(m,k)=s(mM+k) \quad k=0,...,2*LM-1 \quad m=..., -2, -1, 0, 1, 2, ... \quad (5)$$

dobijamo koeficijente:

$$y(m,n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{2LM-1} x(m,k) \times w(k) \times \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M-1}{2}\right)\right) \quad n=0,...,M-1 \quad (6)$$

Ovaj proces predstavlja filter banku za analizu (sl. 1.1.a).

Primjena inverzne transformacije daje odbirke sa preklapanjem spektra (engl. "aliasing").

$$x'(m,k) = \sqrt{\frac{2}{M}} \sum_{n=0}^{M-1} y(m,n) \times w(k) \times \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M-1}{2}\right)\right) \quad k=0,...,2LM-1 \quad (7)$$

Zatim je potrebno primjeniti operaciju “preklopi i saberi” koja omogućava potpunu rekonstrukciju pod uslovom da prozorska funkcija zadovoljava uslov potpune rekonstrukcije:

$$s(mM + p) = \sum_{q=0}^{2L-1} x'(m - q, qM + p) \quad (8)$$

Ovaj proces predstavlja filter banku za sintezu (sl. 1.1.b).

Uslov potpune rekonstrukcije i njegovo opšte rješenje se mogu naći u [1],[2].

Ortogonalna transformacija sa preklapanjem koja zadovoljava uslov potpune rekonstrukcije i za koju vrijedi $L=2M$ se naziva modifikovana diskretna kosinusna transformacija –MDCT (koristi se i naziv TDAC od engleskog “time domain aliasing cancellation” [9]).

1.1.3 Primjeri prozorskih funkcija

Najjednostavnija prozorska funkcija za $L=1$ je kvadratna prozorska funkcija:

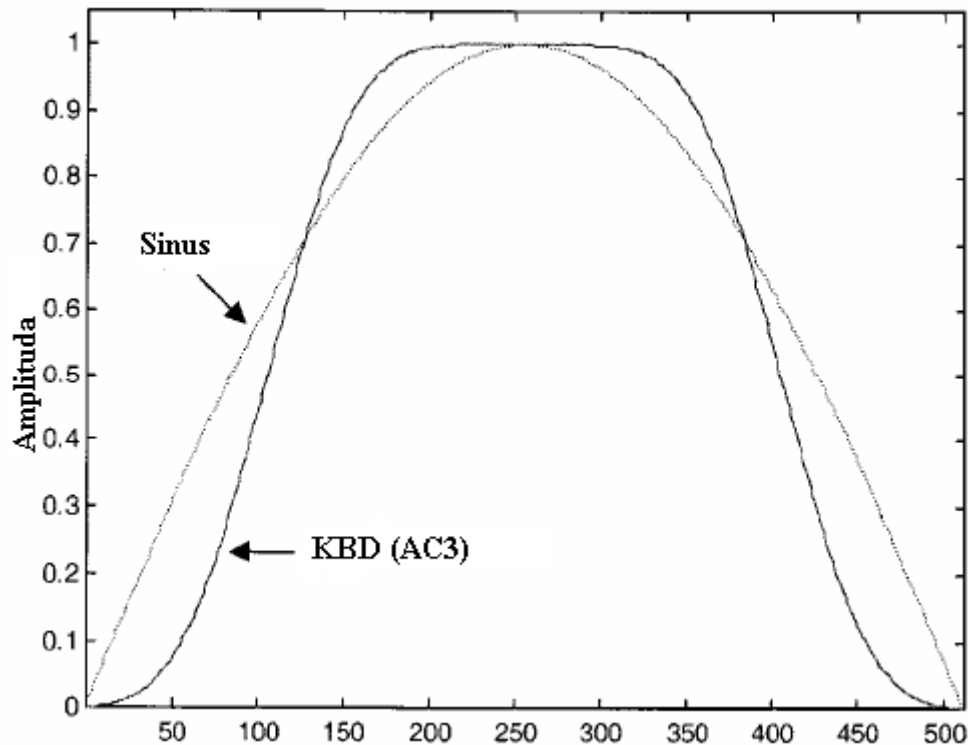
$$w(k) = \text{Sqrt}\left(\frac{1}{2}\right) \quad k=0,...,2M-1 \quad (9)$$

U algoritmima za obradu audio signala koji se zasnivaju na modifikovanoj kosinusnoj transformaciji često se koristi sinusna prozorska funkcija (10) (slika 1.3)

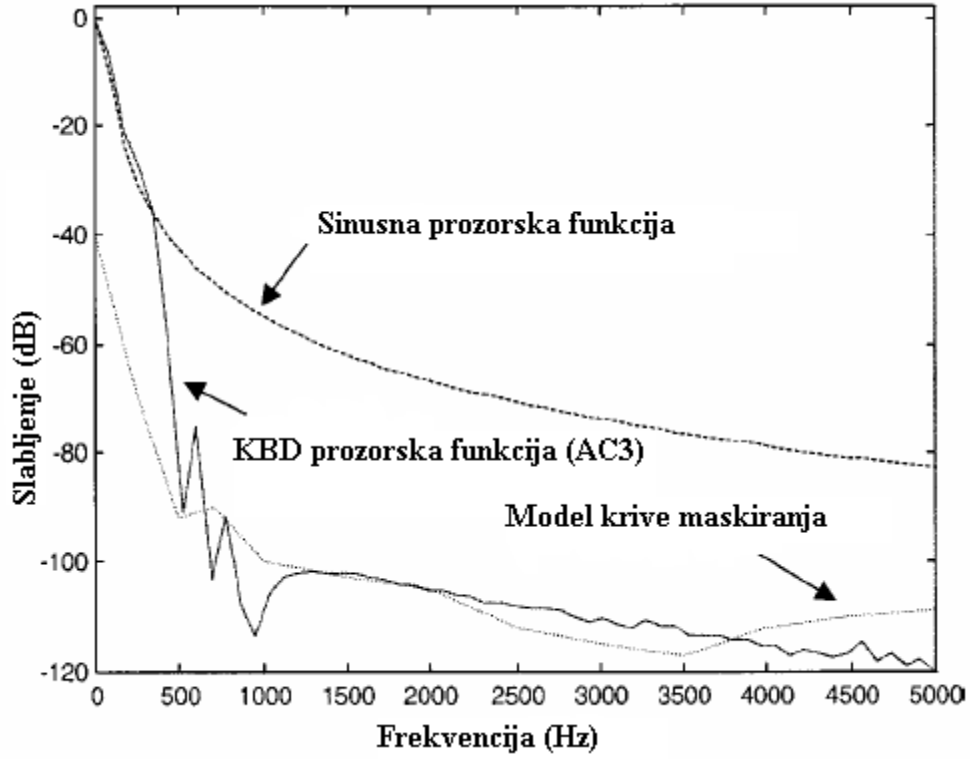
$$w(k) = \sin\left(\frac{\pi}{2M}\left(k + \frac{1}{2}\right)\right) \quad k=0,\dots,2M-1$$

Sinusna prozorska funkcija posjeduje nekoliko osobina koje je izdvajaju u odnosu na druge prozorske funkcije. Između ostalog, energija jednosmjerne komponente je sadržana u jednom koeficijentu, podopsezi filter banke imaju slabljenje od 24 dB i velik dobitak kodovanja. Sinusna prozorska funkcija se koristi u MPEG1 Layer 3 (mp3) i AAC algoritmima.

Druga često korištena prozorska funkcija je Kajzer-Bezel izvedena prozorska funkcija (KBD) (slika 1.3) koja je dobijena procedurom razvijenom u Dolbi laboratorijama korištenjem perceptualnih kriterija. Pomenuta funkcija obezbjeđuje mnogo veće slabljenje u nepropusnom opsegu nego sinusna prozorska funkcija (slika 1.4). Za čist ton smješten u centru nekog podopsega modifikovane diskretne kosinusne transformacije KBD filter banka omogućuje koncentrisanje više energije u jedan koeficijent transformacije što je čini pogodnom kod signala sa podjednako raspoređenim tonalnim komponentama. Ova prozorska funkcija se koristi u AC3 i AAC algoritmima.



Slika 1.3 Sinusna i KBD (AC3) prozorska funkcija



Slika 1.4 Karakteristike sinusne i KBD (AC3) prozorske funkcije

1.1.4 Algoritam za izračunavanje

Algoritam za izračunavanje direktne transformacije (3) može da se opiše u nekoliko koraka.

U prvom koraku ulazni signal se prozorira odabranom prozorskom funkcijom:

$$x_1(k) = w(k) \times x(k) \quad k=0, \dots, 2LM-1 \quad (9)$$

te dobijamo:

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{2LM-1} x_1(k) \times \cos\left(\frac{\pi}{M} \left(n + \frac{1}{2}\right) \left(k + \frac{M+1}{2}\right)\right) \quad n=0, \dots, M-1 \quad (10)$$

U drugom koraku se obavlja decimacija za faktor L:

$$x_2(k) = \sum_{p=0}^{L-1} (-1)^p \times x_1(2pM + k) \quad (11)$$

što daje:

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{2M-1} x_2(k) \times \cos\left(\frac{\pi}{M}\left(n + \frac{1}{2}\right)\left(k + \frac{M-1}{2}\right)\right) \quad n=0,...,M-1 \quad (12)$$

Množenje sa -1 je moguće izvesti i prilikom prozoriranja

U trećem koraku se obavlja decimacija sa faktorom dva na slijedeći način:

$$x_3(k) = x_2\left(\frac{M}{2} - 1 - k\right) + x_2\left(\frac{M}{2} + k\right) \quad k = 0, ..., \frac{M}{2} - 1 \quad (13)$$

$$x_3(k) = x_2\left(\frac{M}{2} + k\right) - x_2\left(\frac{5M}{2} - 1 - k\right) \quad k = \frac{M}{2}, ..., M - 1 \quad (14)$$

odakle slijedi:

$$y(n) = \sqrt{\frac{2}{M}} \sum_{k=0}^{M-1} x_3(k) \times \cos\left(\frac{\pi}{M}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right) \quad n = 0, ..., M - 1 \quad (15)$$

Dobijena transformacija je kvadratna (MxM) ortogonalna transformacija poznata pod imenom DCT-IV. Transformacija (15) ima ulogu jezgra transformacije.

U posljednjem koraku se obavlja izračunavanje transformacije (15). To se uglavnom obavlja primjenom nekog od brzih algoritama (detaljnije u poglavlju 1.3)

1.2 Standardi za kompresiju audio signala zasnovani na ortogonalnim transformacijama sa preklapanjem

Algoritmi za kodovanje ili kompresiju audio signala se koriste u cilju postizanja kompaktne digitalne predstave zvučnih signala koji se prostiru po kompletnom čujnom opsegu sa veoma malim izobličenjima originalnog signala u cilju efikasnog prenosa ili skladištenja. Glavni cilj audio kodovanja je predstavljanje signala sa minimalnim brojem bita uz postizanje transparentne reprodukcije signala, tj., generisanje izlaznog audio signala kod kojeg se ne mogu uočiti razlike u odnosu na originalni signal čak ni od strane iskusnog slušaoca sa izoštrim sluhom (engl. “golden ears”).

Komercijalne primjena algoritama za kompesiju audio signala obuhvata primjenu u bioskopima (digitalan zapis filmskog zvuka), primjenu u uređajima široke potrošnje (engl. “consumer devices”) kao što su Sonijev mini disk, prenosivi uređaji za reprodukciju zvuka (engl. “portable audio player“), konzole za video igre i DVD tj. kućni bioskop (engl. “home theatre”) i primjenu u sistemima za prenos podataka kao što su prenos podataka putem interneta, distribucija zvuka satelitskim i zemaljskim (etrom i kablovskim sistemima) distributivnim sistemima (DAB, DRM) i primjena u digitalnoj televiziji.

Pojavom kompakt diska u ranim 1980-tim ukazale su se sve prednosti digitalne predstave zvuka uključujući nadmoćan kvalitet reprodukovano zvuka u odnosu na dotašnje metode analognog zapisa zvuka, velik dinamički opseg i znatnu robustnost. Standardni sistemi za digitalan zapis zvuka (kompakt disk – CD i digitalna audio traka DAT) zahtijevaju velike količine podataka – 1.41/1.54 Mbit u sekundi za stereo signal odabiran sa 44.1/48 kHz. Iako prilično visoke, te bitske brzine su uspješno primjenjene u prvoj generaciji digitalnih audio aplikacija kao što su CD i DAT.

Druga generacija multimedijalnih aplikacija i sistemi za bežični prenos podataka su podvrgnuti ograničenjima propusnog opsega i malom cijenom koštanja koji su nespojivi sa velikim bitskim brzinama (na primjer 6 kanalni audio signal u 16 bitnom PCM formatu odabiran na 48 kHz zahtijeva 4.6 Mbita u sekundi). Zbog velikog uspjeha prve generacije audio aplikacija krajnji korisnik očekuje od svakog digitalnog sistema kvalitet reprodukcije ekvivalentan kvalitetu reprodukcije kompakt diska (engl. “CD-quality”). Moderni multimedijalni audio sistemi moraju da zadovolje zahtjeve kao što su: smanjenje bitske brzine, transparentan kvalitet, minimalna složenost za realizaciju, razumni zahtjevi za tipovima podataka, primjenjivost na raznovrsan skup signala, robustnost, skalabilnost i proširivost. Pomenuti zahtjevi su motivisali obimna istraživanja tokom prethodne dekade u cilju formulisanja algoritama za kompresiju koji istovremeno zadovoljavaju konfliktne zahtjeve za niskim bitskim brzinama i transparentim kvalitetom reprodukcije visokokvalitetnih audio signala.

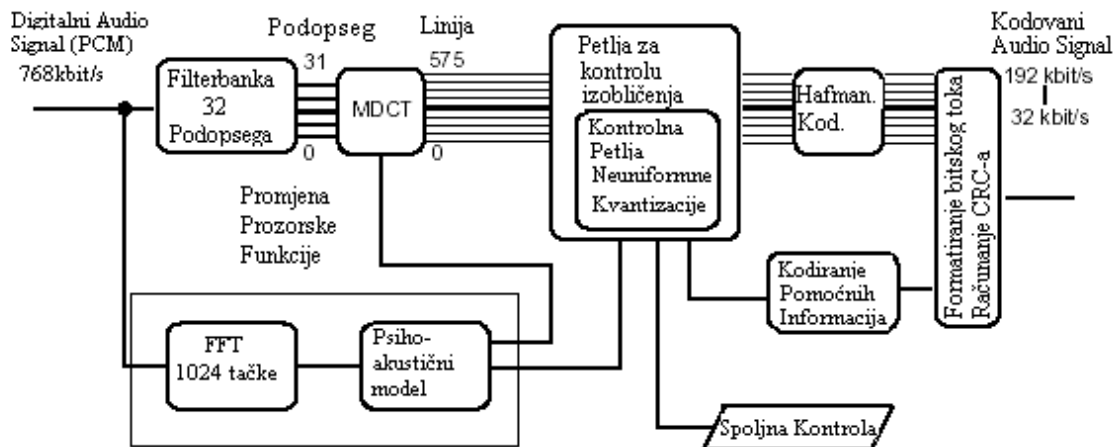
Kao rezultat toga razvijen je veliki broj standardizovanih algoritama za kompresiju audio signala kao što su ISO/IEC MPEG1 Layer III ili MP3 (sl. 1.5), ISO/IEC MPEG2 AAC (sl. 1.6), Dolby AC3 (sl. 1.7), Sony ATRAC (sl. 1.8) i Lucent Technologies PAC (sl. 1.9). U svim pomenutim algoritmima koristi se modifikovana diskretna kosinusna transformacija. Na primjeru MP3 algoritma koji je postao standard za prenos i skladištenje komprimovanog zvuka u internet aplikacijama i prenosnim multimedijalnim aplikacijama i AAC algoritma koji je prihvaćen kao dio MPEG-4 standarda ćemo pokazati neke od načina upotrebe modifikovane diskretne transformacije u algoritmima za kompresiju audio signala.

U MP3 algoritmu MDCT se koristi u kombinaciji sa polifaznom filter bankom za analizu koja obavlja dekompoziciju ulaznog signala na 32 kritično odabirana podopsega. Ovi podopsezi su jednake širine i ulazni signal odabiran na 48 kHz se dijeli na podopsege široke 750 Hz, gdje su podopsezi decimirani 32 puta. Polifazna filter banka koristi prototipski filter 511 reda takav da je ukupno izobličenje polifazne filter banke ispod praga čujnosti. Svaki filter podopsega praćen je adaptivnom MDCT što omogućuje veću spektralnu rezoluciju. Korištenjem MDCT 18 reda postiže se frekventna rezolucija od 41.67 Hz po spektralnoj liniji. Na sličan način se u ATRAC algoritmu kombinuju QMF banka za analizu i MDCT. Za razliku od ova dva koda u AAC koderu za analizu koristi samo MDCT (isto vrijedi i za AC3 i PAC).

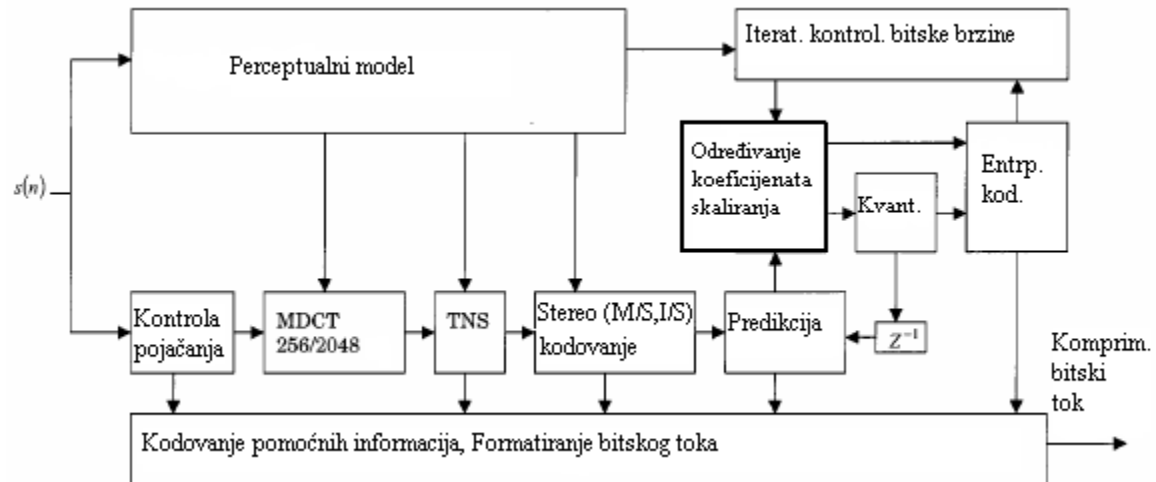
U cilju kontrole prednjeg odjeka (engl. “pre echo”) u MP3 algoritmu adaptivno se bira između MDCT reda 6 i 18 u cilju smanjivanja prednjeg odjeka. Kraći blokovi (4 ms) omogućuju vremensko maskiranje prednjeg odjeka za vrijeme prelaznih perioda dok duži

blokovi omogućuju povećanje stepena kompresije za vrijeme stacionarnih perioda. Na sličan način se i u AAC algoritmu adaptivno mijenja rezolucija filter banke. Stacionarni signali se analiziraju korištenjem prozorske funkcije dužine 2048 tačaka dok se brzo promjenljivi signali analiziraju korištenjem bloka od osam prozora dužine 256 tačaka u cilju održavanja sinhronizacije sa kanalima koji koriste različitu rezoluciju filter banke za vrijeme rada u višekanalnom režimu (AAC je višekanalni koder). Maksimalna rezolucija u frekventnom domenu je 23 Hz za frekvenciju odabiranja od 48 kHz a maksimalna vremenska rezolucija je 2.6 ms. Na sličan način se obavlja izbor između visoke spektralne i visoke vremenske rezolucije u ostalim algoritmima.

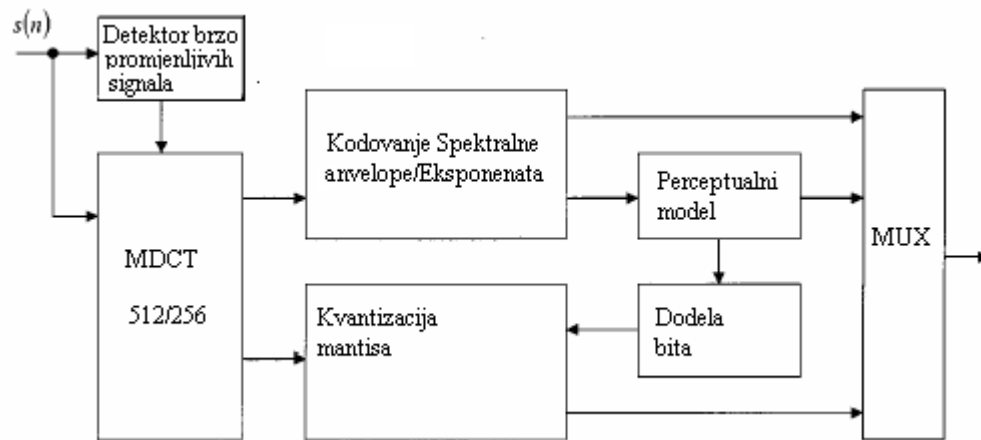
Filter banka koja se koristi u AAC algoritmu posjeduje jedinstvenu karakteristiku da postoji mogućnost izbora između dva različita oblika prozora za analizu MDCT filter banke. Na osnovu ulaznog signala bira se prozorska funkcija čiji oblik maksimizira dobitak perceptualnog kodovanja (engl. “perceptual coding gain”) filter banke. Oba prozora zadovoljavaju uslov potpune rekonstrukcije i eliminisanja preklapanja spektra koje važe za MDCT ali pružaju različite mogućnosti za spektralnu analizu. Sinusni prozor (slike 1.3 i 1.4) se koristi kad je potrebna velika selektivnost u propusnom opsegu nego velika selektivnost u nepropusnom opsegu, što je slučaj kod ulaznih signala sa gusto raspoređenim harmonicima (harmonici udaljeni za manje od 140 Hz) kao kod čembala i nekih duvačkih instrumenata. S druge strane, KBD prozor (slike 1.3 i 1.4) se bira u slučajevima kad je potrebno veće potiskivanje u nepropusnom opsegu ili kada su izražene tonalne komponente udaljene za više od 120 Hz.



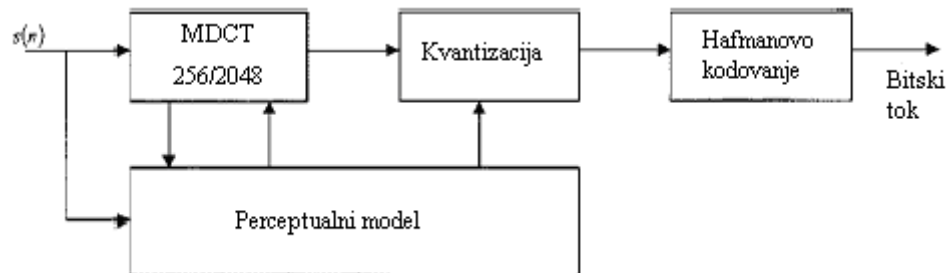
Slika 1.5 Blok dijagram MP3 enkodera



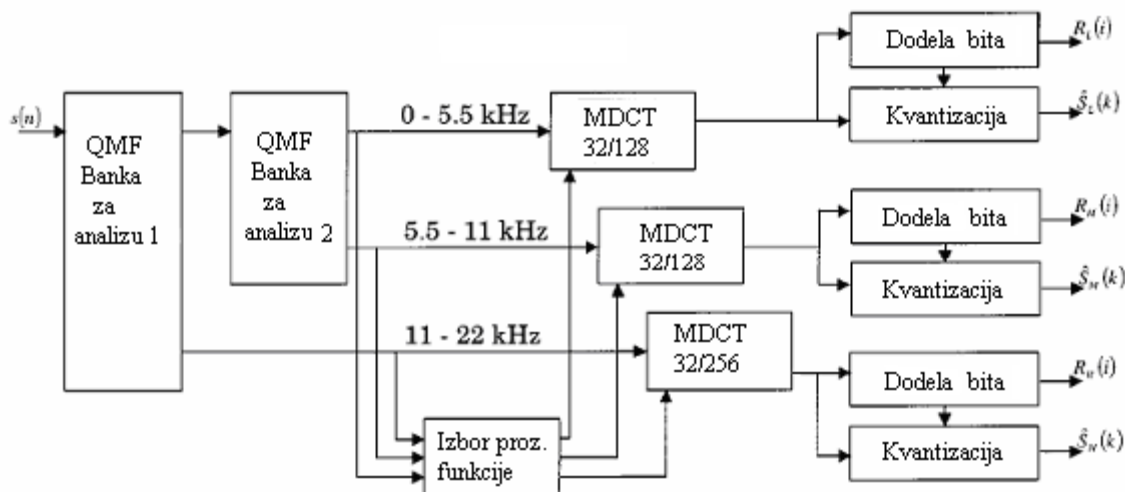
Slika 1.6 Blok dijagram AAC enkodera



Slika 1.7 Blok dijagram AC3 enkodera



Slika 1.8 Blok dijagram PAC enkodera



Slika 1.9 Blok dijagram ATRAC enkodera

1.3 Izračunavanje transformacija sa preklapanjem korištenjem brzih algoritama baziranih na brzjoj Furijeovoj transformaciji

Široko rasprostranjenom primjeni MDCT, naročito u standardima za kompresiju zvuka, doprinijelo je između ostalog i postojanje brzih algoritama za izračunavanje zansovanih na brzjoj Furijeovoj transformaciji. Iako postoje brzi algoritmi za direktno izračunavanje diskretne kosinusne transformacije njihova neregularna struktura ih čini nepogodnim za primjenu na digitalnim signalnim procesorima. Mnogo pogodniji za realizaciju na digitalnim signalnim procesorima su algoritmi za izračunavanje brze Furijeove transformacije (npr. radix algoritmi) koji imaju regularnu strukturu.

DCT-IV može da se izračuna korištenjem brze Furijeove transformacije veličine $M \times M$ na slijedeći način:

$$Y(n) = \sqrt{2} \times e^{\frac{j\pi}{M} \left(n + \frac{1}{4} \right)} \left(\frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} \left(x3(k) \times e^{\frac{j\pi k}{2M}} \right) \times e^{\frac{2j\pi nk}{M}} \right) \quad n = 0, \dots, M-1 \quad (16)$$

$$y(2n) = \text{Re}\{Y(n)\} \quad \text{i} \quad y(2n+1) = -\text{Re}\{Y(M-1-n)^*\} \quad n = 0, \dots, \frac{M}{2} - 1 \quad (17)$$

U praksi se često DCT-IV računa korištenjem brze kompleksne Furijeove transformacije dvostruko manje veličine tako da se prvo primjeni postupak zvani premodulacija, pa zatim izračuna brza Furijeova transformacija, a zatim se na rezultate brze Furijeove transformacije primjeni postupak zvani postmodulacija. Postupak

premodulacije obuhvata preuređivanje i množenje ulaznih vrijednosti sa faktorom zaokreta $e^{\frac{j(k+0.25)\pi}{M}}$. Postupak postmodulacije takođe obuhvata preuređivanje koeficijenata i njihovo množenje sa faktorom zaokreta $e^{\frac{jk\pi}{M}}$.

1.3.1 Brza Fourijeova transformacija (radix-2)

Direktna i inverzna Furijeova transformacija dužine $N=2^p$ je definisana sa

$$y(N, n) = \frac{1}{N} \sum_{k=0}^{N-1} x(N, k) \times w(N)^{nk} \quad n = 0, \dots, N-1 \quad (18)$$

$$x(N, k) = \sum_{n=0}^{N-1} y(N, n) \times w(N)^{-nk} \quad k = 0, \dots, N-1 \quad (19)$$

gdje su koeficijenti w stepeni baze transformacije:

$$w(N) = e^{-\frac{2\pi j}{N}}, j = \sqrt{-1} \quad (20)$$

Vrijede slijedeće osobine:

$$-w(N)^N = 1, w(N)^{\frac{N}{2}} = -1, w(N)^2 = w\left(\frac{N}{2}\right) \quad (21)$$

$$-y(N, rN + n) = y(N, n), x(N, rN + k) = x(N, k) \quad (22)$$

-ako je $x(N, k)$ niz realnih brojeva onda su i $y(N, 0)$ i $y(N, N/2)$ takođe realni brojevi i $y(N, N-n) = y(N, n)^*$ za $n=1, \dots, N/2-1$

1.3.1.1 Algoritam sa preuređivanjem u vremenu

$$y(N, n) = \frac{1}{2} \left[y'\left(\frac{N}{2}, n\right) + w[N, n] * y''\left(\frac{N}{2}, n\right) \right] \quad (23)$$

$$y\left(N, \frac{N}{2} + n\right) = \frac{1}{2} \left[y'\left(\frac{N}{2}, n\right) - w[N, n] * y''\left(\frac{N}{2}, n\right) \right] \quad n = 0, \dots, \frac{N}{2} - 1 \quad (24)$$

gdje je

y' transformacija dužine $N/2$ nad parnim odbircima, i
 y'' transformacija dužine $N/2$ nad neparnim odbircima.

Na početku je potrebno preurediti ulazne odbirke po bit reverznom pravilu. Algoritam se izvršava u $\log_2(N)$ koraka (stepeni).

1.3.1.2 Algoritam sa preuređivanjem u frekvencijskom domenu

Transformacija dužine N se može definisati sa:

$$y(N, 2n) = \frac{1}{2} y' \left(\frac{N}{2}, n \right) \quad (25)$$

$$y(N, 2n+1) = \frac{1}{2} y'' \left(\frac{N}{2}, n \right) \quad n = 0, \dots, \frac{N}{2} - 1 \quad (26)$$

gdje je

y' transformacija dužine $N/2$ nad odbircima definisanim sa:

$$x' \left(\frac{N}{2}, k \right) = x(N, k) + x \left(N, \frac{N}{2} + k \right) \quad k = 0, \dots, \frac{N}{2} - 1 \quad (27)$$

y'' transformacija dužine $N/2$ nad odbircima definisanim sa:

$$x'' \left(\frac{N}{2}, k \right) = x(N, k) - x \left(N, \frac{N}{2} + k \right) \times w(N, k) \quad (28)$$

U ovom slučaju izlazni odbirci se moraju preurediti po bit reverznom pravilu. Algoritam se takođe izvršava u $\log_2(N)$ koraka (stepeni).

1.4 Problematika primjene transformacija sa preklapanjem korištenjem digitalnih signalnih procesora

Transformacije visokog reda zahtijevaju upotrebu arhitektura nepokretnog zarezaja koje koriste velike dužine riječi (engl. “wordlength”) što implicira veću kompleksnost.

Dugačke transformacije se obično izračunavaju korištenjem brzih transformacija. One se uglavnom zasnivaju na brzom Furijeovoj transformaciji pa će greške nastale zbog kvantizacije rezultata množenja i izbjegavanja prekoračenja biti analizirane za slučaj brze Furijeove transformacije.

Greške nastale prilikom premodulacije i postmodulacije takođe utiču na ukupnu grešku izračunavanja, ali zbog jednostavne prirode ovih operacija i njihove sličnosti sa operacijama koje se koriste prilikom izračunavanja brze Furijeove transformacije analiza njihovog uticaja predstavlja podskup analize uticaja grešaka nastalih prilikom izračunavanja brze Furijeove transformacije.

U poglavlju 1.1.1 prilikom prikaza transformacija sa preklapanjem istaknuto je da ovakve transformacije mapiraju blok od L odbiraka u M vrijednosti ($L > M$) kod direktne transformacije, dok inverzna transformacija mapira blok od M vrijednosti u L vrijednosti.

Na slici 1.1b je moguće uočiti da je za realizaciju inverzne transformacije potrebno sačuvati $L/2$ vrijednosti izračunatih u prilikom prethodnog računanja inverzne transformacije i kombinovati ih sa L vrijednosti izračunatih prilikom računanja tekuće transformacije u okviru procedure "preklopi i saberi". U jednom trenutku je potrebno čuvati $3L/2$ vrijednosti što za velike dužine transformacija nije zanemarljiva vrijednost (na primjer, za AAC stereo dekodera bi bilo potrebno 6144 memorijske lokacije). U AAC algoritmu kao i većini poznatih algoritama za kompresiju audio signala koristi se MDCT i IMDCT kod koje je $L=2M$. Uglavnom se pri implementaciji inverzne modifikovane diskretne kosinusne transformacije koristi DCT-IV transformacija kao jezgro transformacije. DCT-IV je kvadratna transformacija i njenim izračunavanjem se dobija M vrijednosti. Moguće je iskoristiti simetričnost veze između IMDCT i DCT-IV koja je predstavljena na slici 1.10. Efikasnim korištenjem osobine simetričnosti rezultata inverzne transformacije dovoljno je čuvati samo polovinu vrijednosti izračunatih prilikom prethodnog izračunavanja transformacije kao i samo M vrijednosti tekuće transformacije ($M/2+M$ vrijednosti u odnosu na $M + 2M$ vrijednosti) što je značajna ušteda (na primjer, kod AAC stereo dekodera moguće je uštedjeti 3072 memorijske lokacije).

Dodatna memorija potrebna za čuvanje koeficijenata prozorske funkcije se takođe može smanjiti korištenjem simetričnosti prozorske funkcije.

DCT-IV (dužina = M)

A	B
---	---

IMDCT (dužina = $2M$)

B	$-B^{\text{obrnut}}$	$-A^{\text{obrnut}}$	$-A$
---	----------------------	----------------------	------

Slika 1.10 Grafička predstava simetričnih veza između DCT-IV i IMDCT

S obzirom da se izračunavanje transformacija sa preklapanjem svodi na izračunavanje brze Furijeove transformacije, arhitektura digitalnog signalnog procesora treba da omogućuje efikasnu implementaciju brze Furijeove transformacije. Izračunavanje brze Furijeove transformacije se sastoji od ponavljanja sekvence jednostavnih operacija. To ukazuje na potrebu da arhitektura omogućuje ponavljanje sekvence jednostavnih operacija bez potrebe za dodatnim kontrolnim operacijama (engl. "zero-overhead loop"). Brza Furijeova transformacija zahtijeva preuređivanje niza vrijednosti po bit-reverznom pravilu pa je poželjno da digitalni signalni procesor podržava takvu tehniku adresiranja.

Elementarna operacija brze Furijeove transformacije se sastoji od kompleksnog množenja i kompleksnog sabiranja. Kompleksno množenje se izvodi kao dva puta po dva realna množenja čiji rezultati se sabiraju (za realnu i za imaginarnu komponentu). Mogućnost izvođenja množenja i sabiranja u okviru iste instrukcije je neohodna za efikasnu implementaciju brze Furijeove transformacije. Takođe je potrebno postojanje

mogućnosti istovremenog pristupa objema memorijskim zonama zbog potrebe za istovremenim čitanjem oba operanda množenja (realna ili imaginarna komponenta međurezultata i realna ili imaginarna komponenta faktora zaokreta) ili smještanjem rezultata istovremeno sa odvijanjem neke računske operacije. U cilju paralelizacije izvođenja operacija poželjno je da digitalni signalni procesor posjeduje veći broj akumulatorskih registara.

Zbog potrebe za kontrolom opsega vrijednosti međurezultata potrebna je oznaka prekoračenja (engl. overflow flag) a poželjno je postojanje i oznake mogućeg prekoračenja (engl. half overflow flag) kao i pomjerači koji omogućuju pomjeranje vrijednosti koja se smješta u memoriju u toku smještanja u memoriju (engl. on-fly shifters) za jedan bit udesno što je ekvivalentno dijeljenju sa dva. Pomjerači istog tipa omogućuju i pomjeranje prilikom čitanja iz memorije.

Rad je organizovan na sljedeći način. Transformacija sa preklapanjem i brzi algoritam za njeno izračunavanje predstavljeni su u prvom poglavlju. Izvori grešaka, akumulacija grešaka i dobijeni odnos signal-šum su predstavljeni u drugom poglavlju. Progresivna akumulacija grešaka je ilustrovana na primjerima postojećih rješenja za implementaciju brzog algoritma na nekoliko karakterističnih signala trećem poglavlju. Poboljšano rješenje je predstavljeno u četvrtom poglavlju, kao i njegovo poređenje sa ostalim rješenjima. Praktičan primjer primjene poboljšanog rješenja u implementaciji AC3 dekodera je prikazan u petom poglavlju. Zaključak je dat u šestom poglavlju.

2. Tačnost izračunavanja brzih algoritama na DSP

2.1. Greške kvantizacije

Prilikom implementacije algoritama za digitalnu obradu signala na digitalnim signalnim procesorima ili specijalizovanim elementima fizičke arhitekture koji koriste aritmetiku nepokretnog zareza neizbježne su greške i ograničenja koja proizilaze iz konačne dužine registara i memorijskih riječi. Glavne kategorije posljedica konačne dužine registara i memorijskih riječi su greške u ulaznim podacima (kvantizacija ulaznih parametara, A/D konverzija), greške zbog zaokruživanja rezultata aritmetičkih operacija, ograničenja nivoa signala postavljena potrebom izbjegavanja prekoračenja opsega i kvantizacija koeficijenata algoritma. U narednim paragrafima će biti razmatrane greške nastale zbog zaokruživanja rezultata aritmetičkih operacija kao i ograničenja nivoa signala postavljena potrebom izbjegavanja prekoračenja i posljedice tih ograničenja.

Pretpostavimo da brojeve u nepokretnom zarezu predstavljamo sa b bita plus bit znaka. Množenjem dva takva broja dobija se broj za čije je predstavljanje u aritmetici nepokretnog zareza potreban dvostruko veći broj bita. Ukoliko je dužina registara i memorijskih riječi b bita tada potrebno izvršiti ograničenje broja bita rezultata na b bita čime se nepovratno gubi preostalih b bita proizvoda (dio informacionog sadržaja podatka) i unosi greška. Čak i ako je postoji mogućnost sačuvanja i iskorištenja preostalih b bita (ili njihovog dijela) nakon narednog množenja rezultat bi imao $3b$ bita, itd., i prije ili kasnije bi bilo neophodno izvršiti ograničenje broja bita rezultata. Prilikom korištenja aritmetike nepokretnog zareza prirodno je u kontekstu obrade signala razmatrati registar kao predstavu razlomljenog broja sa zarezom pozicioniranim krajnje lijevo, neposredno nakon bita znaka. Na taj način proizvod dva broja ostaje razlomljen broj i ograničena dužina registra se može održavati odsjecanjem ili zaokruživanjem bita najmanje težine. U ovakvoj predstavi brojeva rezultat sabiranja (oduzimanja) dva razlomljenja broja ne zahtijeva ni odsijecanje ni zaokruživanje bita ali može da poraste po apsolutnoj vrijednosti tako da zbir eventualno više nije razlomljeni broj (cijeli dio broja postaje različit od nule). Ova pojava se uobičajeno naziva prekoračenje i taj problem može da se riješava zahtijevajući da su ulazni podaci dovoljno mali da bi se izbjegla mogućnost prekoračenja. Prekoračenje opsega kod sabiranja može da se riješava i dodavanjem određenog broja bita registrima na pozicijama lijevo od decimalne tačke (biti prekoračenja – engl. *overflow bits*). Iako se na taj način omogućuje efektivno proširenje opsega kod izvođenja sabiranja time se ne izbjegava potreba za smanjivanjem opsega podataka jer se podaci u proširenom opsegu obično ili smještaju u memoriju ili se u daljoj obradi množe sa drugim brojevima (množać radi samo sa brojevima u opsnovnom opsegu) te ih je potrebno svesti u osnovni opseg. Svođenje brojeva iz proširenog opsega u osnovni opseg se izvodi operacijama skaliranja nadole (dijeljenjem sa stepenom broja dva – obično se izvodi aritmetičkim pomjeranjem broja udesno) i zatim odsjecanjem ili zaokruživanjem bita najmanje težine.

Zbog značaja matematičkih operacija odsijecanja i zaokruživanja bita najmanje težine prilikom primjene aritmetike nepokretnog zareza pomenute operacije će se detaljnije razmotriti.

Prilikom analize matematičkih operacija zaokruživanja i odsijecanja razmatrat ćemo brojeve u nepokretnom zarezu predstavljene kao $b+1$ bitne razlomljene brojeve sa binarnom tačkom na poziciji desno od bita najveće težine (bit znaka). Numerička vrijednost (za pozitivne brojeve) jedinice na bitu najmanje težine je 2^{-b} i na tu vrijednost ćemo se poslije pozivati kao korakom kvantizacije. Razmatrat ćemo samo binarnu predstavu brojeva u formatu potpunog komplementa.

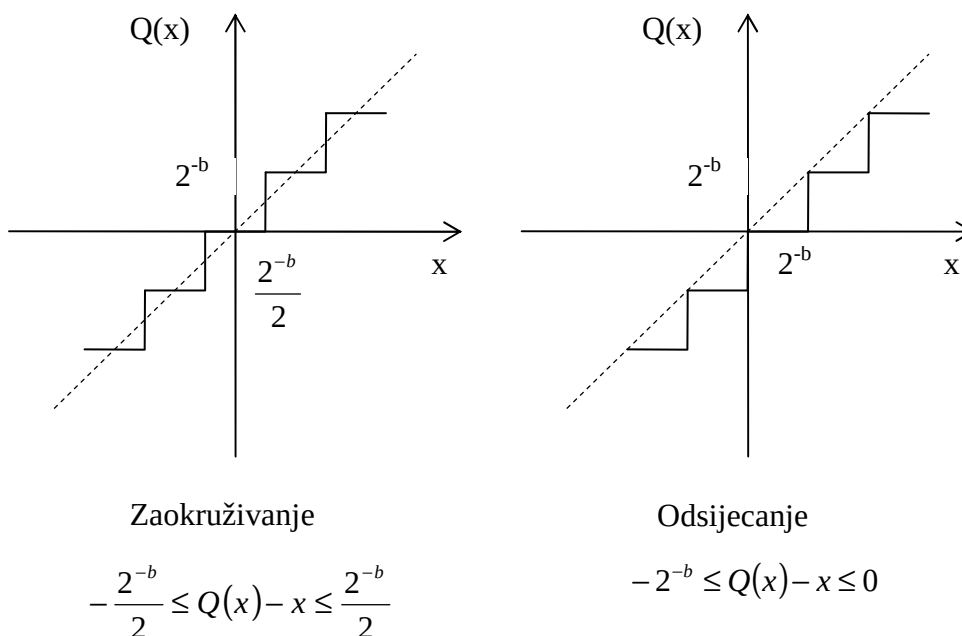
Ako sa E_o označimo grešku nastalu odsijecanjem, tj., vrijednost poslije odsijecanja umanjenu za vrijednost poslije odsijecanja, ova vrijednost će uvijek biti negativna ili nula za pozitivne brojeve. To znači da je posljedica odsijecanja smanjenje veličine brojeva. Kod negativnih brojeva posljedica odsijecanja je povećanje vrijednosti negativnog broja pa je i u ovom slučaju greška nastala odsijecanjem negativna. Ako sa b_1 označimo broj bita broja prije odsijecanja, tada rezultujuća greška E_o zadovoljava sljedeće:

$$0 \geq E_o \geq -(2^{-b} - 2^{-b_1})$$

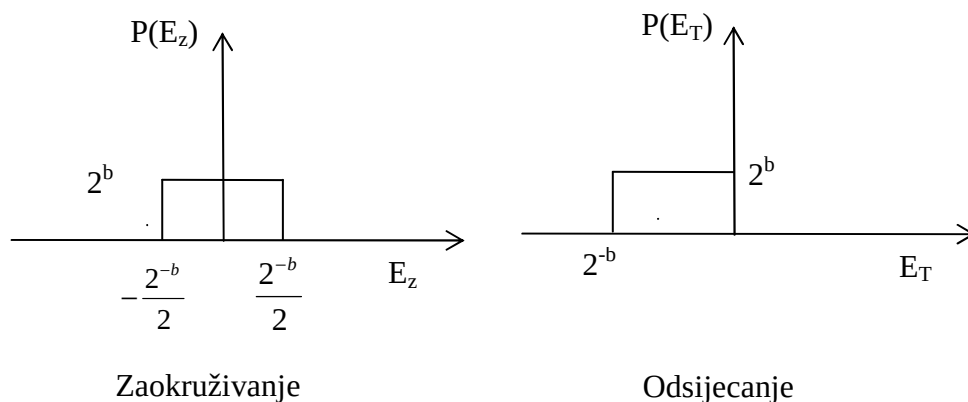
Rezultujuća greška zaokruživanja E_z zadovoljava sljedeće:

$$-\frac{2^{-b}}{2} \leq E_z \leq \frac{2^{-b}}{2}$$

Na slici 2.1 prikazane su prenosne funkcije zaokruživanja i odsijecanja.



Slika 2.1 Prenosne funkcije zaokruživanja i odsijecanja



Slika 2.2 Funkcije raspodjele vjerovatnoće za zaokruživanje i odsijecanje

Statistička predstava greške predstavlja pogodan način za analizu posledica kvantizacije. Greška kvantizacije je predstavljena kao slučajna promjenljiva sa funkcijom raspodjele prikazanom na slici 2.2 operacije zaokruživanja i odsijecanja. U oba slučaja se pretpostavlja da je greška slučajna promjenljiva nezavisna od ulazne vrijednosti x . Varijansa (disperzija) šuma je jednaka za oba slučaja ali je srednja vrijednost šuma kod zaokruživanja jednaka nuli dok je kod odsijecanja srednja vrijednost greške različita od nule.

Razmatra se greška nastalu zaokruživanjem prilikom računanja diskretne DFT preko FFT algoritma. Prvo se analizira greška nastala zaokruživanjem prilikom računanja diskretne Furijeove transformacije koja se koristi kao referenca.

2.2 Greška kvantizacije kod direktnog računanja diskretne Furijeove transformacije (DFT)

Neka je $\{x(n)\}$, $0 \leq n \leq N-1$ konačan niz. Tada je diskretna Furijeova transformacija (DFT) niza $\{x(n)\}$ definisana sa

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{kn} \quad k=0,1,\dots,N-1 \quad (29)$$

gdje je $W_N = e^{-j2\pi/N}$. Pretpostavljamo da je $\{x(n)\}$ niz kompleksnih brojeva. Takođe pretpostavljamo da su realne i imaginarnе komponente od $\{x(n)\}$ i od $\{W_N^{kn}\}$ predstavljeni sa b bita (ne uključujući bit znaka). Za računanje kompleksnog proizvoda $x(n)W_N^{kn}$ potrebno je izvršiti četiri realna množenja. Rezultat svakog realnog množenja je zaokružen sa $2b$ bita na b bita, što znači da se kod svakog kompleksnog množenja javljaju četiri greške kvantizacije.

Kod direktnog računanja diskretne Furijeove transformacije postoji N kompleksnih množenja za svaku tačku diskretne Furijeove transformacije. Znači da je

ukupan broj realnih množenja potreban za računanje jedne tačke diskretne Furijeove transformacije $4N$ te je i broj kvantizacionih grešaka $4N$.

Da bi odredili varijansu grešaka kvantizacije kod računanja diskretne Furijeove transformacije korištenjem aritmetike nepokretnog zareza pretpostavit ćemo da kvantizacione greške posjeduju sljedeće statističke osobine:

- Kvantizacione greške nastale zaokruživanjem su uniformno raspoređene slučajne promjenljive u intervalu $(-\Delta/2, \Delta/2)$ gdje je $\Delta = 2^{-b}$.
- Svih $4N$ kvantizacionih grešaka su međusobno nezavisne.
- Svih $4N$ kvantizacionih grešaka su nezavisne od ulaznog niza $\{x(n)\}$.

S obzirom da greška kvantizacije ima varijansu

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12} \quad (30)$$

varijansa (disperzija) grešaka kvantizacije od $4N$ množenja je

$$\sigma_q^2 = 4N\sigma_e^2 = N \frac{2^{-2b}}{3} \quad (31)$$

Iz toga proizilazi da je varijansa greške kvantizacije proporcionalna redu diskretne Furijeove transformacije. Takođe se može primjetiti da povećanje reda transformacije četiri puta zatijeva povećanje tačnosti računanja (broj bita) za jedan da bi se kompenzovala dodatne greške kvantizacije.

Da bi izbjegli prekoračenje potrebno je skalirati ulazni niz. Očigledno je da je gornja granica za $|X(k)|$

$$|X(k)| \leq \sum_{n=0}^{N-1} |x(n)| \quad (32)$$

Ako je dinamički opseg sabiranja $(-1,1)$ tada $|X(k)| < 1$ zahtijeva da je

$$\sum_{n=0}^{N-1} |x(n)| < 1 \quad (33)$$

Ako je $|x(n)|$ inicijalno skalirano tako da je $|x(n)| < 1$ za svako n , tada je potrebno svaku element ulaznog niza podijeliti sa N da bi ispunili prethodni zahtijev.

Prethodno skaliranje je prilično restriktivno. Pod pretpostavkom da je ulazna sekvenca bijeli šum i da su nakon skaliranja članovi ulaznog niza uniformno raspoređeni u opsegu $(-1/N, 1/N)$. Tad je varijansa ulaznog signala

$$\sigma_x^2 = \frac{(2/N)^2}{12} = \frac{1}{3N^2} \quad (34)$$

i varijansa rezultata diskretne Furijeove transformacije $|X(k)|$ je

$$\sigma_X^2 = N\sigma_x^2 = \frac{1}{3N} \quad (35)$$

Za ovakav signal odnos signal-šum je

$$\frac{\sigma_X^2}{\sigma_q^2} = \frac{2^{2b}}{N^2} \quad (36)$$

Može se primjeti da je skaliranje uzrokovalo smanjenje odnosa signal-šum N puta i da je kombinacija skaliranja i grešaka kvantizacije rezultirala u ukupnom smanjenju koje je proporcionalno sa N^2 . Očigledno je da skaliranje ulaznog niza $\{x(n)\}$ u cilju izbjegavanja prekoračenja ima bitan uticaj na odnos signal-šum u direktnom računanju diskretne Furijeove transformacije.

Ako se umjesto skaliranja ulaznog niza $\{x(n)\}$ sa N uvede manje strogo ograničenje $|x(n)| < 1$ onda se mora obezbijediti dovoljno velik dinamički opseg kod sabiranja ($|X(k)| < N$). U tom slučaju varijansa niza $\{x(n)\}$ je $\sigma_x^2 = \frac{1}{3}$ pa varijansa $|X(k)|$ iznosi

$$\sigma_X^2 = N\sigma_x^2 = \frac{N}{3} \quad (37)$$

i u tom slučaju bi odnos signal-šum bio

$$\frac{\sigma_X^2}{\sigma_q^2} = 2^{2b} \quad (38)$$

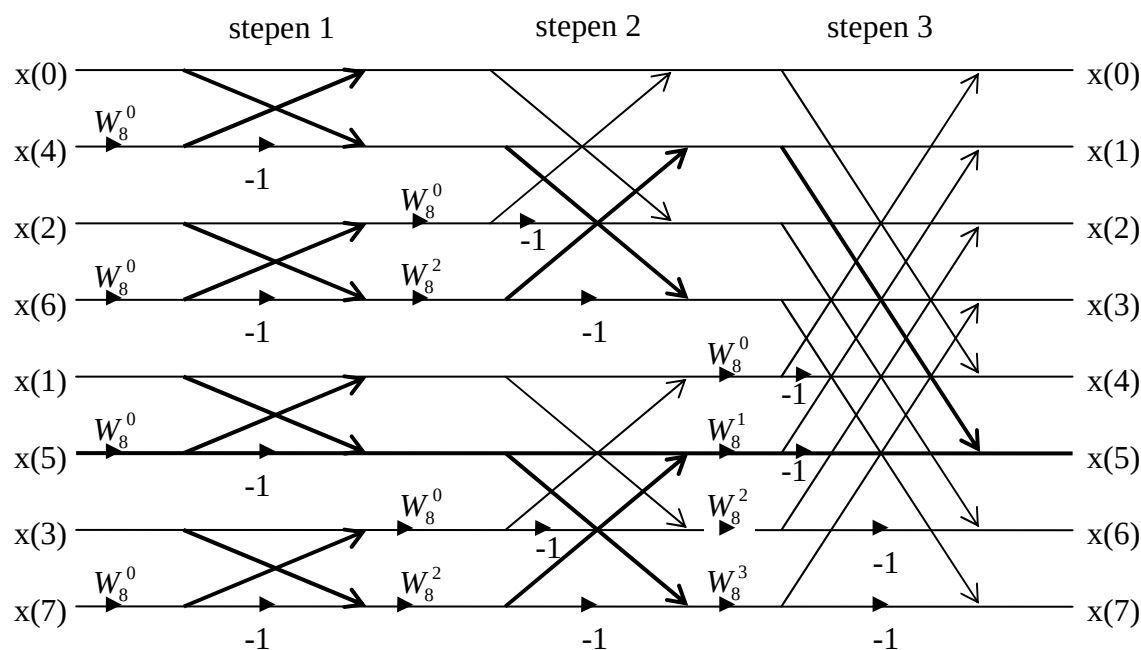
U ovom slučaju odnos signal šum ne zavisi od dužine transformacije nego samo od širine registara i memorijskih riječi b (odnos signal-šum u decibelima bi bio b puta 6dB). Za ovo je potrebno obezbijediti $\log_2(N)$ dodatnih bita u sabiračima (tzv. biti prekoračenja).

2.3 Greška kvantizacije kod računanja brze Furijeove transformacije

Poznato je da algoritmi za izračunavanje brze Furijeove transformacije zahtijevaju začajno manji broj množenja nego direktno izračunavanje diskretne Furijeove transformacije. Iz toga bi se moglo zaključiti da bi izračunavanje diskretne Furijeove

transformacije preko brze Furijeove transformacije rezultiralo manjom greškom kvantizacije.

Razmatrat ćemo izračunavanje brze Furijeove transformacije korištenjem radix-2 algoritma. Radix-2 algoritam koji se bazira na decimaciji u vremenu je ilustrovan na slici 2.3 za slučaj $N=8$. Rezultujuće greške kvantizacije za taj algoritam su slične greškama kvantizacije kod ostalih radix algoritama.



Slika 2.3 Radix-2 FFT DIT za $N=8$

Osnovni element radix-2 algoritama – baterflaj – se uobičajeno predstavlja kao

$$\underline{x}_{k+1}(i) = \underline{x}_k(i) + \underline{w}_N(k,j) * \underline{x}_k(j) \quad (39)$$

$$\underline{x}_{k+1}(j) = \underline{x}_k(i) - \underline{w}_N(k,j) * \underline{x}_k(j) \quad (40)$$

za algoritam decimacije u vremenu i

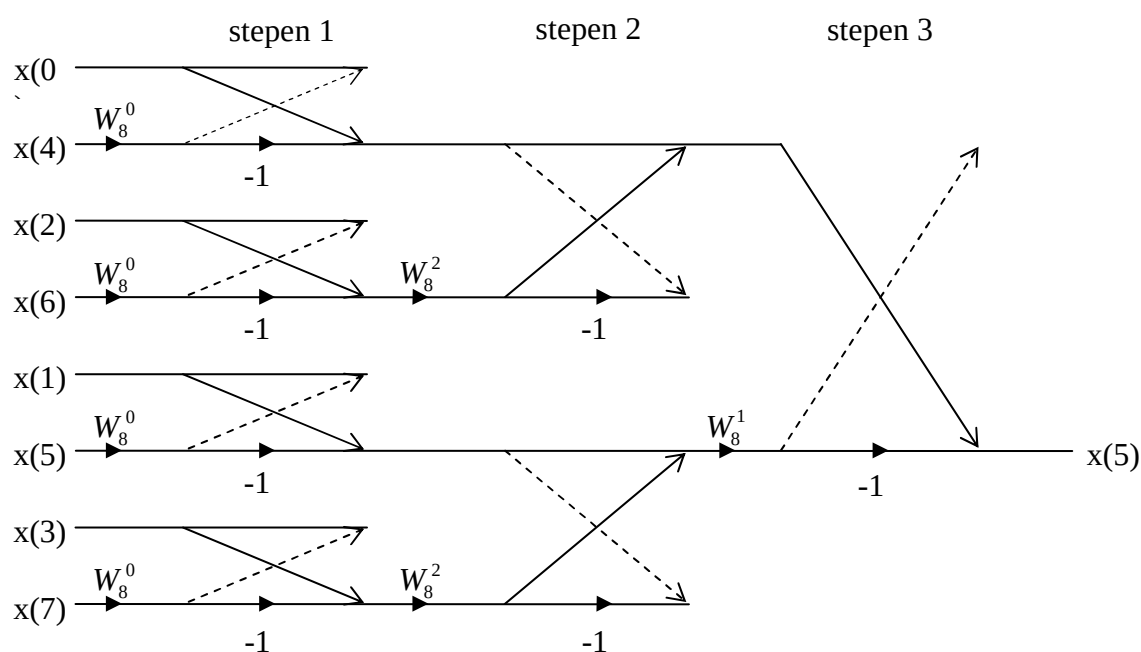
$$\underline{x}_{k+1}(i) = \underline{x}_k(i) + \underline{x}_k(j) \quad (41)$$

$$\underline{x}_{k+1}(j) = [\underline{x}_k(i) - \underline{x}_k(j)] * \underline{w}_N(k,j) \quad (42)$$

za algoritam decimacije u frekvencije, gdje u oba slučaja $\underline{x}_k(i)$ označava i-ti kompleksni koeficijent u k-tom prolazu brze Furijeove transformacije, a $\underline{w}_N(k,j)$ predstavlja faktor faznog zaokreta za j-ti koeficijent u k-tom prolazu transformacije dužine N .

Može se primjetiti da računanje svakog baterflaja uključuje jedno kompleksno množenje što je ekvivalentno četiri kompleksna sabiranja. Zanimljivo je činjenicu da neki baterflaji sadrže trivijalna množenja sa ± 1 ($\pm j$). Ako posmatramo baterflaje koji utiču na izračunavanje bilo koje tačke DFT (na slici 2.4 izdvojeni su baterflaji koji utiču na izračunavanje $x(5)$) možemo uočiti da ih se u prvom stepenu FFT-a nalazi $N/2$, u drugom $N/4$, $N/8$ u trećem i tako dalje sve do posljednjeg u kojem se nalazi samo jedan. Slijedi da je broj baterflaja po izlaznoj tački (uz $v = \log_2(N)$)

$$2^{v-1} + 2^{v-2} + \dots + 2 + 1 = 2^{v-1} \left[1 + \left(\frac{1}{2}\right) + \dots + \left(\frac{1}{2}\right)^{v-1} \right] = 2^v \left[1 - \left(\frac{1}{2}\right)^v \right] = 2^v - 1 = N - 1 \quad (43)$$



Slika 2.4 Prikaz baterflaja koji utiču na računanje $x(5)$

Greške kvantizacije nastale u svakom baterflaju se prostiru do izlaza. Može se uočiti da greške kvantizacije nastale u prvom stepenu se prostiru kroz $(v-1)$ stepen, one nastale u drugom stepenu se prostiru kroz $(v-2)$ stepena itd. Prilikom prostiranja tih kvantizacionih grešaka kroz uzastopne stepene FFT-a dolazi do faznog zaokreta za fazni faktor W_N^{kn} . Ti fazni zaokreti ne mijenjaju statističke osobine grešaka kvantizacije.

Ako pretpostavimo da greške kvantizacije u bilo kojem baterflaju nisu korelisane (nezavisne) sa greškama kvantizacije u ostalim baterflajima tada postoji $4(N-1)$ grešaka koje utiču na svaku tačku na izlazu FFT-a. Kao posledicu toga dobijamo da je varijansa ukupne greške kvantizacije na izlazu

$$\sigma_q^2 = 4(N-1)\frac{\Delta^2}{12} \approx \frac{N\Delta^2}{3} \quad (44)$$

gdje je $\Delta = 2^{-b}$. Slijedi

$$\sigma_q^2 = N \frac{2^{-2b}}{3} \quad (45)$$

Identičan rezultat je dobijen za direktno izračunavanje diskretne Furijeove transformacije. U suštini FFT algoritmi ne smanjuju (bitno) broj množenja potreban za računanje jedne tačke DFT-a. Oni koriste periodičnost W_N^{kn} i na taj način smanjuju ukupan broj množenja prilikom računanja cijelokupnog bloka od N tačaka u DFT.

Kao i kod direktnog računanja DFT potrebno je skalirati ulazni niz da bi izbjegli prekoračenje (ako je $|x(n)| < 1/N$ onda je $|X(k)| < 1$). Rezultujući odnos signal-šum je jednak odnosu signal-šum dobijenom za direktno računanje DFT.

S obzirom da se algoritam za računanje brze Furijeove transformacije sastoji iz niza stepeni, i da svaki stepen sadrži baterflaje koji uključuju parove tačaka, moguće je koristiti drugačiju strategiju skaliranja koja nije tako nezgodna (za odnos signal-šum) kao dijeljenje svakog ulaznog podatka sa N. Prilikom računanja FFT međuvrijednosti $|X_n(k)|$ u stepenima $n=1,2,\dots, v$ FFT algoritma zadovoljavaju sljedeće uslove

$$\max[|X_{n+1}(k)|, |X_{n+1}(l)|] \geq \max[|X_n(k)|, |X_n(l)|] \quad (46)$$

$$\max[|X_{n+1}(k)|, |X_{n+1}(l)|] \leq 2\max[|X_n(k)|, |X_n(l)|] \quad (47)$$

Iz ovih relacija proizilazi da je moguće raspodijeliti ukupno skaliranje od $1/N$ na sve stepene FFT algoritma. Ako je $|x(n)| < 1$ tad se primjeni skaliranje za dva (množenje sa $1/2$) u prvom stepenu tako da je $|x(n)| < \frac{1}{2}$. Zatim se rezultati svakog narednog stepena FFT algoritma skaliraju za $\frac{1}{2}$ tako da nakon v stepeni se dostigne ukupan iznos skaliranja od $\left(\frac{1}{2}\right)^v = \frac{1}{N}$. Na taj način je izbjegnuto prekoračenje prilikom računanja diskretne Furijeove transformacije FFT algoritmom.

Ovakvo skaliranje ne utiče na nivo izlaznog signala na izlazu FFT algoritma ali značajno smanjuje varijansu šuma kvantizacije na izlazu. Svaki faktor skaliranja od $\frac{1}{2}$ smanjuje iznos varijanse greške kvantizacije za faktor $\frac{1}{4}$. Na taj način je za 4 (N/2) grešaka kvantizacije nastalih u prvom stepenu umanjena varijansa za $\left(\frac{1}{4}\right)^{v-1}$, za 4 (N/4)

grešaka kvantizacije nastalih u drugom stepenu je umanjena varijansa za $\left(\frac{1}{4}\right)^{v-2}$ i tako dalje. Kao posljedica toga ukupna greška kvantizacije na izlazu FFT algoritma je

$$\begin{aligned}\sigma_q^2 &= \frac{\Delta^2}{12} \left\{ 4 \left(\frac{N}{2} \right) \left(\frac{1}{4} \right)^{v-1} + 4 \left(\frac{N}{4} \right) \left(\frac{1}{4} \right)^{v-2} + 4 \left(\frac{N}{8} \right) \left(\frac{1}{4} \right)^{v-3} + \dots + 4 \right\} \\ &= \frac{\Delta^2}{3} \left\{ \left(\frac{1}{2} \right)^{v-1} + \left(\frac{1}{2} \right)^{v-2} + \left(\frac{1}{2} \right)^{v-3} + \dots + \frac{1}{2} + 1 \right\} \\ &= \frac{2\Delta^2}{3} \left[1 - \left(\frac{1}{2} \right)^v \right] \approx \frac{2}{3} \cdot 2^{-2b}\end{aligned}\quad (48)$$

gdje je faktor $\left(\frac{1}{2}\right)^v$ zanemarljiv.

Može se primjetiti da varijansa šuma kvantizacije na izlazu više nije proporcionalna N. Varijansa signala je $\sigma_x^2 = \frac{1}{3N}$ pa je odnos signal-šum

$$\frac{\sigma_x^2}{\sigma_q^2} = \frac{2^{2b}}{2N} \quad (49)$$

Znači da je raspodjelom skaliranja $1/N$ ravnomjerno kroz FFT algoritam dobijen odnos signal-šum koji je inverzno proporcionalan sa N umjesto sa N^2 .

U prethodnoj analizi je pretpostavljeno da je jedini uzrok prekoračenja prilikom računanja diskretne Furijeove transformacije FFT algoritmom kompleksno sabiranje i da je ulazni niz $|x(n)| < 1$. Kod korištenja brze Furijeove transformacije za brzo izračunavanje transformacija sa preklapanjem ulazni niz u opštem slučaju ne zadovoljava prethodnu relaciju (realna i imaginarna komponenta su nastale od nezavisnih vrijednosti) te postoji mogućnost prekoračenja i prilikom izračunavanja kompleksnog množenja.

Koristi se predstava kompleksnog broja u obliku zbira realnog i imaginarnog dijela

$$x = \text{Re}[x] + j \cdot \text{Im}[x] \quad (50)$$

i vrijedi slijedeće ograničenje (aritmetici nepokretnog zareza):

$$-1 \leq \text{Re}[x], \text{Im}[x] \leq 1 \quad (51)$$

Kompleksno množenje može se predstaviti na sljedeći način

$$W_N^{kn} * x_k(n) = \cos(\theta_N(k,n)) * \text{Re}[x_k(n)] - \sin(\theta_N(k,n)) * \text{Im}[x_k(n)] \\ + j * [\cos(\theta_N(k,n)) * \text{Im}[x_k(n)] + \sin(\theta_N(k,n)) * \text{Re}[x_k(n)]] \quad (52)$$

Iako množenje sa W_N^{kn} (fazni zaokret) ne utiče na moduo rezultata

$$|x_k(n)| = |x_k(n) \cdot W_N^{kn}| \quad (53)$$

ono utiče na raspodjelu vrijednosti između realne i imaginarne komponente čime se stvara mogućnost prekoračenja.

Graničan slučaj bi bio onaj u kom znaci realnog i imaginarnog dijela od $x_k(n)$ odgovaraju znacima sinusa i kosinusa ugla $(\theta_N(k,n))$ ($\sin(\theta_N(k,n))$ and $\cos(\theta_N(k,n))$) tako da je apsolutna vrijednost sume produkata jednaka sumi apsolutnih vrijednosti produkata

$$|\cos(\theta_N(k,n)) * \text{Re}[x_k(n)] - \sin(\theta_N(k,n)) * \text{Im}[x_k(n)]| \\ = |\cos(\theta_N(k,n)) * \text{Re}[x_k(n)]| + |\sin(\theta_N(k,n)) * \text{Im}[x_k(n)]| \quad (54)$$

ili

$$|\cos(\theta_N(k,n)) * \text{Im}[x_k(n)] + \sin(\theta_N(k,n)) * \text{Re}[x_k(n)]| \\ = |\cos(\theta_N(k,n)) * \text{Im}[x_k(n)]| + |\sin(\theta_N(k,n)) * \text{Re}[x_k(n)]| \quad (55)$$

i vrijednosti realnog i imaginarnog dijela od $x_k(n)$ su jednake 1. Tada se dobija

$$|\text{Re}(W_N^{nk} x_k(n))| = |\cos(\theta_N(k,n))| + |\sin(\theta_N(k,n))| \quad (56)$$

ili

$$|\text{Im}(W_N^{nk} x_k(n))| = |\cos(\theta_N(k,n))| + |\sin(\theta_N(k,n))| \quad (57)$$

$$\max(|\cos \varphi| + |\sin \varphi|) = \sqrt{2}/2 + \sqrt{2}/2 = \sqrt{2} \text{ for } \varphi = \pi/4 + \kappa \pi/2 \quad (58)$$

To ukazuje na mogućnost prekoračenja kod kompleksnog množenja W_N^{kn} sa $x_k(n)$.

Iako maksimum koji iznosi $\sqrt{2}$ ne može biti dosegnut za svako kompleksno množenje prilikom izračunavanja FFT algoritma množenja sa rezultatima većim od jedan se mogu javiti u svakom stepenu FFT algoritma osim u prva dva kod DIT algoritma ili posljednja dva kod DIF algoritma (stepeni sa trivijalnim množenjima).

U svakom stepenu osim prva dva postoje dvije grupe baterflaja kod kojih se javlja mogućnost prekoračenja kompleksnog množenja maksimalne vrijednosti (mogućnost prekoračenja se može javiti i za druge uglove zaokreta faze) i javljaju se maksimalno jednom za bilo koju putanju.

2.4 Prostiranje greške kvantizacije

Prethodna analiza uticaja grešaka kvantizacije na izračunavanje diskretne Furijeove transformacije direktnim izračunavanjem kao i izračunavanjem uz pomoć jednog od algoritama brze Furijeove transformacije (radix-2 decimacija u vremenu) ukazuje na način nastajanja grešaka i njihovo prostiranje. Prilikom analize korištene su pretpostavke da greška kvantizacije posjeduje uniformnu raspodjelu i da je nezavisna od ulaznog signala. Iako to nije uvijek slučaj (npr. trivijalna kompleksna množenja se ponekad realizuju bez greške), to ne utiče na prirodu greške, prostiranje greške kao ni na ovisnost ukupne greške o veličini transformacije. Svaka kvantizacija rezultata, zaokruživanje ili odsijecanje, nastala prilikom svođenja rezultata množenja u osnovni dinamički opseg ili zbog izbjegavanja prekoračenja unosi određenu grešku u krajnji rezultat. Skaliranje međurezultata utiče na doprinos pojedinačnih grešaka ukupnoj grešci tako da umanjuje uticaj grešaka nastalih u ranijim fazama algoritma, tj., prilikom izračunavanja početnih stepeni algoritma brze Furijeove transformacije. Iako skaliranje međurezultata doprinosi smanjivanju doprinosa grešaka kvantizacije ukupnoj grešci izračunavanja brze Furijeove transformacije ono takođe dovodi i do smanjivanja dinamičkog opsega signala (varijanse signala).

Porast vrijednosti maksimalnog elementa niza međurezultata $\{x_k(n)\}$ od stepena do stepena FFT algoritma zavisi prvenstveno od statističkih osobina ulaznog signala – ulaznog niza $\{x_k(n)\}$. Porast maksimalne vrijednosti međurezultata FFT algoritma je uglavnom manji od graničnog (udvostručenje maksimalne vrijednosti nakon svakog stepena) a često je nedovoljan da prouzrokuje prekoračenje. To znači da je preventivno skaliranje međurezultata dobijenih prilikom izračunavanja FFT algoritma sa dva ponekad suvišno. Ukoliko je neohodno postići ukupno skaliranje od N tada je bolje ta suvišna skaliranja (skaliranja u slučajevima kad ne bi došlo do prekoračenja) obaviti na kraju. Ukupno pojačanje FFT algoritma neće se promijeniti ali će se varijansa ukupne greške kvantizacije smanjiti jer će neophodna normalizacija (skaliranje nadole) biti urađena na kraju (poslije posljednjeg stepena).

Prilikom prethodne analize (paragraf 2.3) pretpostavljeno je da su elementi ulaznog niza ograničeni po modulu (niz kompleksnih brojeva) s gornje strane i da je ta granica 1 ($|x(n)| < 1$). Svrha ovog ograničenja je izbjegavanje prekoračenja prilikom izvođenja kompleksnog množenja. Dodatno je potrebno skalirati međurezultate sa dva nakon svakog stepena FFT algoritma. Ukoliko se koristi neka od metoda kod koje se ne obavlja preventivno automatsko skaliranje međurezultata FFT algoritma za dva ova metoda u opštem slučaju ne obezbjeđuje izbjegavanje prekoračenja. S obzirom da ovakve metode dozvoljavaju porast vrijednosti realnih i imaginarnih komponenti do punog opsega (uslov $\text{Re}(x), \text{Im}(x) < 1$) postavljanje uslova za amplitudu ulaznog signala ne može donijeti zadovoljavajuće rješenje te je potrebno koristiti druge metode za izbjegavanje prekoračenja.

3. Postojeće metode za smanjenje greške unesene prilikom izračunavanja brze Furijeove transformacije korištenjem aritmetike nepokretnog zarez a i analiza njihove tačnosti

U prethodnom poglavlju je izvršena analiza uzroka nastanka i prostiranja grešaka kvantizacije i predstavljeni neki od načina za izbjegavanje prekoračenja opsega prilikom izračunavanja diskretne Furijeove transformacije korištenjem algoritma brze Furijeove transformacije. Analiza je pokazala da način izbjegavanja pojave prekoračenja utiče na odnos signal-šum izlaznog signala. U nastavku su prikazani neki od metoda za izbjegavanje prekoračenja kao i rezultati korištenja tih metoda na primjerima nekih karakterističnih signala.

3.1 Kratak prikaz postojećih metoda

3.1.1 Implementacija FFT algoritma automatskim skaliranjem

Ova metoda je najjednostavnija za realizaciju ali ne daje zadovoljavajući rezultat prilikom izračunavanja dugačkih transformacija. S obzirom da je široko poznata biće korištena kao referenca.

Ideja algoritma je da se prekoračenje prilikom računanja FFT algoritma izbjegava tako što se izlazni rezultati računanja svakog baterflaja dijele sa dva. Da bi se izbjeglo prekoračenje prilikom izračunavanja kompleksnog množenja potrebno je da ulazni signal zadovoljava uslov da je zbir kvadrata realne i imaginarne komponente manji ili jednak jedan.

Ovaj uslov nije uvijek automatski ispunjen (na primjer kad se kompleksna FFT koristi za računanje transformacije dvije realne sekvence) pa je potrebno izvršiti ulazno skaliranje za $\sqrt{2}$ (množenje sa $1/\sqrt{2}$). Da bi se obezbijedio korektan rezultat, izlazne vrijednosti je potrebno pomnožiti sa $\sqrt{2}$ (ulazno skaliranje nije dio FFT algoritma). Konačan rezultat je podijeljen sa N (veličina transformacije).

3.1.2. Metode zasnovane na blok pokretnom zarezu

Mana prethodne metode je u tome što porast vrijednosti međurezultata ne mora da izazove prekoračenje u svakom prolazu (stepenu) FFT algoritma, tj., u zavisnosti od statistike signala, vrijednosti međurezultata mogu ostati u opsegu i bez izvršenog skaliranja za dva što znači da bi automatsko skaliranje u tom slučaju dovelo do nepotrebnog gubitka informacije.

Ova grupa metoda je zasnovana na ideji da se skaliranje obavlja samo ako postoji mogućnost prekoračenja u narednom prolazu [5][6][7]. Na taj način se smanjuje broj nepotrebnih skaliranja za dva u odnosu na prethodnu metodu. Ukupno skaliranje zavisi

od statističkih osobina signala i njegova vrijednost nije poznata sve do završetka izračunavanja.

Pri primjeni ove metode svaki prolaz FFT algoritma se tretira kao blok podataka kome je pridružen jedinstven eksponent. Ovakav način predstavljanja brojeva podjeća na predstavu brojeva u formatu pokretnog zareza pa se naziva predstava u obliku blok-pokretnog zareza (engl. block floating point). Kod predstave brojeva u obliku blok-pokretnog zareza vrijednost svakog elementa iz grupe (bloka) vrijednosti ima značenje mantise kao kod reprezentacije brojeva u obliku pokretnog zareza a svi elementi grupe imaju zajednički eksponent. Nakon svakog stepena u kojem je došlo do dijeljenja sa dva eksponent međurezultata prethodnog stepena se uvećava za jedan. Nakon završetka izračunavanja FFT algoritma potrebno je izvršiti dodatno skaliranje u cilju svodjenja vrijednosti rezultata u potreban opseg (vrijednost potrebnog konačnog skaliranja je fiksna broj koji zavisi od konkretnog algoritma).

Metode iz ove grupe se međusobno razlikuju po kriterijumu odlučivanja o mogućnosti pojave prekoračenja i o načinu izbjegavanja prekoračenja do kojeg može doći kod kompleksnog množenja.

3.1.2.1 Dijeljenje sa dva ako postoji mogućnost prekoračenja u narednom stepenu FFT-a

Kod ove metode kriterijum prekoračenja je ispunjen ako postoji barem jedan međurezultat koji ima realnu ili imaginarnu komponentu po apsolutnoj vrijednosti veću od polovine opsega i tada se svi međurezultati dijele sa dva.

U DSP arhitekturama se u cilju otkrivanja mogućeg prekoračenja uvodi automatsko poređenje vrijednosti sa polovinom opsega (0.5 za pozitivne, -0.5 za negativne) i ako se otkrije porast vrijednosti međurezultata preko polovine opsega posebna oznaka namijenjena otkrivanju potencijalnog prekoračenja se postavlja ("Scaling bit" kod nekih Motorolinskih DSP-ova). Uobičajena realizacija oznake prekoračenja podrazumjeva da se oznaka briše samo posebnom instrukcijom (u engleskom jeziku ovakve oznake se nazivaju "latched flag") a postavlja samo ukoliko je vrijednost koja je upisana u memoriju veća od polovine opsega. Nakon završenog računanja jednog stepena FFT-a ispituje se oznaka prekoračenja i u zavisnosti od njene vrijednosti svi međurezultati se dijele sa dva u narednom stepenu. Prije početka pojedinog stepena FFT algoritma briše se oznaka potencijalnog prekoračenja.

Ova metoda daje bolje rezultate prilikom izračunavanja FFT algoritma nego prethodna metoda ali se i kod nje može desiti da je skaliranje nepotrebno obavljeno (moguće je otkriti samo postojanje potrebnog uslova za prekoračenje, ali ne i dovoljnog uslova). Kao i kod prethodne metode prilikom svakog nepotrebnog dijeljenja međurezultata sa dva dolazi do bespovratnog gubljenja jednog bita informacionog sadržaja.

3.1.2.2 Dijeljenje sa dva ako postoji mogućnost prekoračenja u narednom stepenu FFT-a i izbjegavanje prekoračenja kod kompleksnog množenja skaliranjem unutar baterflaja

U prethodnoj metodi se ne vodi računa o mogućem prekoračenju unutar baterflaja, tj., prekoračenju kod izvođenja kompleksnog množenja. Na taj način se kod određenih signala postiže bolji rezultat nego ako se koristi neki mehanizam za izbjegavanje prekoračenja unutar baterflaja. Greške koje se zbog toga mogu pojaviti smanjuju mogućnost njene primjene u nekim aplikacijama pa se ova metoda često koristi u kombinaciji sa nekim načinom za izbjegavanje prekoračenja [6],[7].

Prekoračenje unutar baterflaja se može izbjeći skaliranjem unutar baterflaja za maksimalan rezultat kompleksnog množenja - $\sqrt{2}$. Umjesto skaliranja unutar baterflaja sa maksimalnom vrijednošću kompleksnog množenja koristi se skaliranje sa polovinom vrijednosti maksimalnog prekoračenja - $\frac{2}{(1+\sqrt{2})}$ (rezultat kompleksnog množenja kao i

drugi sabirak kompleksnog sabiranja se množe sa ~ 0.828) [6],[7]. Uz otkrivanje mogućeg prekoračenja ovakvo skaliranje omogućuje izbjegavanje prekoračenja kod izračunavanja FFT algoritma. S obzirom da ovo skaliranje nije dio transformacije ukupno skaliranje nastalo kao rezultata skaliranja unutar baterflaja je potrebno kompenzovati nakon izračunavanja FFT algoritma. Da bi se izbjeglo dodatno množenje rezultata FFT algoritma recipročnom vrijednošću ekvivalentnog skaliranja koeficijent skaliranja unutar baterflaja se bira tako da je ekvivalentno skaliranje stepen broja dva (npr. ako se skaliranje izvodi u sedam prolaza koeficijent skaliranja je $0.25^{\frac{1}{7}}$ što približno iznosi 0.820 što je nešto manje nego potrebnih ~ 0.828 , tako da je ovako modifikovano skaliranje restriktivnije od potrebnog skaliranja). Na taj način se dodatno skaliranje unutar baterflaja kompenzuje množenjem sa stepenom broja dva što je ekvivalentno aritmetičkom pomjeranju ulijevo. Ovakvo skaliranje se ne obavlja u prva dva stepena FFT algoritma jer se u njima vrše samo trivijalna množenja. Takođe je moguće pomnožiti faktore zaokreta W_N^{kn} sa faktorom skaliranja (uglavnom se koriste unaprijed izračunate vrijednosti) u cilju smanjenja broja dodatnih realnih množenja prilikom izračunavanja baterflaja sa četiri na dva.

3.1.2.3 Automatsko dijeljenje sa dva izlaznih vrijednosti i množenje ulaznih vrijednosti sa dva ako ne postoji mogućnost prekoračenja u narednom stepenu FFT-a

Prethodne dvije metode se mogu jednostavno realizovati na digitalnim signalnim procesorima kod kojih oznaka prekoračenja otkriva svaku pojavu prekoračenja polovine opsega. Ukoliko to nije slučaj bilo bi potrebno tražiti maksimalne vrijednosti niza međurezultata i ispitati da li apsolutne vrijednosti njegove realne i imaginarne komponente prelaze polovinu opsega što znatno povećava vrijeme obrade.

Ponekad se oznaka prekoračenja realizuje tako da otkriva samo prekoračenje do koga bi došlo da nije bilo dijeljenja sa dva (aritmetičkog pomjeraja za jedno mjesto

udesno) prilikom smještanja rezultata u memoriju (P oznaka kod MAS DSP-a)[8]. Ovakva izvedba oznake potencijalnog prekoračenja omogućava otkrivanje da li je dijeljenje sa dva bilo potrebno pa je potrebno stalno (automatski) skalirati vrijednosti međurezultata za dva. Ukoliko je dijeljenje sa dva bilo nepotrebno ulazne vrijednosti narednog stepena se množe sa dva (aritmetičko pomjeranje ulijevo). Aritmetičko pomjeranje ulijevo se obavlja prilikom čitanja vrijednosti iz memorije i, kao i aritmetičko pomjeranje udesno prilikom upisa vrijednosti u memoriju, ne zahtijeva dodatne instrukcijske cikluse.

3.1.2.4 Automatsko dijeljenje sa dva izlaznih vrijednosti i množenje ulaznih vrijednosti sa dva ako ne postoji mogućnost prekoračenja u narednom stepenu FFT-a uz izbjegavanje prekoračenja dijeljenjem sa dva rezultata množenja ukoliko postoji mogućnost prekoračenja kod izvođenja kompleksnog množenja

Kod prethodne metode se kao i kod metode izložene u 3.1.2.1 ne vodi računa o mogućnosti prekoračenja prilikom izvođenja kompleksnog množenja. S obzirom da metoda za izbjegavanje prekoračenja prilikom izvođenja kompleksnog množenja izložena u 3.1.2.2 zahtijeva velik broj dodatnih operacija u praktičnim realizacijama prethodne metode ova vrsta prekoračenja se izbjegava dijeljenjem ulaznih vrijednosti narednog stepena sa dva ukoliko je vrijednost realne ili imaginarne komponente nekog od međurezultata po apsolutnoj vrijednosti veća od zadate vrijednosti praga. Teoretska vrijednost praga je $\frac{\sqrt{2}}{2}$ (~ 0.707), ali se uglavnom koristi vrijednost praga od 0.5 zbog jednostavnije praktične izvedbe poređenja sa pragom.

3.2 Kriterijumi tačnosti izračunavanja

Najčešći u praksi korišteni kriterijum za ocjenu tačnosti izračunavanja nekog algoritma je odnos signal-šum. S obzirom da je kod izračunavanja FFT algoritma šum nastao kvantizacijom vrijednosti međurezultata ovaj kriterijum se uobičajeno naziva odnos signal-kvantizacioni šum (engl. signal to quantization noise ratio). Odnos signal-kvantizacioni šum se računa po sljedećoj formuli

$$SQNR = \frac{\sum_{k=0}^{N-1} (X(k))^2}{\sum_{k=0}^{N-1} (X_q(k) - X(k))^2} \quad (59)$$

gdje je su sa $X(k)$ označani elementi niza izlaznih vrijednosti “tačno” izračunate transformacije (obično izračunate korištenjem aritmetike pokretnog zarez u jednostrukoj

ili dvostrukoj preciznosti), sa $X_q(k)$ označeni elementi niza izlaznih vrijednosti transformacije izračunate posmatranom metodom korištenjem aritmetike nepokretnog zarez a N je dužina transformacije, odnosno broj elemenata niza izlaznih vrijednosti FFT algoritma. Odnos signal-kvantizacioni šum se obično iskazuje u decibelima i pri tom se koristi sljedeća formula

$$SQNR_{dB} = 10 \log \left(\frac{\sum_{k=0}^{N-1} (X(k))^2}{\sum_{k=0}^{N-1} (X(k) - X_q(k))^2} \right). \quad (60)$$

Može se primjetiti da ovaj kriterijum daje informacije o veličini ukupne greške izračunavanja ali ne daje dovoljno informacija o veličini pojedinačnih grešaka. Zbog toga se pri analizi tačnosti koristi i kriterijum apsolutne vrijednosti maksimalne greške

$$\max(|X_q(k) - X(k)|) \quad (61)$$

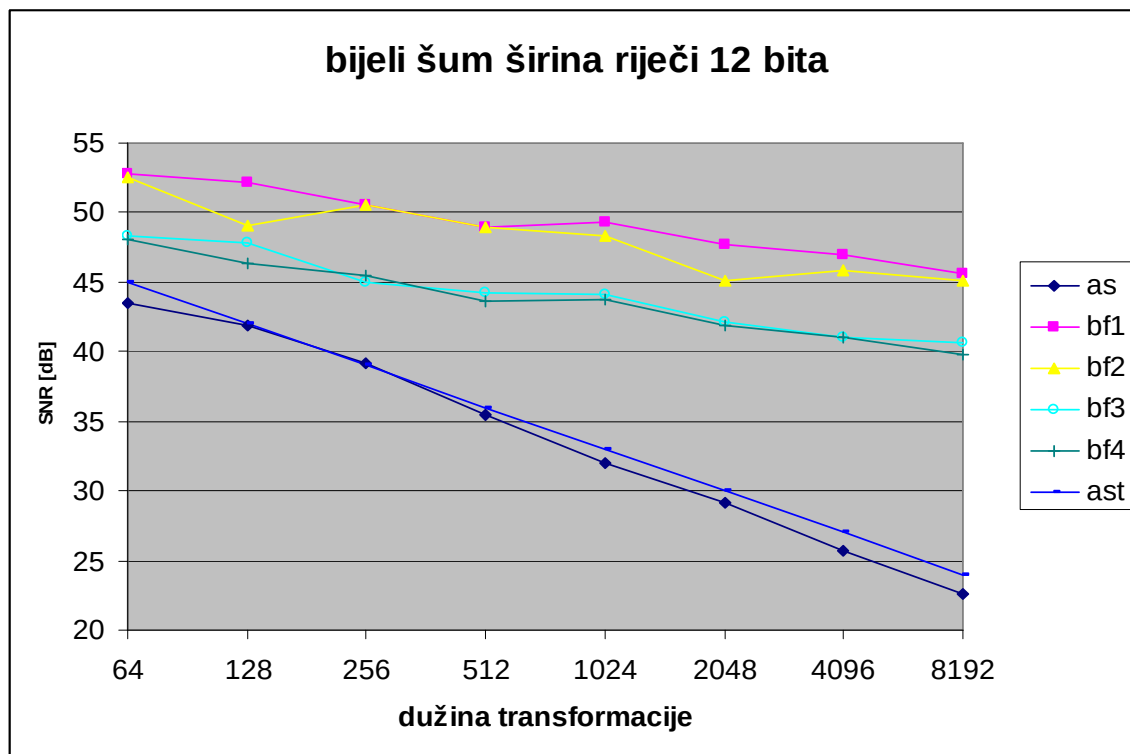
koji omogućava otkrivanje pojedinačnih grešaka koje odstupaju od srednje greške, ali se zbog veličine transformacije (velikog broja vrijednosti koje učestvuju u izračunavanju) ne mogu (jednostavno) otkriti pomoću odnosa signal-šum kvantizacije. Takve greške ponekad znaju biti sistemske prirode gdje se velika greška pojavljuje uvijek u istim elementima izlaznog niza (najčešće na nultoj poziciji). U slučaju primjene brzih transformacija u algoritmima za kompresiju audio signala takva greška može imati oblik ekvidistantnih impulsa i ponekad ju je moguće otkriti slušanjem. U slučaju primjene FFT algoritma u okviru OFDM algoritma greška tog tipa bi odgovarala konstantnoj grešci na odgovarajućem nosiocu.

3.3 Analiza tačnosti postojećih metoda na primjerima nekih karakterističnih signala

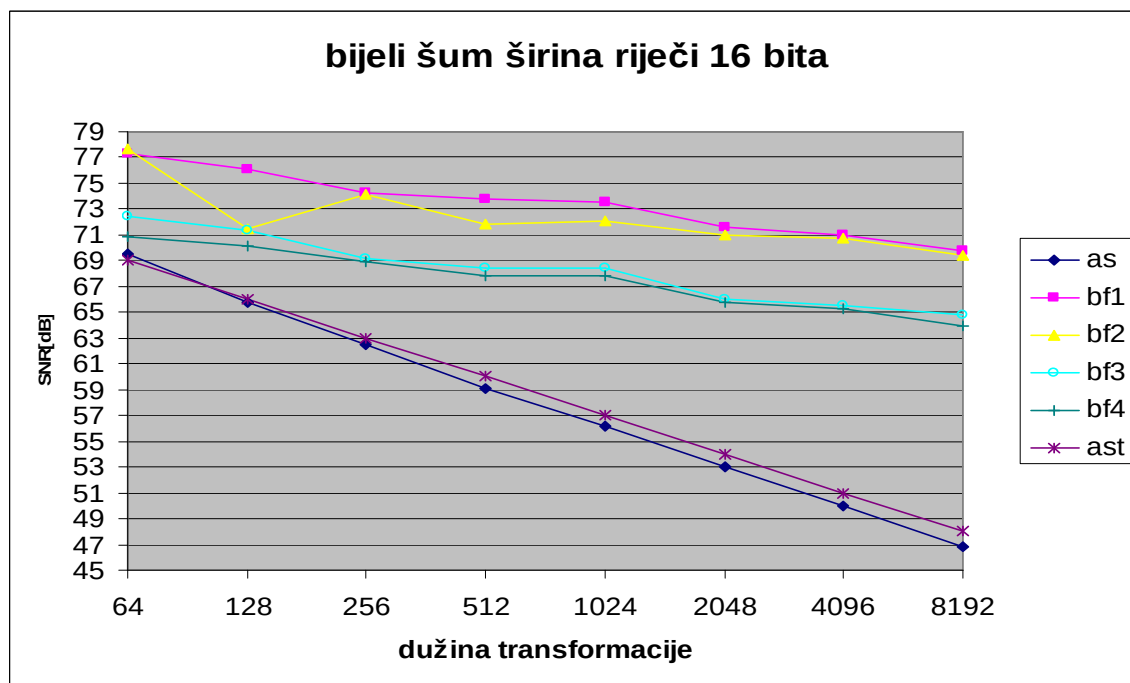
Tačnost metoda prikazanih u prethodnim paragrafima će biti analizirana na primjerima signala bijelog šuma i više-tonskih signala sa 32 odnosno 64 ravnomjerno raspoređena (ekvidistanti u spektralnom domenu) sinusna signala. Zbog kompaktnijeg prikaza izložene metode će biti označene sljedećim skraćenicama:

- as – metoda automatskog skaliranja
- ast – teoretska vrijednost za metodu automatskog skaliranja (prema formuli iz 2.3)
- bf1 – metoda izložena u 3.1.2.1
- bf2 – metoda izložena u 3.1.2.2
- bf3 – metoda izložena u 3.1.2.3
- bf4 – metoda izložena u 3.1.2.4

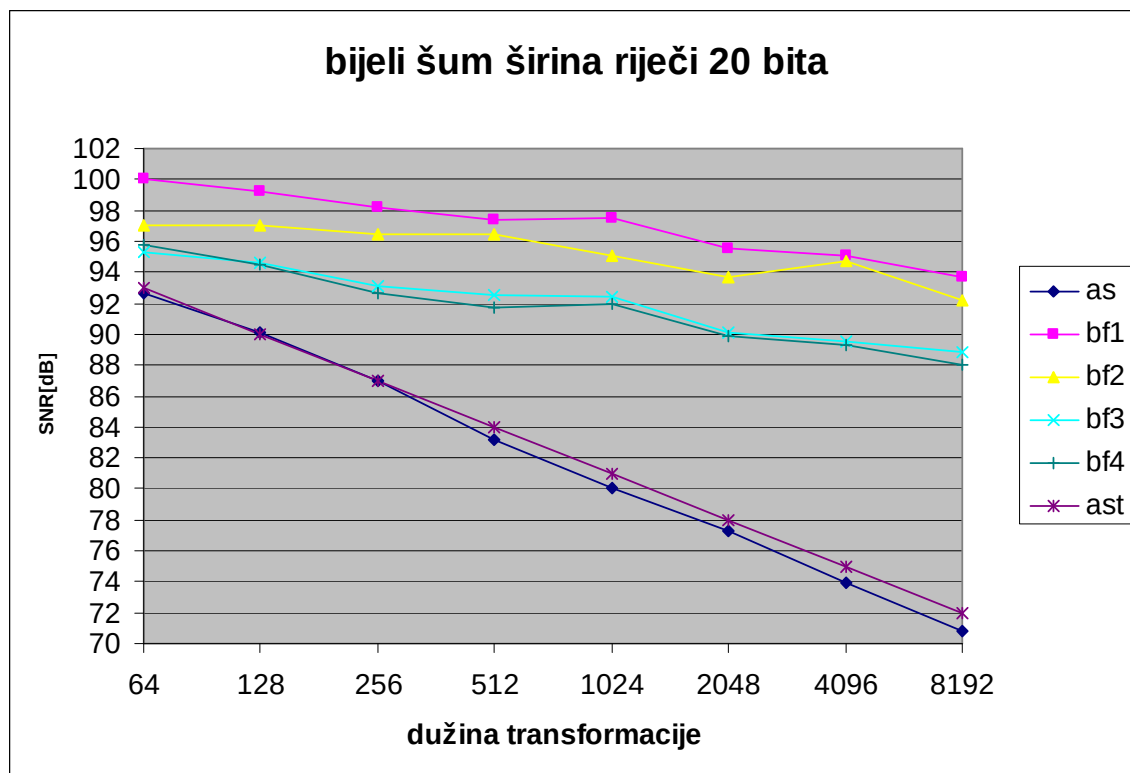
Odnos signal-kvantizacioni šum za širine riječi od 12, 16 i 20 bita za pomenute metode je prikazan na slikama 3.1, 3.2 i 3.3



Slika 3.1 Odnos signal-šum za signal bijelog šuma i širinu riječi od 12 bita

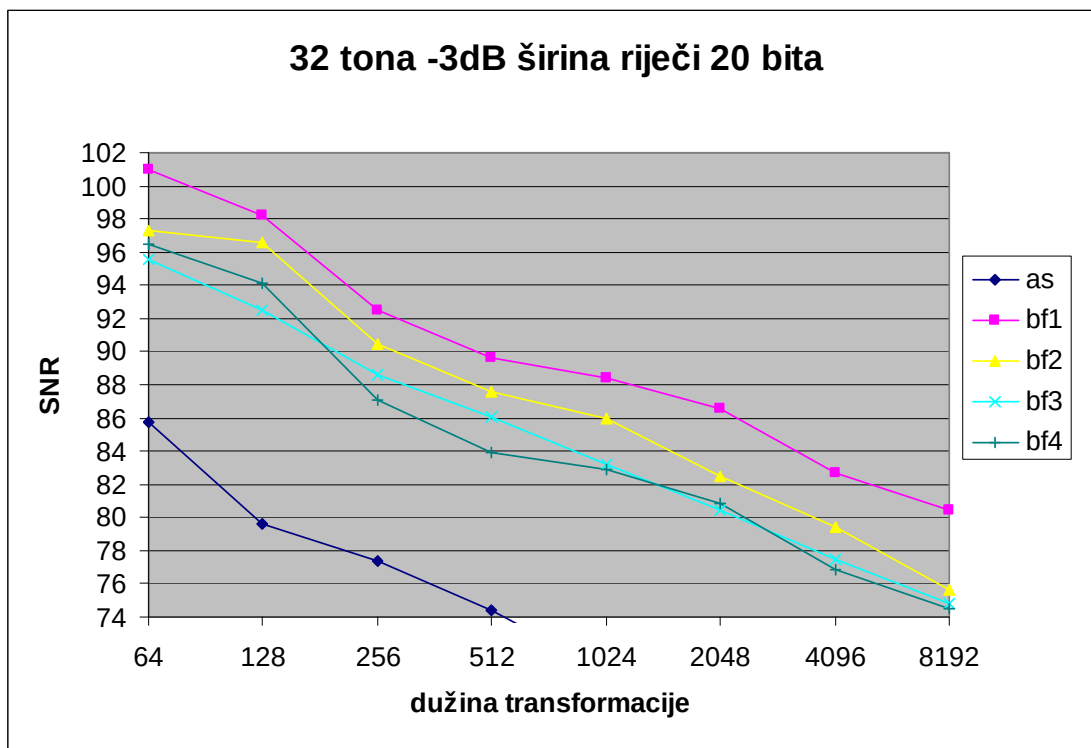


Slika 3.2 Odnos signal-šum za signal bijelog šuma i širinu riječi od 16 bita

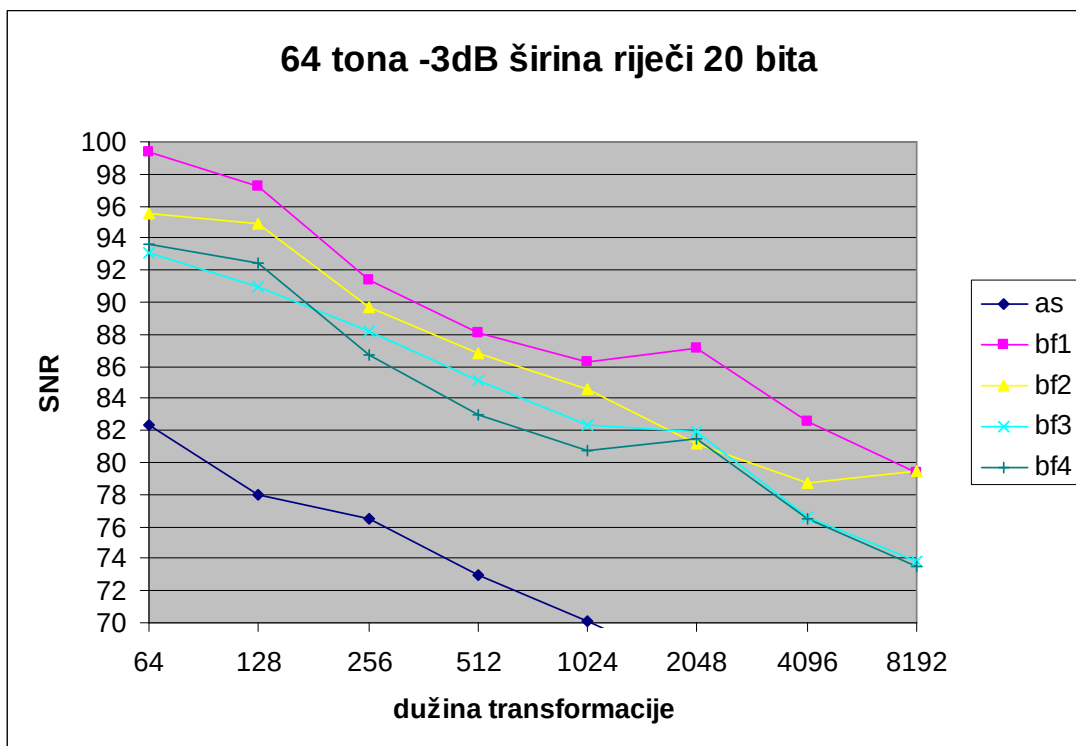


Slika 3.3 Odnos signal-šum za signal bijelog šuma i širinu riječi od 20 bita

Na prethodnim slikama se može primjetiti da rezultati metode automatskog skaliranja ne odstupaju mnogo od teoretskih rezultata. Takođe se može uočiti da metoda automatskog skaliranja daje mnogo manji odnos signal-šum kvantizacije nego ostale metode.



Slika 3.4 Odnos signal-šum za multitonski signal (32 tona) i širinu riječi od 20 bita



Slika 3.5 Odnos signal-šum za multitonski signal (64 tona) i širinu riječi od 20 bita

Apsolutna greška nastala prilikom izračunavanja brze Furijeove transformacije opisanim metodama prikazana je tabelarno tako da brojčana vrijednost u određenoj koloni za posmatranu konfiguraciju odgovara broju izlaznih vrijednosti (realna ili imaginarna komponenta kompleksnog broja) čiji broj tačnih bita je jednak indeksu kolone (na primjer, u tabeli 3.1 vrijednost 4 u drugom redu peta kolona označava da su za konfiguraciju 'as' četiri izlazne vrijednosti imale samo 12 tačnih bita).

	10	11	12	13	14	15	16	17	18	19
as	0	0	4	28	36	28	17	7	5	3
bf1	0	0	0	0	1	11	37	37	21	21
bf2	0	0	0	0	2	39	37	27	10	13
bf3	0	0	0	0	10	40	31	24	14	9
bf4	0	0	0	0	8	30	39	26	13	12

Tabela 3.1 Distribucija netačnih bita za signal 32 tona i N=64

	10	11	12	13	14	15	16	17	18	19
as	0	0	8	59	85	52	25	20	3	4
bf1	0	0	0	0	0	6	51	83	53	63
bf2	0	0	0	0	0	14	73	77	46	46
bf3	0	0	0	0	5	49	85	50	35	32
bf4	0	0	0	0	0	39	79	51	39	48

Tabela 3.2 Distribucija netačnih bita za signal 32 tona i N=128

	10	11	12	13	14	15	16	17	18	19
as	0	9	111	164	120	61	25	7	8	7
bf1	0	0	0	1	44	143	152	92	47	33
bf2	0	0	0	5	87	173	105	71	34	37
bf3	0	0	1	18	109	153	117	62	26	26
bf4	0	0	0	32	152	134	90	51	29	24

Tabela 3.3 Distribucija netačnih bita za signal 32 tona i N=256

	10	11	12	13	14	15	16	17	18	19
as	0	11	229	338	214	117	53	33	18	11
bf1	0	0	0	7	64	255	298	219	94	87
bf2	0	0	0	7	161	333	230	162	61	70
bf3	0	0	2	23	200	312	226	143	61	57
bf4	0	0	3	81	247	289	192	100	66	46

Tabela 3.4 Distribucija netačnih bita za signal 32 tona i N=512

	10	11	12	13	14	15	16	17	18	19
as	0	27	461	650	425	204	134	83	29	35
bf1	0	0	0	3	53	376	660	450	261	245
bf2	0	0	0	2	168	614	570	337	183	174
bf3	0	0	1	38	339	636	456	300	144	134
bf4	0	0	1	40	380	662	457	250	135	123

Tabela 3.5 Distribucija netačnih bita za signal 32 tona i N=1024

	10	11	12	13	14	15	16	17	18	19
as	0	0	3	35	36	28	13	5	5	3
bf1	0	0	0	0	0	8	41	33	17	29
bf2	0	0	0	0	2	28	44	23	17	14
bf3	0	0	0	0	11	35	39	24	11	8
bf4	0	0	0	0	7	34	33	29	11	14

Tabela 3.6 Distribucija netačnih bita za signal 64 tona i N=64

	10	11	12	13	14	15	16	17	18	19
as	0	0	11	64	62	59	32	16	5	7
bf1	0	0	0	0	0	5	53	87	56	55
bf2	0	0	0	0	0	14	78	72	39	53
bf3	0	0	0	0	9	42	89	56	32	28
bf4	0	0	0	0	0	48	79	67	27	35

Tabela 3.7 Distribucija netačnih bita za signal 64 tona i N=128

	10	11	12	13	14	15	16	17	18	19
as	0	3	127	152	106	64	33	16	5	6
bf1	0	0	0	2	44	118	155	90	55	48
bf2	0	0	0	6	70	163	112	85	36	40
bf3	0	0	1	11	91	167	114	70	32	26
bf4	0	0	2	19	126	167	96	49	27	26

Tabela 3.8 Distribucija netačnih bita za signal 64 tona i N=256

	10	11	12	13	14	15	16	17	18	19
as	0	14	264	314	200	114	60	30	15	13
bf1	0	0	0	10	78	272	291	174	101	98
bf2	0	0	0	8	140	342	264	143	59	68
bf3	0	0	2	19	202	314	232	134	65	56
bf4	0	0	2	65	284	294	185	99	49	46

Tabela 3.9 Distribucija netačnih bita za signal 64 tona i N=512

	10	11	12	13	14	15	16	17	18	19
as	0	14	458	678	426	227	135	60	25	25
bf1	0	0	0	3	100	497	630	385	214	219
bf2	0	0	0	8	179	657	550	312	172	170
bf3	0	0	2	29	342	656	472	281	139	127
bf4	0	0	4	79	447	653	432	218	107	108

Tabela 3.10 Distribucija netačnih bita za signal 64 tona i N=1024

Analizom grešaka nastalih prilikom izračunavanja brze Furijeove transformacije iskazanih u obliku signal-šum kvantizacije i u obliku apsolutne greške za opisane metoda može se primjetiti da metode zasnovane na blok pokretnom zarezu unose mnogo manju grešku od metode automatskog skaliranja. Najbolji rezultat daje metoda bf1 kod koje se ne obavlja izbjegavanje prekoračenja kod kompleksnog množenja. Metoda bf2 daje bolje rezultate od metoda bf3 i bf4 koje su pokazale približne rezultate (metoda bf3 je dala malo bolje rezultate).

4. Prijedlog nove metode za smanjenje greške unesene prilikom izračunavanja brze Furijeove transformacije korištenjem aritmetike nepokretnog zareza

4.1 Opis metode

4.1.1 Izbjegavanje prekoračenja kod sabiranja

Kao i metode opisane u poglavlju 3.1.2. i ova metoda je zasnovana na predstavljanju brojeva u obliku blok pokretnog zareza. Kod metoda opisanih u poglavljima 3.1.2.1, 3.1.2.2 skaliranje se obavlja ako postoji mogućnost prekoračenja. Kod metoda opisanih u poglavljima 3.1.2.3 i 3.1.2.4 dijeljenje sa dva se obavlja nakon svakog stepena (automatski) a ukoliko je dijeljenje bilo nepotrebno međurezultati se množe sa dva. Za razliku od prethodno opisane dvije grupe metoda kod predložene metode skaliranje se obavlja samo kad je to stvarno neophodno i to tako da se istovremeno upisuju u memoriju neskaliрана i skalirana vrijednost. Nakon izračunatog stepena FFT-a ispituje se da li se desilo prekoračenje i ukoliko je to slučaj računanje se nastavlja sa međurezultatima koji su podijeljeni sa dva. Pri tom se interni brojač koji ima ulogu zajedničkog eksponenta uvećava za jedan.

Prethodne dvije grupe metoda se zasnivaju na korištenju oznake prekoračenja polovine opsega. Predložena metoda se zasniva na korištenju oznake prekoračenja. Oznaka prekoračenja se koristi za utvrđivanje da li je prilikom upisa u memoriju došlo do prekoračenja. Pomenuta oznaka se briše posebnom instrukcijom prije svakog stepena FFT-a, a postavlja se prilikom pokušaja upisa u memoriju broja koji se nalazi izvan dozvoljenog opsega. Jednom postavljena oznaka može se obrisati samo primjenom za to namjenjene instrukcije (eng. "latched flag").

U odnosu na metode izložene u prethodnom poglavlju primjena ove metoda zahtijeva dodatnu memoriju zbog potrebe smještanja i skaliranih i ne skaliranih vrijednosti. Ukoliko se obavlja zaokruživanje rezultata prije smještanja u memoriju tada je to potrebno uraditi za obe vrijednosti – podijeljenu sa dva i originalnu (akumulatori obično imaju veću bitsku širinu nego memorija). Primjena metode zahtijeva dvije dodatne operacije po vrijednosti (i za realni i za imaginarni dio) ukoliko se obavlja zaokruživanje rezultata prije smještanja u memoriju odnosno jednu dodatnu operaciju (dodatni upis) ukoliko se ne obavlja zaokruživanje rezultata prije smještanja u memoriju.

U slučaju da se za naredni stepen FFT-a ne koriste vrijednosti smještene u očekivanom bloku memorije zamjena blokova se obično svodi na manipulaciju vrijednostima pokazivača. U određenim slučajevima neophodno je kopirati blok sa korektnim vrijednostima na željeno mjesto što može dodatno usporiti izvršenje.

4.1.2 Izbjegavanje prekoračenja kod kompleksnog množenja

Kao i za prethodno opisane metode i za ovu metodu može se u nekim primjenama javiti potreba za izbjegavanjem prekoračenja kod kompleksnog množenja. Primjenom neke od metode za izbjegavanje prekoračenja kod kompleksnog množenja dolazi do smanjenja tačnosti prilikom izračunavanja Furijeove transformacije određenih signala, ali se i izbjegavaju greške koje se mogu pojaviti usljed pomenutog prekoračenja.

4.1.2.1 Izbjegavanje prekoračenja kod kompleksnog množenja dodatnim upisom u memoriju

Na ovaj način se izbjegavanje prekoračenja kod kompleksnog množenja obavlja na isti način kao i izbjegavanje prekoračenja kod sabiranja – istovremenim upisom u memoriju neskalirane i skalirane vrijednosti. Broj vrijednosti koje se izračunavaju kompleksnim množenjem je dvostruko manji od broja vrijednosti nastalih kompleksnim sabiranjem te je potrebno i dvostruko manje memorije za zapis dodatnih vrijednosti. Upisom rezultata kompleksnog množenja u memoriju svaki stepen FFT-a (osim trivijalnih) se dijeli na dio u kojem se izračunavaju sva kompleksna množenja i dio u kojem se izračunavaju kompleksna sabiranja. Dodatna memorija koja se koristi za izbjegavanje prekoračenja kod sabiranja se može iskoristiti za izbjegavanje prekoračenja kod kompleksnog množenja jer su na ovaj način te operacije razdvojene.

4.1.2.2 Izbjegavanje prekoračenja kod kompleksnog množenja skaliranjem unutar baterflaja

Na sličan način kao u metodi opisanoj u poglavlju 3.1.2.2 izbjegavanje prekoračenja kod kompleksnog množenja se može obavljati skaliranjem unutar baterflaja

polovinom vrijednosti maksimalnog prekoračenja - $\frac{2}{(1+\sqrt{2})}$ (rezultat kompleksnog

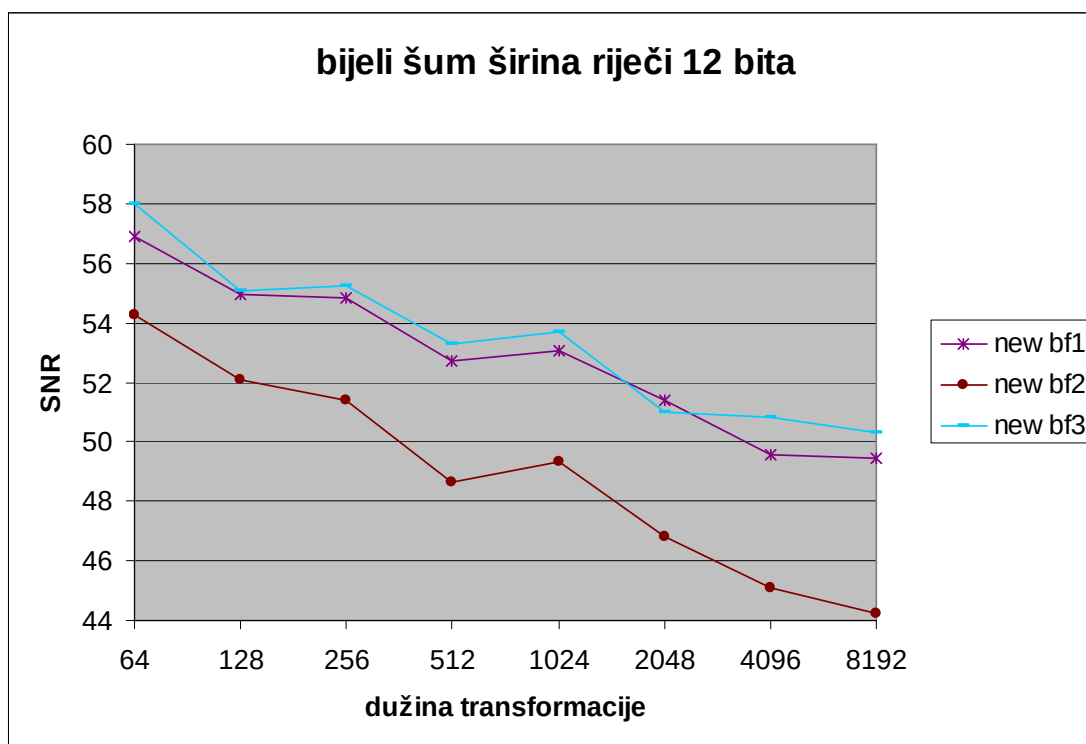
množenja kao i drugi sabirak kompleksnog sabiranja se množe sa ~ 0.828 pri čemu je potrebno rezultate FFT algoritma pomnožiti recipročnom vrijednošću ekvivalentnog skaliranja) ili takvom vrijednošću da ekvivalentno skaliranje odgovara stepenu broja dva (u ovom slučaju rezultate FFT algoritma je potrebno pomnožiti odgovarajućim stepenom broja dva što odgovara pomjeranju u lijevo).

4.2 Analiza greške kvantizacije u DSP aplikaciji

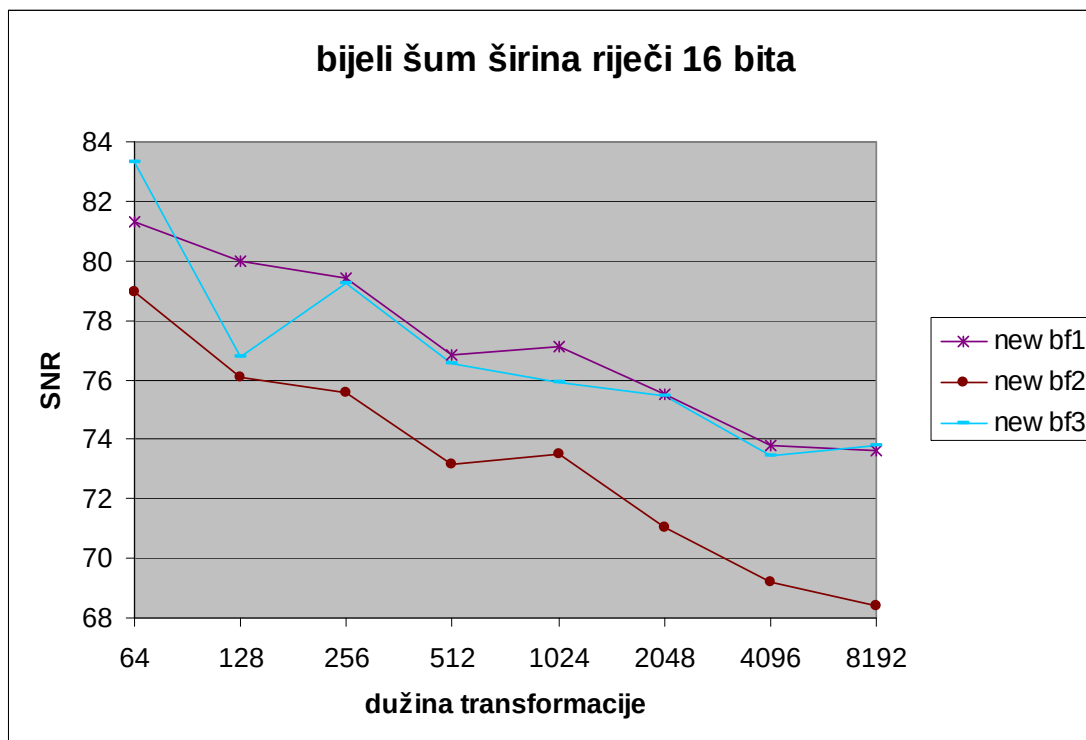
Tačnost metoda varijanti predložene metode prikazanih u prethodnim paragrafima će biti analizirana na primjerima signala bijelog šuma i više-tonskih signala sa 32 odnosno 64 ravnomjerno raspoređena (ekvidistanti u spektralnom domenu) sinusna signala. Zbog kompaktnijeg prikaza izložene metode će biti označene sljedećim skraćenicama:

- New bf1 – metoda izložena u 4.1.1
- New bf2 – metoda izložena u 4.1.2.1
- New bf1 – metoda izložena u 4.1.2.2

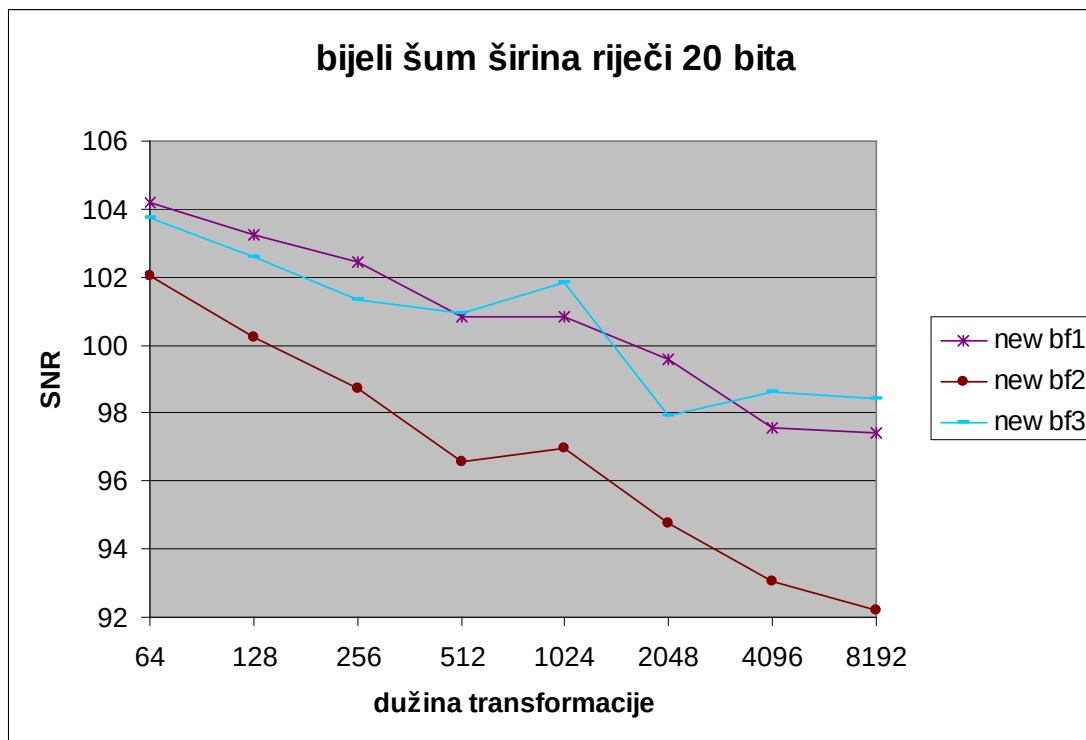
Odnos signal-kvantizacioni šum za širine riječi od 12, 16 i 20 bita za pomenute metode je prikazan na slikama 4.1, 4.2 i 4.3 za signal bijelog šuma. Odnos signal-kvantizacioni šum za širinu riječi od 20 bita za pomenute metode je prikazan na slikama 4.4 i 4.5 za multitonske signale.



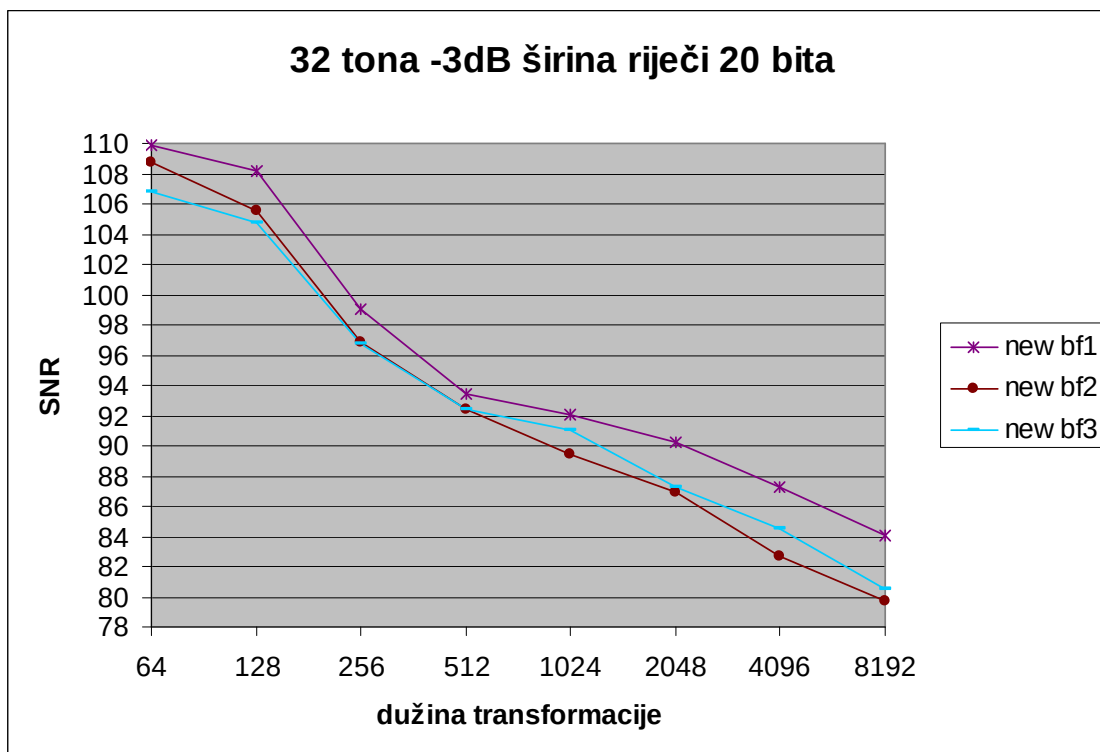
Slika 4.1 Odnos signal-šum za signal bijelog šuma i širinu riječi od 12 bita



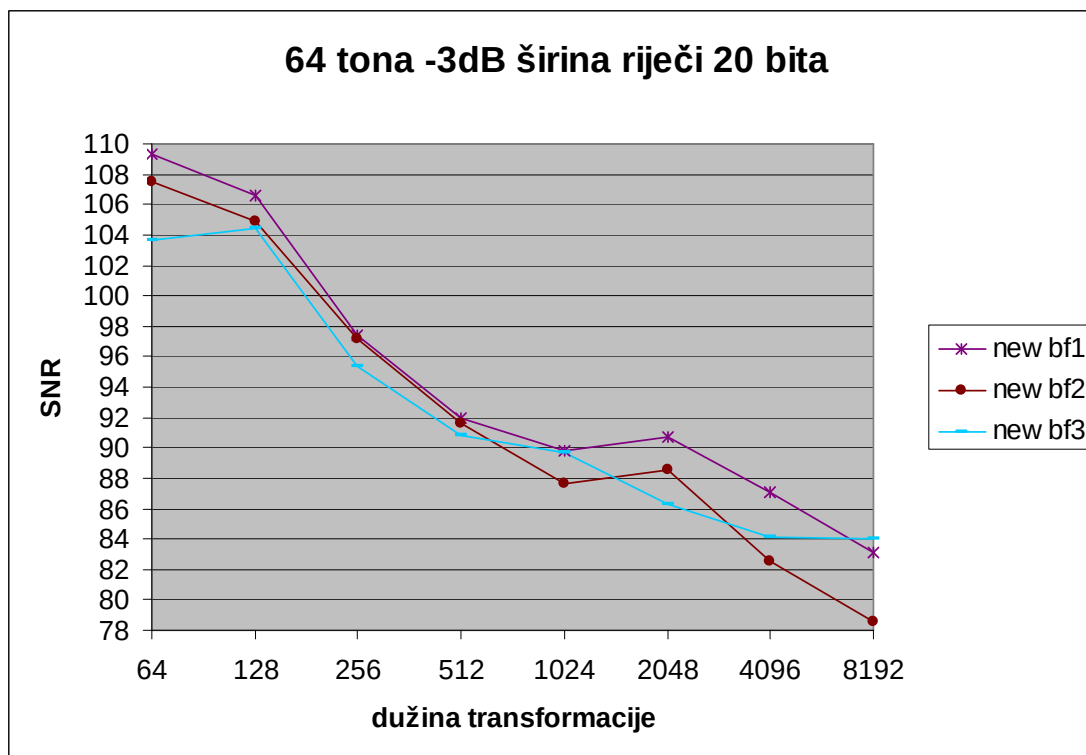
Slika 4.2 Odnos signal-šum za signal bijelog šuma i širinu riječi od 16 bita



Slika 4.3 Odnos signal-šum za signal bijelog šuma i širinu riječi od 20 bita



Slika 4.4 Odnos signal-šum za multitonski signal (32 tona) i širinu riječi od 20 bita



Slika 4.5 Odnos signal-šum za multitonski signal (64 tona) i širinu riječi od 20 bita

Apsolutna greška nastala prilikom izračunavanja brze Furijeove transformacije opisanim metodama prikazana je tabelarno na isti način kao i u poglavlju 3.3.

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	0	0	4	24	45	55
new bf2	0	0	0	0	0	0	7	29	46	46
new bf3	0	0	0	0	0	0	15	40	33	40

Tabela 4.1 Distribucija netačnih bita za signal 32 tona i N=64

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	0	0	0	14	83	159
new bf2	0	0	0	0	0	0	0	46	69	141
new bf3	0	0	0	0	0	0	4	51	76	125

Tabela 4.2 Distribucija netačnih bita za signal 32 tona i N=128

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	1	31	160	135	83	102
new bf2	0	0	0	0	5	67	174	110	94	62
new bf3	0	0	0	0	3	81	159	120	67	82

Tabela 4.3 Distribucija netačnih bita za signal 32 tona i N=256

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	1	11	143	314	244	142	169
new bf2	0	0	0	1	20	177	316	236	134	140
new bf3	0	0	0	0	22	226	308	217	138	113

Tabela 4.4 Distribucija netačnih bita za signal 32 tona i N=512

	10	11	12	13	14	15	16	17	18	19
New bf1	0	0	0	0	9	146	541	602	381	369
New bf2	0	0	0	2	44	274	603	529	292	304
New bf3	0	0	0	0	4	243	635	536	301	329

Tabela 4.5 Distribucija netačnih bita za signal 32 tona i N=1024

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	0	0	0	19	27	82
new bf2	0	0	0	0	0	0	2	29	33	64
new bf3	0	0	0	0	0	0	15	49	27	37

Tabela 4.6 Distribucija netačnih bita za signal 64 tona i N=64

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	0	0	0	17	89	150
new bf2	0	0	0	0	0	0	0	38	95	123
new bf3	0	0	0	0	0	0	0	47	87	122

Tabela 4.7 Distribucija netačnih bita za signal 64 tona i N=128

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	2	38	140	153	80	99
new bf2	0	0	0	0	3	38	143	152	80	96
new bf3	0	0	0	0	4	86	151	135	66	70

Tabela 4.8 Distribucija netačnih bita za signal 64 tona i N=256

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	1	25	138	313	247	154	146
new bf2	0	0	0	1	25	161	287	260	136	154
new bf3	0	0	0	1	30	208	339	195	114	137

Tabela 4.9 Distribucija netačnih bita za signal 64 tona i N=512

	10	11	12	13	14	15	16	17	18	19
new bf1	0	0	0	0	18	228	634	520	332	316
new bf2	0	0	0	3	63	333	610	487	269	283
new bf3	0	0	0	0	7	278	670	494	297	302

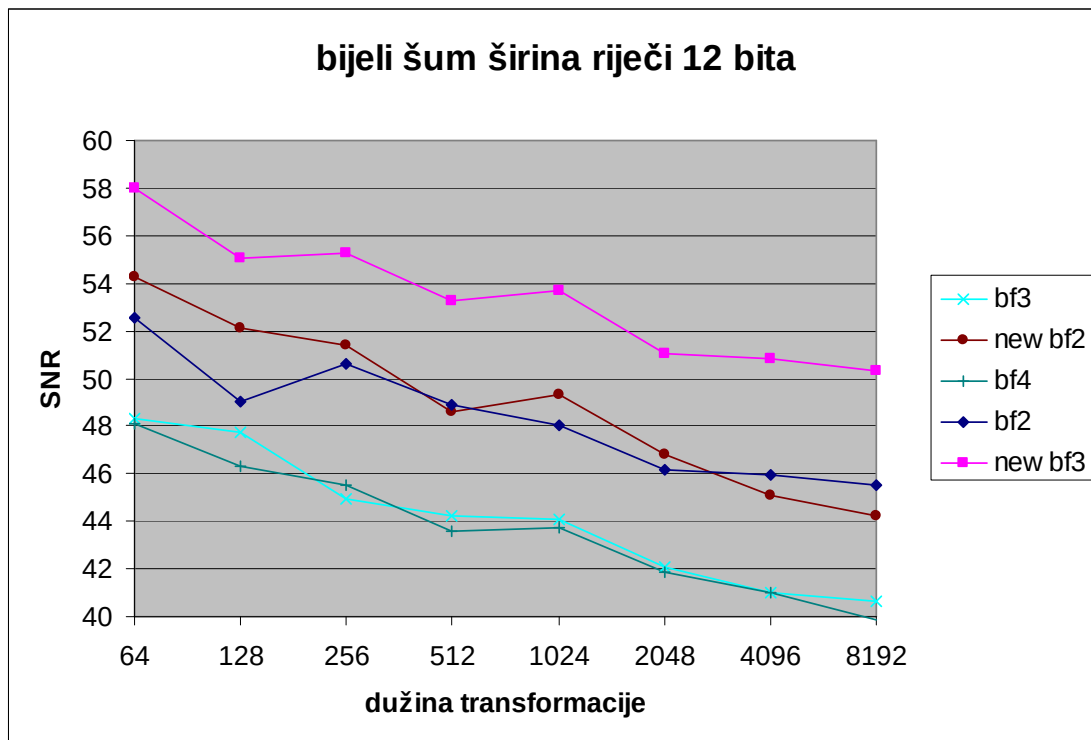
Tabela 4.10 Distribucija netačnih bita za signal 64 tona i N=1024

Analiza grešaka nastalih prilikom izračunavanja brze Furijeove transformacije iskazanih u obliku signal-šum kvantizacije i u obliku apsolutne greške za opisane metoda je pokazala da sve tri varijante predložene metoda daju približno jednake rezultate sa malom prednošću metode “new bf1” kod koje se ne obavlja izbjegavanje prekoračenja kod kompleksnog množenja.

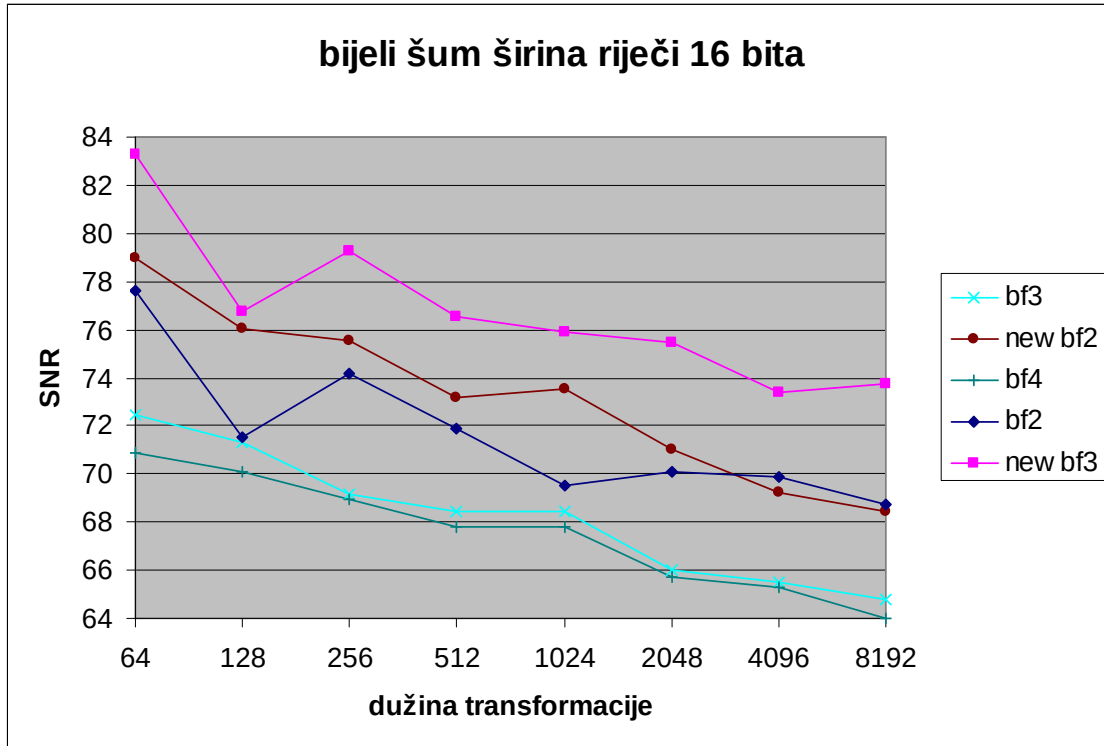
4.3 Poređenje sa drugim metodama

Zbog odsustva izbjegavanja prekoračenja kod kompleksnog množenja metode “bf1” i “new bf1” kao i metoda automatskog skaliranja zbog primjetno lošijih rezultata nisu među metodama koje će biti međusobno upoređene u ovom poglavlju. Izbjegavanje prekoračenja kod kompleksnog množenja u metodi “new bf1” je izvedeno u realizacije u programabilnoj sekvencijalnoj mreži ali ga nije moguće realizovati na digitalnom signalnom procesoru.

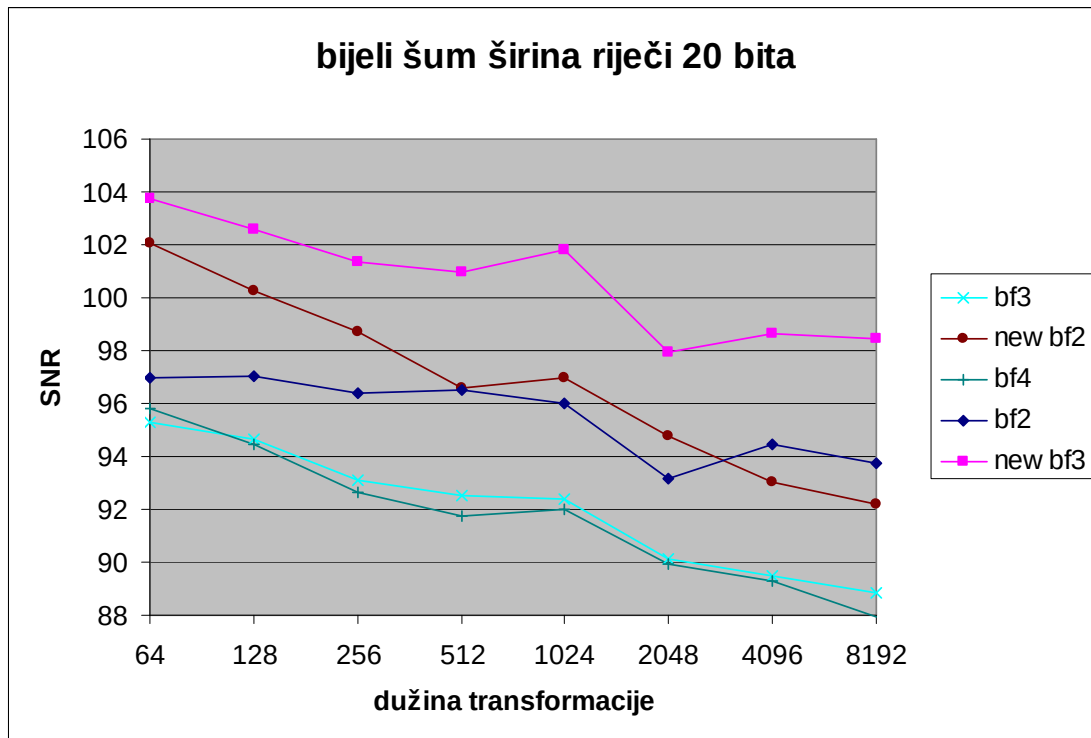
Metode bf2, bf3 i bf4 iz prve grupe metoda i metode new bf2 i new bf3 iz grupe predloženih metoda će biti upoređene u ovom poglavlju. Metode bf2 i new bf3 koriste istovjetan metod za izbjegavanje prekoračenja kod kompleksnog množenja pa se uglavnom i slično ponašaju. Primjetno je lošije ponašanje ovih metoda kod izračunavanja brze Furijeove transformacije dužina 64 i 128 zbog korištenja faktora skaliranja koji je mnogo strožiji od neophodnog (~ 0.707 odnosno ~ 0.758 dok je dovoljan faktor skaliranja ~ 0.828).



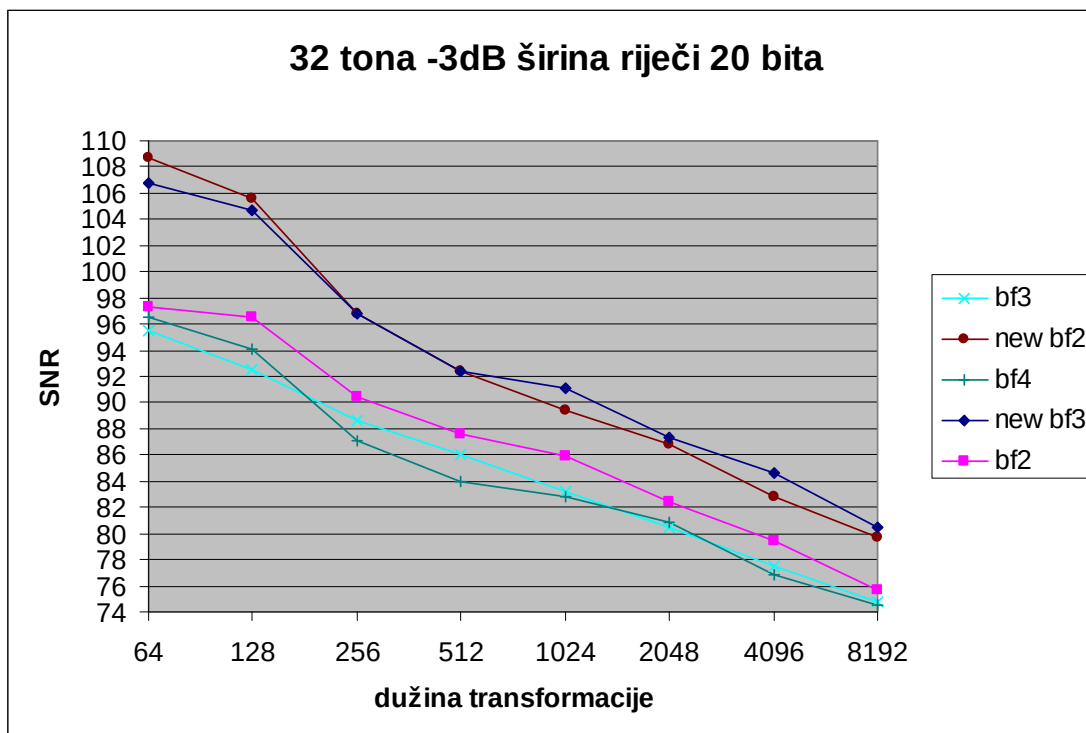
Slika 4.6 Odnos signal-šum za signal bijelog šuma i širinu riječi od 12 bita



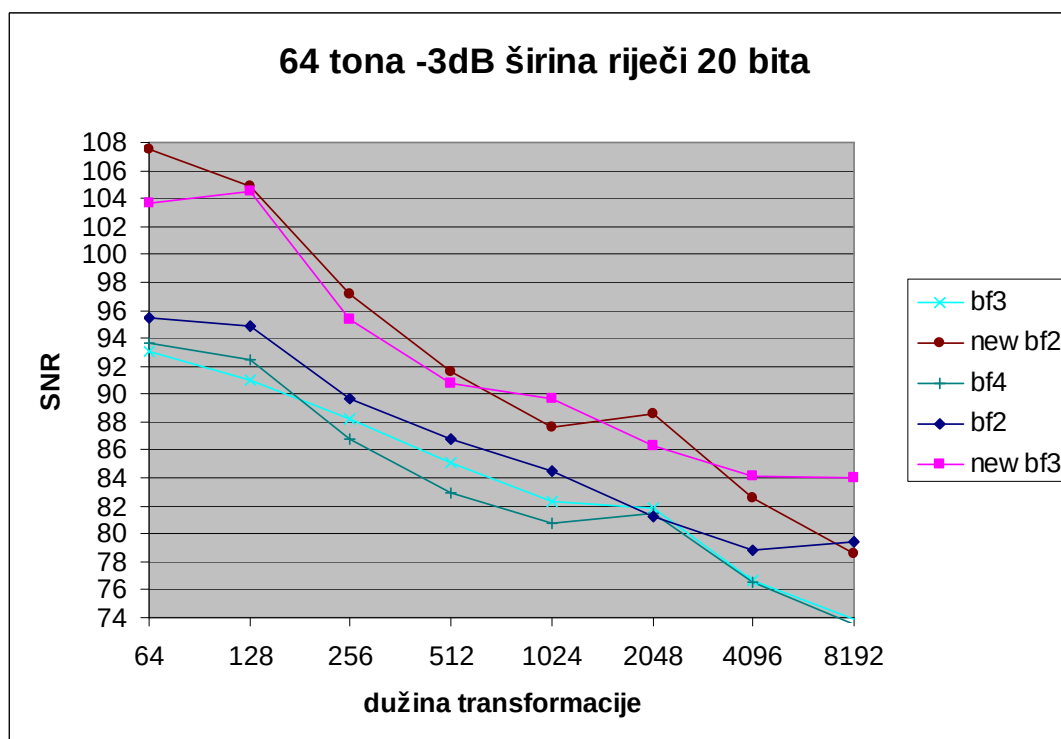
Slika 4.7 Odnos signal-šum za signal bijelog šuma i širinu riječi od 16 bita



Slika 4.8 Odnos signal-šum za signal bijelog šuma i širinu riječi od 20 bita



Slika 4.9 Odnos signal-šum za multitonski signal (32 tona) i širinu riječi od 20 bita



Slika 4.10 Odnos signal-šum za multitonski signal (64 tona) i širinu riječi od 20 bita

Apsolutna greška nastala prilikom izračunavanja brze Furijeove transformacije opisanim metodama prikazana je tabelarno na isti način kao i u poglavlju 3.3.

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	0	2	39	37	27	10	13
bf3	0	0	0	0	10	40	31	24	14	9
bf4	0	0	0	0	8	30	39	26	13	12
new bf2	0	0	0	0	0	0	7	29	46	46
new bf3	0	0	0	0	0	0	15	40	33	40

Tabela 4.11 Distribucija netačnih bita za signal 32 tona i N=64

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	0	0	14	73	77	46	46
bf3	0	0	0	0	5	49	85	50	35	32
bf4	0	0	0	0	0	39	79	51	39	48
new bf2	0	0	0	0	0	0	0	46	69	141
new bf3	0	0	0	0	0	0	4	51	76	125

Tabela 4.12 Distribucija netačnih bita za signal 32 tona i N=128

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	5	87	173	105	71	34	37
bf3	0	0	1	18	109	153	117	62	26	26
bf4	0	0	0	32	152	134	90	51	29	24
new bf2	0	0	0	0	5	67	174	110	94	62
new bf3	0	0	0	0	3	81	159	120	67	82

Tabela 4.13 Distribucija netačnih bita za signal 32 tona i N=256

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	7	161	333	230	162	61	70
bf3	0	0	2	23	200	312	226	143	61	57
bf4	0	0	3	81	247	289	192	100	66	46
new bf2	0	0	0	1	20	177	316	236	134	140
new bf3	0	0	0	0	22	226	308	217	138	113

Tabela 4.14 Distribucija netačnih bita za signal 32 tona i N=512

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	2	168	614	570	337	183	174
bf3	0	0	1	38	339	636	456	300	144	134
bf4	0	0	1	40	380	662	457	250	135	123
new bf2	0	0	0	2	44	274	603	529	292	304
new bf3	0	0	0	0	4	243	635	536	301	329

Tabela 4.15 Distribucija netačnih bita za signal 32 tona i N=1024

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	0	2	28	44	23	17	14
bf3	0	0	0	0	11	35	39	24	11	8
bf4	0	0	0	0	7	34	33	29	11	14
new bf2	0	0	0	0	0	0	2	29	33	64
new bf3	0	0	0	0	0	0	15	49	27	37

Tabela 4.16 Distribucija netačnih bita za signal 64 tona i N=64

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	0	0	14	78	72	39	53
bf3	0	0	0	0	9	42	89	56	32	28
bf4	0	0	0	0	0	48	79	67	27	35
new bf2	0	0	0	0	0	0	0	38	95	123
new bf3	0	0	0	0	0	0	0	47	87	122

Tabela 4.17 Distribucija netačnih bita za signal 64 tona i N=128

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	6	70	163	112	85	36	40
bf3	0	0	1	11	91	167	114	70	32	26
bf4	0	0	2	19	126	167	96	49	27	26
new bf2	0	0	0	0	3	38	143	152	80	96
new bf3	0	0	0	0	4	86	151	135	66	70

Tabela 4.18 Distribucija netačnih bita za signal 64 tona i N=256

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	8	140	342	264	143	59	68
bf3	0	0	2	19	202	314	232	134	65	56
bf4	0	0	2	65	284	294	185	99	49	46
new bf2	0	0	0	1	25	161	287	260	136	154
new bf3	0	0	0	1	30	208	339	195	114	137

Tabela 4.19 Distribucija netačnih bita za signal 64 tona i N=512

	10	11	12	13	14	15	16	17	18	19
bf2	0	0	0	8	179	657	550	312	172	170
bf3	0	0	2	29	342	656	472	281	139	127
bf4	0	0	4	79	447	653	432	218	107	108
new bf2	0	0	0	3	63	333	610	487	269	283
new bf3	0	0	0	0	7	278	670	494	297	302

Tabela 4.20 Distribucija netačnih bita za signal 64 tona i N=1024

Analiza grešaka nastalih prilikom izračunavanja brze Furijeove transformacije iskazanih u obliku signal-šum kvantizacije i u obliku apsolutne greške za opisane metoda je pokazala da obe varijante predložene metode (“new bf2” i “new bf3”) daju primjetno bolje rezultate od metoda prikazanih u poglavljima 3.1.2.2, 3.1.2.3 i 3.1.2.4.

Metode “bf2” i “new bf3” su podjednako računski složene i na većini arhitektura su računski složenije od preostalih metoda. Metoda “new bf3” je pokazala primjetno bolje rezultate od metode “bf2”. Razlika se kreće u opsegu od 4 do 9 dB ako se posmatra odnos signal-šum kvantizacije, ako se posmatra apsolutna greška tada razlika ide i do dva bita tačnosti.

Preostale metode (“bf3”, “bf4” i “new bf2”) posjeduju približnu računsku složenost i primjetno je bolje rezultate pokazala metoda “new bf2” (od 5 do 14 dB, odnosno i do dva bita (čak i tri bita u odnosu na metodu “bf3”)).

5. DSP primjena nove metode

5.1 Preporuke i mogući problemi u zavisnosti od raspoloživih resursa

U literaturi se izračunavanje transformacija uključujući i brzu Furijeovu transformaciju uglavnom obrađuje sa stanovišta brzine izračunavanja a na tačnost izračunavanja se ne obraća mnogo pažnje. U dijelu radova u kojima se analizira tačnost izračunavanja brze transformacije ne analiziraju se detalji implementacije, a često ni sama implementacija. U ovom poglavlju su izložene neke praktične preporuke za poboljšanje tačnosti prilikom izračunavanja brze Furijeove transformacije. Dio preporuka se odnosi na radix-2 FFT a većina je primjenjiva uopšte (sa manjim i li većim modifikacijama)

Čest je slučaj da je moduo maksimalne vrijednosti ulaznog niza mnogo manji od jedan, tj., dinamički opseg (broj značajnih bita) elemenata ulaznog niza je mnogo manji od dinamičkog opsega arhitekture na kojoj se algoritam izvršava. S obzirom da je analiza posljedica kvantizacije u FFT algoritmu pokazala da je šum kvantizacije nastao u prethodnim stepenima FFT algoritma umanjen skaliranjem u narednim stepenima, množenjem takve ulazne sekvence odgovarajućim brojem može dovesti do povećanja odnosa signal-šum. Idealno bi bilo koristiti recipročnu vrijednost maksimuma realnih i imaginarnih komponenti ($1/\max\{\max[|\operatorname{Re}[x(n)]|], \max[|\operatorname{Im}[x(n)]|\}\}$). Praktičnije je koristiti mnogo jednostavniji metod u kom se ulazni niz množi takvim brojem da se apsolutna vrijednost maksimuma realnih i imaginarnih komponenti dovode u opseg (0.5,1) tako da vrijedi

$$0.5 \leq \max\{\max[|\operatorname{Re}[x(n)]|], \max[|\operatorname{Im}[x(n)]|\}\} \leq 1 \quad (62)$$

gdje se množenje može izvesti kao aritmetičko pomjeranje ulijevo a određivanje broja potrebnih pomjeraja se svodi na određivanje pozicije najznačajnijeg bita (operacija podržana u mnogim fizičkim arhitekturama). Očigledno je da je konačan rezultat izračunavanja diskretne Furijeove transformacije preko FFT algoritma potrebno podijeliti istom vrijednosti. Ukoliko je prilikom izračunavanja došlo do skaliranja međurezultata sa dva konačan rezultat je potrebno podijeliti stepenom broja dva koji predstavlja razliku između broja pomjeraja u lijevo ulazne sekvence i efektivnog broja pomjeraja udesno međurezultata dobijenih prilikom izračunavanja FFT algoritma. Posljedica ovakvog načina skaliranja je smanjenje vrijednosti ukupne greške kvantizacije za vrijednost kojom su podijeljene izlazne vrijednosti transformacije.

Prva dva stepena radix-2 DIT FFT algoritma sadrže samo trivijalna množenja koja se mogu svesti na sabiranje i preuređivanje (zamjena realnih i imaginarnih komponenti) pa se na taj način ova dva stepena mogu izračunati potpuno tačno ukoliko ne dođe do pojave prekoračenja (u tom slučaju se mora obaviti zaokruživanje ili odsjecanje).

Prve dva stepena se mogu realizovati i korištenjem radix-4 baterflaja. Radix-4 baterflaj se sastoji od 4 sabiranja pa je maksimalna vrijednost međurezultata 4. U cilju izbjegavanja prekoračenja potrebno je ili skalirati izlazne vrijednosti za faktor 4 ili odrediti maksimalnu izlaznu vrijednost i njoj prilagoditi skaliranje. Primjena prve metode dovodi do značajne degradacije odnosa signal-šum a druga metoda zahtijeva dodatnu obradu čime se poništava ušteda postignuta korišćenjem radix-4 baterflaja. Primjenom radix-4 baterflaja u nekim (rijetkim) slučajevima može da se izbjegne skaliranje nakon prvog stepena ako je porast vrijednosti u prvom stepenu kompenzovan smanjenjem u drugom. Na mnogim arhitekturama je zbog toga pogodnije umjesto jednog stepena sa radix-4 baterflajima koristiti dva stepena sa radix-2 baterflajima.

U prethodnom tekstu se podrazumjevalo da se svaka kvantizacija obavlja zaokruživanjem. Operacija zaokruživanja obično nije podržana arhitekturom (automatsko zaokruživanje prilikom upisa u memoriju) nego se ona obavlja programski. Najjednostavniji način za izvođenje zaokruživanja je dodavanje jedinice na poziciji prvog bita koji se odbacuje i zatim upisa vrijednosti u memoriju praćenim odsijecanjem. Ovakva izvedba zaokruživanja zahtijeva jednu dodatno sabiranje za svaku vrijednost koja se zaokružuje. Dodatno je potrebno čuvati vrijednost koja se dodaje (jedinica na poziciji prvog bita koji se odbacuje) u nekom akumulatoru čime se smanjuje broj raspoloživih akumulatora.

Iako daje bolji rezultat nego prosto odsijecanje ovakva izvedba zaokruživanja nije dobra ukoliko se odbacuje samo jedan bit jer je u tom slučaju srednja vrijednost greške zaokruživanja različita od nule (pojavljuje se greška kod zaokruživanja negativnih vrijednosti). Ukoliko se odbacuje više bita tada je srednja greška kvantizacije blizu nule što je prihvatljivo. Tačna izvedba zaokruživanja zahtijeva upotrebu više programskih instrukcija i obično se ne koristi. Ukoliko se primjenjuje metoda iz poglavlja 4 tada je potrebno izvršiti zaokruživanje obe vrijednosti koje se upisuju u memoriju (skalirane i neskaliirane). Zaokruživanje je naročito važno u početnim stepenima brze Furijeove transformacije zbog kumulativne prirode greške kvantizacije. Izračunavanje posljednjih prolaza može da se obavi i korištenjem odsijecanja zbog manjeg uticaja greške kvantizacije u posljednjim prolazima na ukupnu grešku ukoliko je vrijeme izvršavanja kritično. Na taj način se štedi procesorsko vrijeme ali se i smanjuje odnos signal-šum (potrebno je naći kompromis).

Ukoliko se ne koristi zaokruživanje u ranim fazama izračunavanje FFT algoritma akumulirana greška kvantizacije može da poprими vrijednost koja je uporediva sa vrijednošću signala jer se ona obično koncentriše u pojedinim tačkama (nulta pozicija i pozicija $N/2$ su posebno kritične). Ova pojava ponekad ne utiče na vrijednost odnosa signal-šum ali može da ima neugodne posljedice na kvalitet reprodukovanog signala.

5.2 Primer konkretne primene – AC3 dekodera

5.2.1 Opis problema

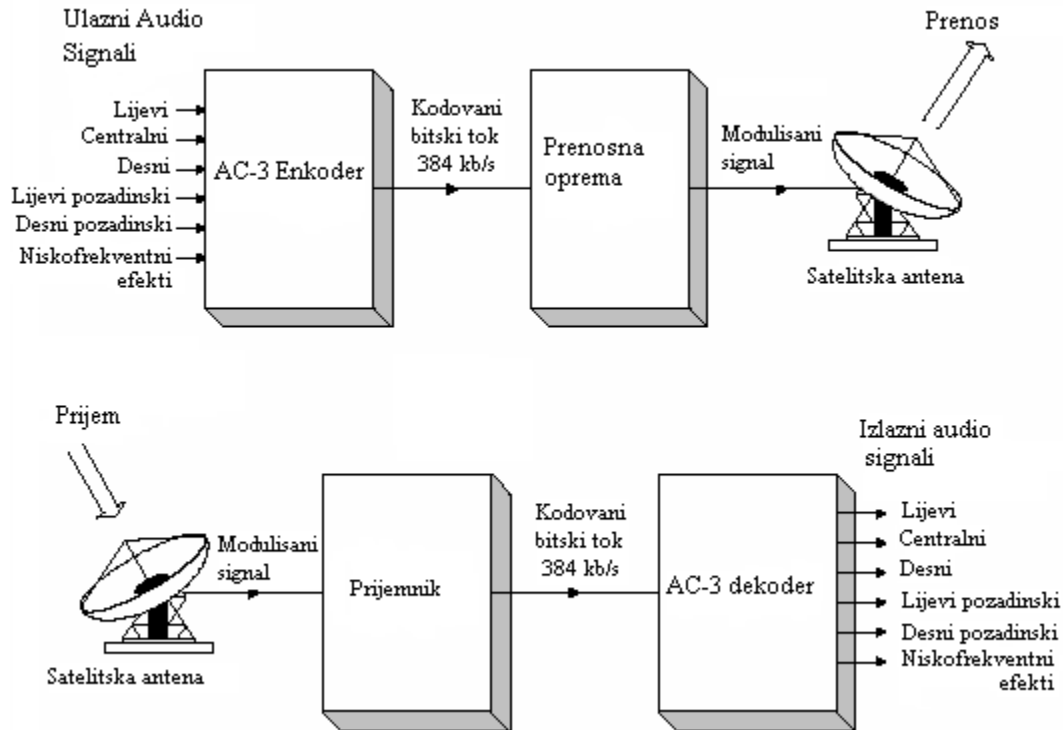
Moderni algoritmi za kompresiju audio signala, kao što su MP3, AAC, AC3, nazivaju se i “algoritmi sa gubitkom” ili perceptualni algoritmi za kompresiju. To znači da nakon kompresije i dekompresije izlaz nije potpuno ista reprodukcija originala. Algoritam za kompresiju uklanja dijelove signala za koje je utvrdio da se ne mogu otkriti slušanjem. Algoritam za kompresiju takođe i kvantizuje signal sto unosi šum na brižljivo kontrolisan način – šum koji slušalac neće čuti jer će biti maskiran drugim zvukovima. Uklanjanje dijelova signala i kvantizacija otežavaju testiranje, jer će tradicionalne mjere za utvrđivanje kvaliteta reprodukovano zvuka kao što je odnos signal šum pokazati značajnu degradaciju signala a slušalac vjerovatno neće čuti te razlike. Slično tome, kvantitativno zanemarljive razlike u signalu kao što je jedan izlazni odbirak sa pogrešnom vrijednošću mogu biti veoma primjetne slušaocu.

Zbog pomenutih poteškoća pri mjerenju audio kvaliteta perceptualnih algoritama dekodera se uobičajeno testiraju sa posebnim test vektorima i izlaz se poredi sa izlazom referentnog dekodera. Test vektori i referentni dekodera se obično mogu dobiti od tvorca algoritma za kompresiju (Dolbi laboratorije za AC3) ili standardizacije ustanove (ISO/IEC za MP3/AAC).

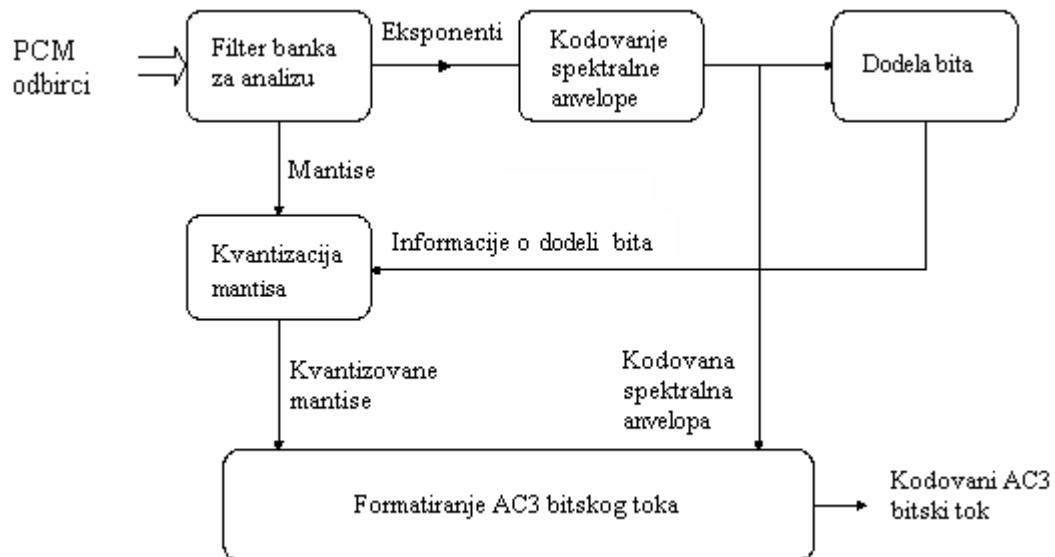
Referentni izlaz je obično generisan dekodrom koji koristi aritmetiku pokretnog zareza, dok tipične DSP arhitekture koriste aritmetiku nepokretnog zareza. U idealnom slučaju izlaz dekodera koji koristi aritmetiku nepokretnog zareza odgovara izlazu referentnog dekodera koji koristi aritmetiku pokretnog zareza ali to zavisi od kvaliteta implementacije dekodera koji koristi aritmetiku nepokretnog zareza. Kreiranje kvalitetne implementacije dekodera koji koristi aritmetiku nepokretnog zareza zahtijeva potpuno razumjevanje potencijalnog dinamičkog opsega algoritma i prostiranja greške aritmetike nepokretnog zareza.

5.2.1.1 AC3 algoritam

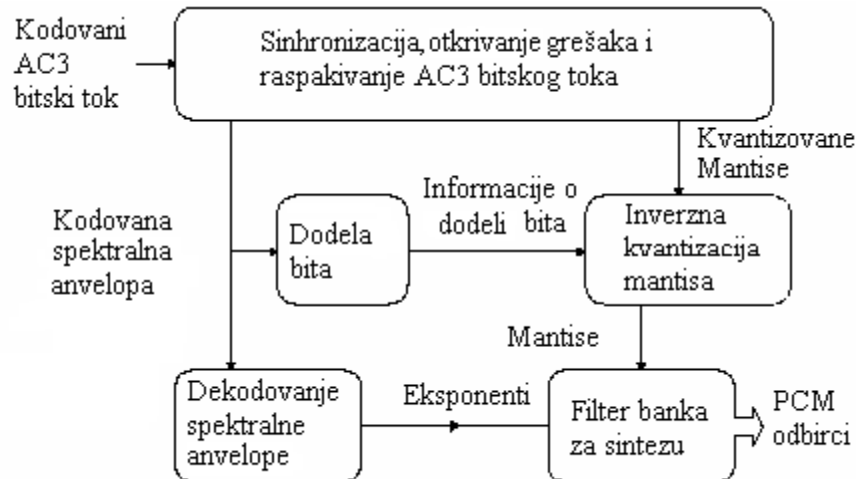
Dolby AC3 je fleksibilan algoritam za kompresiju zvuka koji omogućava efikasnu kompresiju više kanalnog zvuka. AC3 algoritmom je moguće komprimovati se od jednog (monofonski signal) do 5.1 kanala (lijevi, centralni desni, lijevi pozadinski, desni pozadinski i kanal sa niskim frekvencijama). Kanal sa oznakom “.1” je namijenjen niskofrekventnim efektima i ograničen je na frekvencije niže od 120 Hz. Moguće je enkodovati audio signal odabiran na frekvencijama od 32, 44.1 i 48 kHz na bitskim brzinama od 32-640 kbita u sekundi. Algoritam omogućuje visoko kvalitetan zvuk pri bitskim brzinama od 64 kbita po kanalu za višekanalni zvuk. Primjer primjene AC3 algoritma prikazan je na slici 5.1. Blok dijagram AC3 enkodera i dekodera su dati na slikama 5.2. i 5.3.



Slika 5.1 Primjer primjene AC3 algoritma



Slika 5.2 Blok dijagram AC3 enkodera



Slika 5.3 Blok dijagram AC3 dekodera

5.2.1.1.1 Filter banka

Izbor filter banke za analizu-sintezu za audio koder je kompromis između frekvente rezolucije, vremenske rezolucije i potrebnih resursa (potrebna memorija i broj potrebnih računskih operacija). Preciznija frekventna rezolucija je pogodna za vremenski stacionarne signale dok je za vremenski brzo promjenljive signale potrebna bolja vremenska rezolucija. Nije moguće istovremeno postići veliku vremensku i frekventnu rezoluciju.

U AC3 algoritmu se koristi filter banka zasnovana na modifikovanoj diskretnoj kosinusnoj transformaciji. Preklopljeni blokovi od 512 prozoriranih odbiraka se transformišu u blokove od 256 tačaka u frekventnom domenu. Audio signal odabiran na 48 kHz se dijeli u preklopljene blokove dužine 10.66 ms i pri tome se transformacije obavljaju svakih 5.55 ms. Skupina od 6 blokova se kombinuje u jedan AC3 okvir. Blokovi od 512 odbiraka se formiraju od 256 novih odbiraka i 256 odbiraka iz prethodnog bloka. Preklapanje blokova doprinosi izbjegavanju primjetnih izobličenja koji nastaju usljed preklapanja spektra.

U cilju smanjenja prednjeg eha koriste se dvije dužine prozorske funkcije. Prozorska funkcija dužine 512 odbiraka (omogućuje frekventnu rezoluciju od 93.75 Hz na 48 kHz) se koristi za vrijeme vremenski stacionarnih signala, dok se za vrijeme vremenski brzo promjenljivih signala koristi par prozorskih funkcija dužine 256 odbiraka (omogućujući vremensku rezoluciju od 2.67ms na 48kHz). Prilikom otkrivanja vremenski brzo promjenljivog signala na bilo kojem od 5.1 kanala moguće je na tom kanalu koristiti kraću prozorsku funkciju potpuno nezavisno od drugih kanala.

Namjenski razvijena KBD prozorska funkcija se koristi za postizanje kompromisa između selektivnosti u okolini propusnog opsega i potiskivanja u

nepropusnom opsegu. Prozorska funkcija omogućuje perfektu rekonstrukciju nakon primjene operacije preklopi i saberi u dekodneru.

5.2.1.1.2 Spektralna anvelopa

Svaki pojedini transformacioni koeficijent se predstavlja u obliku eksponenta i mantise. Eksponent omogućuje veći dinamički opseg, dok se mantisa koduje sa ograničenom tačnošću što za posljedicu ima pojavu šuma kvantizacije. Skup eksponenata je način predstavljanja spektra signala i naziva se spektralna anvelopa. U AC3 algoritmu se koristi mehanizam kodiranja eksponenata u cilju smanjenja potrebne količine podataka. Prvo se eksponenti koji odgovaraju svakom od šest blokova koji čine jedan AC3 okvir porede u cilju identifikovanja razlika među blokovima. Ako je razlika mala jedinstven skup eksponenata se koristi za svih šest blokova. Ako postoji značajna razlika među eksponentima unutar okvira tada se jedinstven skup eksponenata formira za blokove u kojima razlike nisu značajne. Eksponenti koji odgovaraju susjednim frekvencijama se rijetko razlikuju za više od 2 (1 odgovara razlici u nivou od 6dB) zbog prirode frekventne karakteristike individualnih filtera u filter banci za analizu (slabljenje je približno 12 dB na frekvenciji susjednog filtera). Zahvaljujući tome eksponenti se koduju diferencijalno po susjednim frekvencijama. Prvi eksponent u bloku (odgovara jednosmjernoj komponenti) se postavlja na svoju originalnu vrijednost a ostali eksponenti predstavljaju razliku između tekućeg eksponenta i njemu prethodnog eksponenta. Vrijednosti razlika su ograničene na +2,+1,0, -1 i -2.

U zavisnosti od sadržaja audio signala bira se jedan od tri mehanizma za kodiranje. Kad je potrebno postići finu spektralnu rezoluciju za vrijeme relativno stacionarnih signala i kad je spektralna anvelopa relativno nepromjenljiva tri razlike se koduju u jednu 7 bitnu riječ. Ovom metodom, koja se naziva D15 kodovanje, se svaki eksponent koduje sa 2.33 bita. Kada spektar nije stabilan potrebno je češće slati procjene spektra. U cilju izbjegavanja pretjeranog porasta potrebne količine podataka koriste se dva dodatna moda (srednji i niski). Kod moda sa srednjom rezolucijom, nazvanog D25, razlika eksponenata se prenosi za svaku drugu frekvenciju što predstavlja kodovanje para eksponenata sa 2.33 bita, tj., kodovanje svakog eksponenta sa 1.16 bita. Ovaj mod se koristi kad je spektar relativno stabilan na dva do tri bloka i i onda se znatno promjeni. D25 metoda ne omogućuje spektralnoj anvelopi da tačno prati doline u spektru signala kod veoma tonalnog spektra ali je tjere da prati spektralne vrhove. Posljednji mod, nazvan D45, omogućuje kodovanje razlike eksponenata za svaki četvrti spektralni koeficijent čime se prepolovljava bitska brzina u odnosu na D25 metodu i uglavnom se koristi za vrijeme brzo promjenljivih signala u okviru jednog audio bloka.

Brzo promjenljivi signali uglavnom ne zahtijevaju finu spektralnu rezoluciju jer po svojoj prirodi imaju širok spektar. Primjenom ovih metoda moguće je postići finu spektralnu rezoluciju za vrijeme relativno stacionarnih signala i finu vremensku rezoluciju za vrijeme brzo promjenljivih signala. Efektivan stepen kodovanja eksponenata je obično 0.39 bita po eksponentu (što odgovara 0.39 bita po odbirku). Za većinu signala eksponenti kodovani D15 metodom se šalju za vrijeme prvog bloka i

koriste i u narednim blokovima. Za vrijeme brzo promjenljivih signala eksponenti se šalju mnogo češće.

5.2.1.1.3 Dodela bita

AC3 algoritam koristi hibridnu tehniku za dodelu bita koja je adaptivna unaprijed i unazad. Osnova je unazad adaptivni algoritam za dodelu bita koja se primjenjuje i u enkoderu i u dekoderu. Osnovni algoritam je relativno jednostavan i zasniva se na psihoakustičnom modelu. Upravljan spektralnom anvelopom koja je ujedno sadržana u enkodovanim podacima. Filter banka za analizu, filter banka za sintezu i spektralna anvelopa posjeduju jednaku vremensku i spektralnu rezoluciju.

Adaptacija unazad omogućava podešavanje parametara psihoakustičnog modela i izmjenu dodele bita (engleski “delta bit allocation”). Osnovni algoritam za dodelu bita izvodi pretpostavke o maskirajućim osobinama signala i čovjekom čujnom aparatu. Određeni parametri modela se eksplicitno prenose u okviru bitskog toka. Enkoder može da obavi dodelu bita korištenjem psihoakustičkog modela bilo koje kompleksnosti i uporedi tako dobijene rezultate sa rezultatima dobijenim osnovnim algoritmom. Ako je moguće izvesti bolju dodelu bita promjenom jednog ili više parametara modela korištenih u osnovnom algoritmu, enkoder može da obavi date izmjene i prenese dekoderu informacije o promjenama prenoseći eksplicitno nove vrijednosti parametara. U slučaju da nije moguće postići približno idealnu dodelu bita prostom izmjenom parametara enkoder može eksplicitno da pošalje podatke o dodeli bita. Sintaksa AC3 algoritma omogućava enkoderu da pošalje informacije za izmjenu dodele bita koje omogućavaju smanjenje ili povećanje broja dodjeljenih bita u uskim frekventim opsezima. Konačne dodele bita korištene u enkoderu i dekoderu moraju biti identične i sastoje se od rezultata dobijenih primjenom osnovne metode modifikovanih promjenom parametara algoritma i informacija o izmjeni dodele bita dobijenih od enkodera.

Tačnost predstavljanja mantisa zavisi od širine riječi izvora ulaznog signala. Uglavnom se mantise predstavljaju sa 16-20 bita, a moguće je predstaviti mantise i sa do 24 bita. Normalizovane mantise se kvantizuju na od 0 do 16 bita. Broj bita pridružen svakoj mantisi je određen osnovnim metodom za dodelu bita koji je identičan u enkoderu i u dekoderu. Osnovni metod za dodelu bita AC3 algoritma se naziva adaptivnim unazad zato što se informacije predstavljene spektralnom anvelopom koriste za izračunavanje konačne dodele bita.

5.2.1.1.4 Sprezanje kanala

Iako tehnike za kompresiju zvuka korištene u AC3 algoritmu omogućuju prilično dobru kompresiju prilikom rada enkodera pri veoma niskim bitskim brzinama postoje situacije u kojima dekoder ne bi uspio da enkoduje signal dozvoljenim brojem bita. U tom slučaju se koristi tehnika pod imenom sprezanje spektra (engl. “coupling”). Tehnika sprezanja spektra koristi način na koji ljudsko uho određuje određuje smjer izvora

visokofrekventnih sinala u cilju smanjenja neophodne količine podataka. Unutar uskog dijela spektra ljudsko uho lokalizuje visoke frekvencije (2 kHz) više na osnovu spektralne anvelope nego na osnovu potpunog talasnog oblika. Ljudsko uho nije u stanju da utvrdi smjer izvora dva u spektralnom domenu bliska visoko frekventna signala. Korištenjem ove tehnike kombinuje se visokofrekventni sadržaj oba signala i dekodernu se šalju podaci o anvelopama individualnih kanala zajedno sa kanalom koji sadrži kombinovani spektar. Frekvencija na kojoj počinje sprežanje kanala se naziva frekvencija sprežanja. Spregnuti kanal se koduje na isti način kao i individualni kanali preko svoje spektralne anvelope i skupa kvantizovanih mantisa. U enkoderu se određuje snaga signala ulaznih kanala u uskim opsezima spektra kao i snaga u istovjetnim opsezima spektra spregnutog kanala. Enkoder računa koordinate sprežanja za svaki kanal koje predstavljaju odnos snage originalnog signala i snage signala u spregnutom kanalu u odgovarajućim opsezima spektra. Kanali koji se sprežu se prenose do frekvencije sprežanja a zatim se prenose samo koordinate sprežanja. Pojedine koordinate sprežanja se množe sa koeficijentima spregnutog kanala u dekodernu da bi se rekonstruisao visokofrekventni sadržaj spregnutih kanala.

5.2.1.1.5 Kontrola nivoa glasnosti

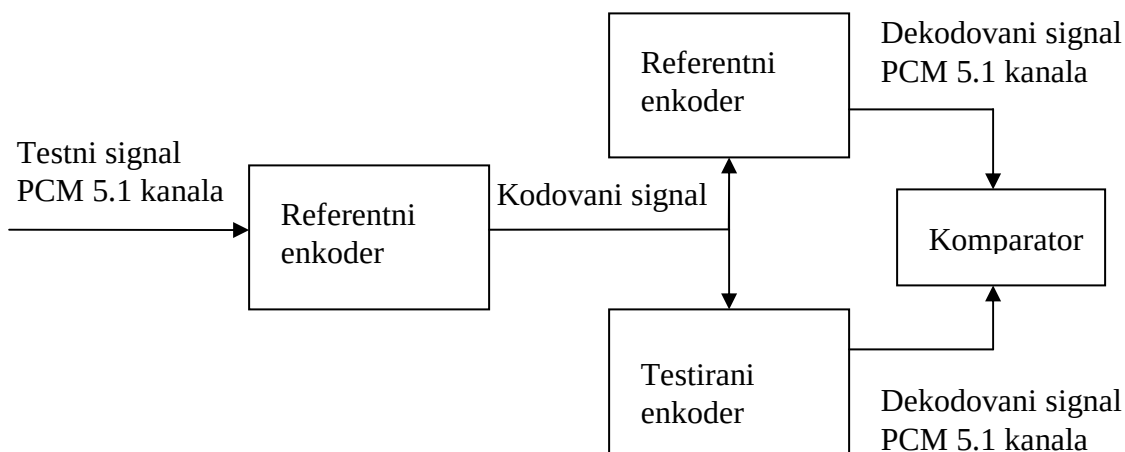
Prilikom prenosa TV signala često se smanjuje dinamički opseg audio signala. Ponekad zabavni dio programa posjeduje prirodniji dinamički opseg sa nivoom manjim od maksimalnog nivoa što nije slučaj sa reklamnim porukama (prilično su bučne). Rezultat toga je prilična razlika u nivou audio signala pojedinog TV kanala, kao i među različitim TV kanalima. AC3 algoritam posjeduje mogućnost kontrole dinamičkog opsega kojom se omogućuje promjena dinamičkog opsega reprodukovanog signala za svaki blok. Primjenom kontrole dinamičkog opsega moguće je eliminisati takvu razliku u nivou signala i omogućiti TV distributerima kompresiju dinamičkog opsega za većinu gledalaca istovremeno omogućujući gledaocima (slušaocima) koji to žele da uživaju u punom dinamičkom opsegu audio signala. Ravnomjernost glasnosti zvuka se postiže određivanjem subjektivnog nivoa zvuka kod normalnog dijaloga i kodiranjem tog nivoa u tok podataka. Tu vrijednost AC3 dekodeer može da kombinuje sa nivoom glasnosti sistema za reprodukciju koji je odredio korisnik. Kad slušaoc podesi nivo glasnosti tada će nivo reprodukovanog normalnog dijaloga (bez šaptanja ili vikanja) biti postavljen na željeni subjektivni nivo zvuka. Za dijelove programa sa nivoom zvuka iznad nivoa dijaloga kontrolne informacije dinamičkog opsega će indicirati smanjenje nivoa. Za zvuk sa nivoom ispod nivoa dijaloga kontrolne informacije će indicirati povećanje nivoa.

Kompresijom dinamičkog opsega utišavaju se glasni sigali približavajući njihov nivo nivou dijaloga a tihi signali postaju primjetniji (lakše se čuju). Ukoliko se u dekodernu ne koriste kontrolne informacije dinamičkog opsega onda se reprodukuje zvuk punog (originalnog) dinamičkog opsega.

5.2.1.2 Način provjere AC3 dekodera

Verifikacija konkretne implementacije dekodera nekog algoritma za kompresiju audio signala predstavlja skup metoda kojima se utvrđuje u kojoj mjeri pomenuta implementacija zadovoljava kriterijume koje je zadao proizvođač (tvorac) algoritma.

Metode se mogu podijeliti na subjektivne i objektivne metode. Subjektivne metode se zasnivaju na slušanju testnih sekvenci dekomprimovanih dekoderom koji se testira u cilju identifikovanja primjetnih izobličenja unesenih u procesu dekodovanja koja se ne mogu utvrditi objektivnim metodama. Objektivne metode se zasnivaju na poređenju tesnih sekvenci dekomprimovanih referentnim dekoderom i dekoderom koji se testira (slika 5.4). Objektivne metode su razvijene u cilju pojednostavljenja i ubrzavanja procesa verifikacije implementacija dekodera. Nakon što je tvorac algoritma verifikovao referentni dekoder (slično je i kod enkodera) verifikacije pojedinih realizacija dekodera se zasnivaju na poređenju sa referentnim dekoderom. Metode se nazivaju objektivnim jer se zasnivaju na poređenju mjerljivih karakteristika.



Slika 5.4 Blok dijagram objektivnog testiranja implementacije dekodera

Testiranje korištenjem objektivnih metoda obuhvata izvođenje funkcionalnih testova i testova za utvrđivanje performanse dekodera.

Prva grupa testova je namjenjena utvrđivanju da li je dekodер ispunjava osnovne uslove (korektnost implementacije) kao što su sinhronizacija, ispravno interpretiranje kontrolnih informacija, mogućnost korektnе reprodukcije spektra pri različitim nivoima signala, ispravan rad dekodera u različitim režimima rada, separacija kanala (naročito kod višekanalnih algoritama), mogućnost dekodovanja komprivanih audio signala pri različitim bitskim brzinama, mogućnost dekodovanja komprimovanih audio signala

odabiranih različitim frekvencijama odabiranja, mogućnost dekodovanja komprimiranih audio signala kreiranih sa budućim verzijama enkodera (poboljšane verzije enkodera u odnosu na enkodere koji se trenutno koriste) i otpornost na greške u ulaznom bitskom toku (dekoder mora brzo da oporavi i nastavi korektan rad, nivo izlaznog signala treba da se maksimalno utiša tako da to nije neprijatno za slušaoca).

Druga grupa testova je namjenjena utvrđivanju kvaliteta implementacije koji se prvenstveno ogleda u mogućnosti dekodovanja zahtijevnih sekvenci komprimiranih audio signala (zahtijevaju veliku procesnu moć za dekodovanje) i u aritmetičkoj tačnosti dekodera koja se ogleda u količini unesenog šuma, nivou harmonijskih izobličenja i dinamičkom opsegu.

5.2.2 Realizovano rješenje

Pored izmjena originalnog koda i algoritma AC3 dekodora izvedenih u cilju bržeg izvršavanja obavljen je su i izmjene u cilju smanjenja šuma unesenog korištenjem aritmetike nepokretnog zareza.

Prva izmjena je načinjenja na prelazu između korištenja cjelobrojne aritmetike i aritmetike nepokretnog zareza kod raspakivanja i normalizacije mantisa gdje je uvedeno zaokruživanje mantisa prije upisa u memoriju. Takođe se mantise upisuju normalizovane originalnim eksponentom (eksponenti odgovaraju broju dijeljenja sa dva i imaju negativnu ili nula vrijednost) kao i originalnim eksponentom uvećanim za jedan (maksimalna vrijednost normalizovane mantise može biti manja od polovine opsega). Ukoliko nije došlo do prekoračenja koristi se vrijednost mantisa sa većim brojem značajnih bita.

Nakon normalizacije mantisa, koja je dio originalnog algoritma, obavlja se normalizacija svih vrijednosti mantisa koje pripadaju jednom bloku na opseg između polovine maksimalne vrijednosti i maksimalne vrijednosti čime se mijenja i predstava vrijednosti iz format nepokretnog zareza u format blok nepokretnog zareza.

Prilikom izračunavanja inverzne transformacije koristi se dinamičko skaliranje. Inverzna transformacija se sastoji iz stepena za premodulaciju, sedam (šest ukoliko se koristi kraća prozorska funkcija zbog izbjegavanja prednjeg odjeka) stepeni radix-2 brze Furijeove transformacije, stepena za postmodulaciju i dijela za obavljanje prozoriranja i operacije preklopi i saberi. Zbog potrebe za brzim izvršavanjem algoritma dinamičko skaliranje se ne koristi u dijelu za prozoriranje i operaciju preklopi i saberi. Inverzna transformacija je grafički prikazana na slici 5.5.

Prvi dio radix-2 brze Furijeove transformacije koji je u referentnom kodu implementiran preko jednog stepena radix-4 baterflaja je podijeljen na dva trivijalna radix-2 baterflaja zbog jednostavnijeg izbjegavanja prekoračenja.

U preostalim stepenima je svaki ne-trivijalni radix-2 baterflaj podijeljen na dio u kojem se obavi kompleksno množenje i dio u kojem se obavlja kompleksno sabiranje u

cilju izbjegavanja prekoračenja kod kompleksnog množenja i kompleksnog sabiranja. Prije upisa u memoriju obavlja se zaokruživanje vrijednosti na 20 bita (širina memorijske riječi MAS procesora) dodavanjem jedinice na poziciji prvog bita koji se odbacuje (operacija sabiranja) i zatim odsijecanje prilikom upisa u memoriju.

U stepenima za premodulaciju se takođe obavlja zaokruživanje i izbjegavanja prekoračenja kod kompleksnog množenja metodom opisanom u poglavlju 4.1.2.1. Nakon stepena za postmodulaciju obavlja se reskaliranje vrijednosti na očekivani opseg (prije premodulacije je obavljeno skaliranje ulijevo, tokom transformacije je obavljen određen broj skaliranja udesno, ukupan broj skaliranja unutar transformacije treba da bude nula).

Prozoriranje i procedura preklopi i saberi se obavljaju istovremeno uz zaokruživanje rezultata na prethodno opisan način.

Transformacioni koeficijenti	DCT premodulacija	iFFT	DCT postmodulacija	Prozoriranje Preklopi i saberi	PCM izlaz
stepeni	1	2	5	1	1
aritmetika	Blok nepokretni zarez			Nepokretni zarez	

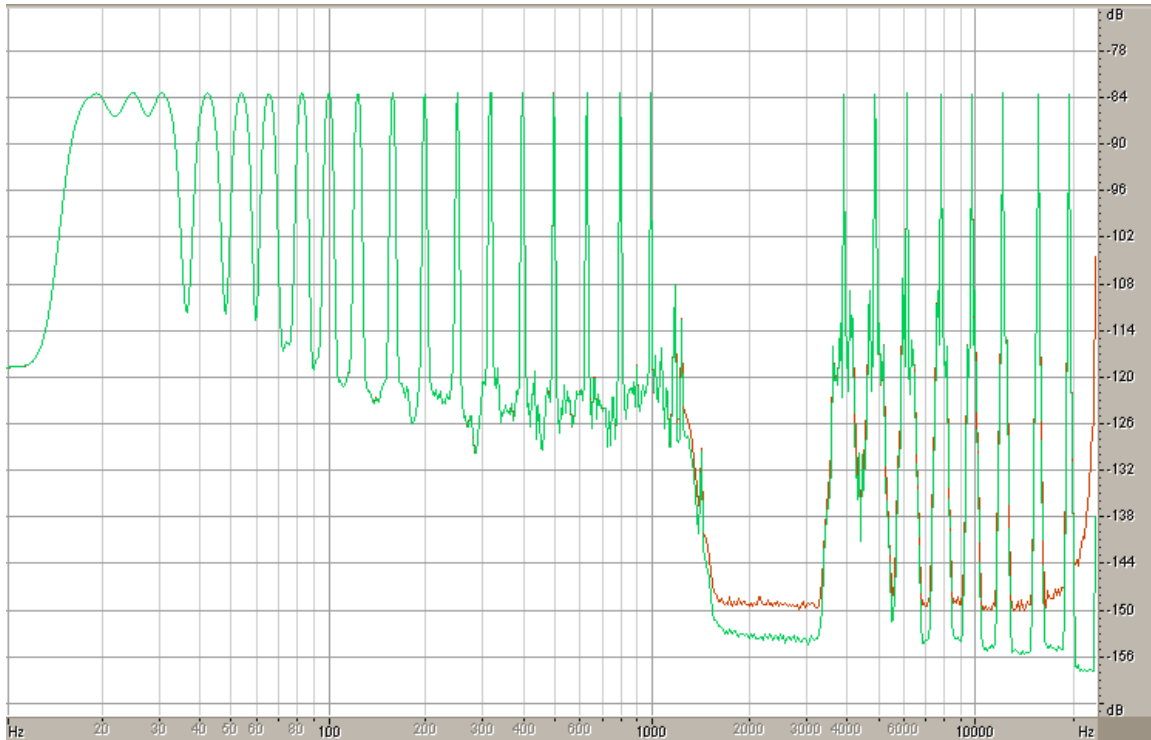
Slika 5.5 Grafički prikaz inverzne transformacije

5.2.3 Dobijeni rezultati

Prilikom ocjene kvaliteta realizacije AC3 dekodera bitan pokazatelj je količina unijetog šuma (uglavnom šum kvantizacije) u dijelovima spektra koji nisu maskirani tonalnim komponentama. Odnos signal-šum kvantizacije daje informaciju o ukupnom iznosu šuma. Kod signala pretežno tonalne prirode dio tog šuma će biti maskiran (u perceptualnom smislu) tonalnim komponentama i neće biti otkriven od strane slušaoca. Uneseni šum koji se nalazi u dijelovima spektra u kojima ne postoji signal (uglavnom) nije maskiran signalom i može biti otkriven od strane slušaoca. Iz tog razloga su važni testovi kojima se utvrđuje količina unijetog šuma u dijelovima spektra u kojima nema signala. Ukoliko postoji takav šum tada on treba da bude uniformno raspodjeljen (aditivan bijeli šum). Ako to nije slučaj, tada takav šum odgovara unošenju tonalnih komponenti u signal (tonalni šum).

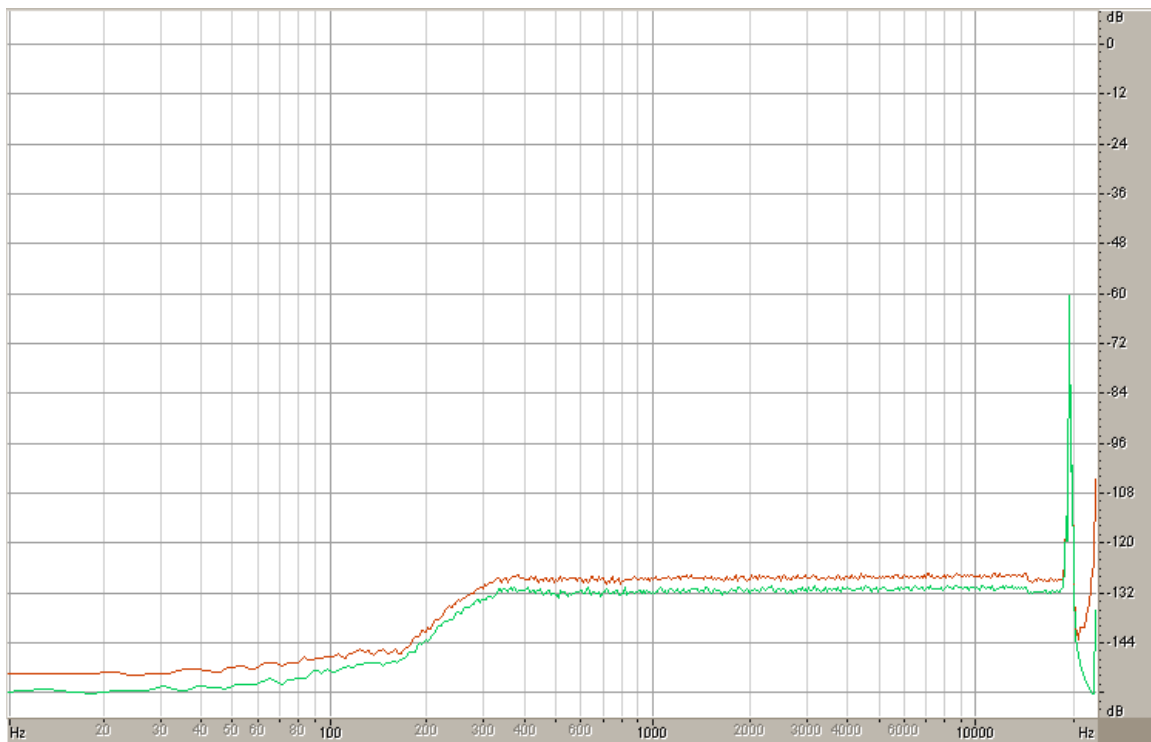
Prvi test takvog tipa je test kod kojeg se koristi multitonski signal sa podjednakim sinusnim komponentama na svakoj trećoj oktavnoj frekvenciji osim u opsegu od 1k-4kHz sa nivoom od -60dBFS. Cilj testa je utvrđivanje aritmetičke tačnosti dekodera pomoću nivoa šuma u dijelu spektra u kojem ne postoji signal (sinsune komponente). Na slici 5.6

prikazan je spektar multitonskog signala dobijenog testiranim dekoderom (aritmetika nepokretnog zarez) i referentnim dekoderom (aritmetika pokretnog zarez). Može se uočiti da se spektralne karakteristike poklapaju u dijelu spektra do 1kHz gdje je i veća koncentracija sinusnih komponenti a da u ostatku spektra postoje odstupanja. U posmatranoj oblasti (1.5-3 kHz) nivo šuma koji je unio testirani dekodera je za 5 dB viši od referentnog nivoa šuma. Unijeti šum je ravnomjerno raspoređen na intervalu. Takođe se može primjetiti da je nivo unesenog šuma veći u dijelu spektra od 4-24kHz ali je taj šum maskiran tonalnim komponentama i ne utiče na percepciju audio signala.

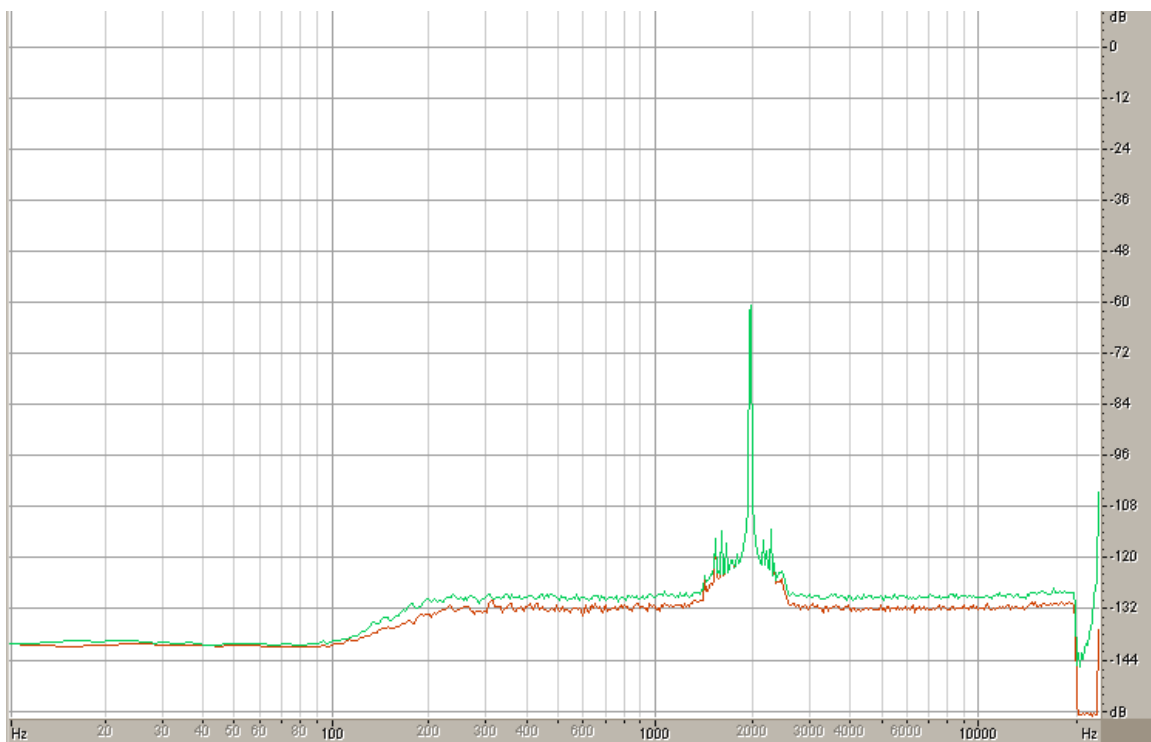


Slika 5.6 Multitonski signal

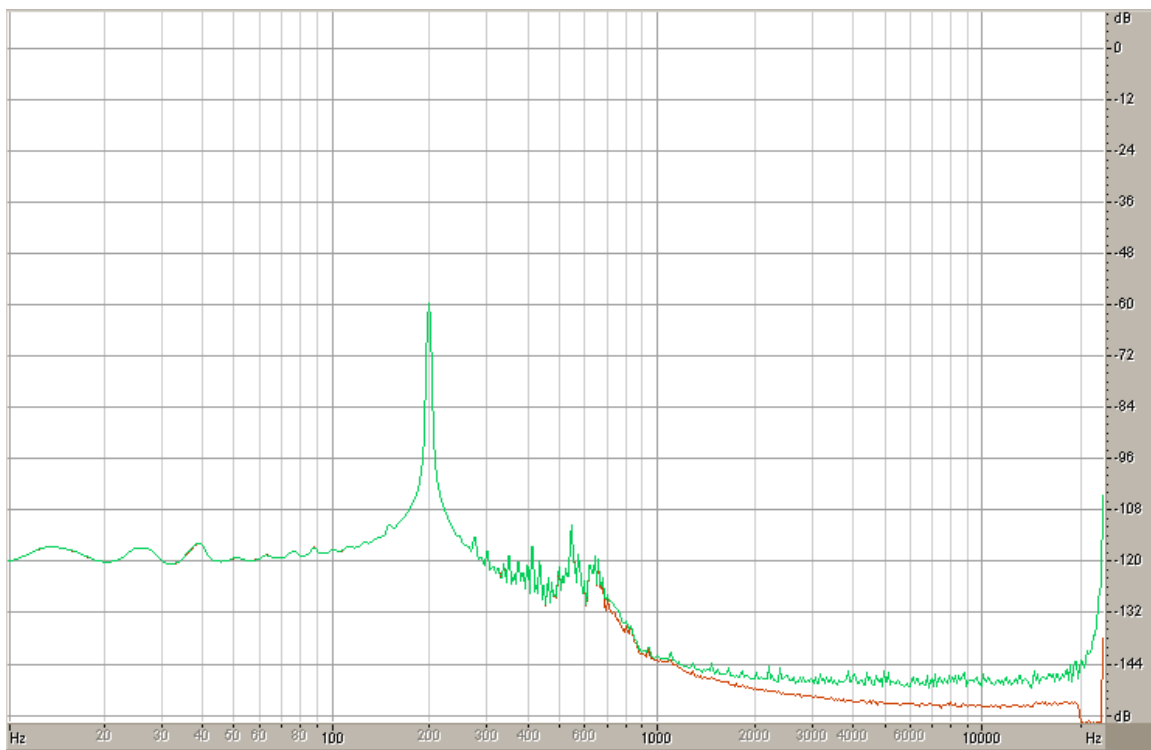
Sa sličnim ciljem je kreirana grupa testova u kojima se koristi sinusni signal na frekvencijama 20 Hz, 200 Hz, 2kHz i 20kHz pri nivoima 0dB, -20dB, -40dB i -60 dB. Najveća odstupanja su uočena na signalima sa nivoom od -60dB i njihove spektralne karakteristike su prikazane na slikama 5.7, 5.8, 5.9 i 5.10. Na pomenutim slikama se može primjetiti da odstupanje nivoa šuma izlaznog signala testiranog dekodera ne odstupa više od 5 dB u odnosu na referentni nivo šuma.



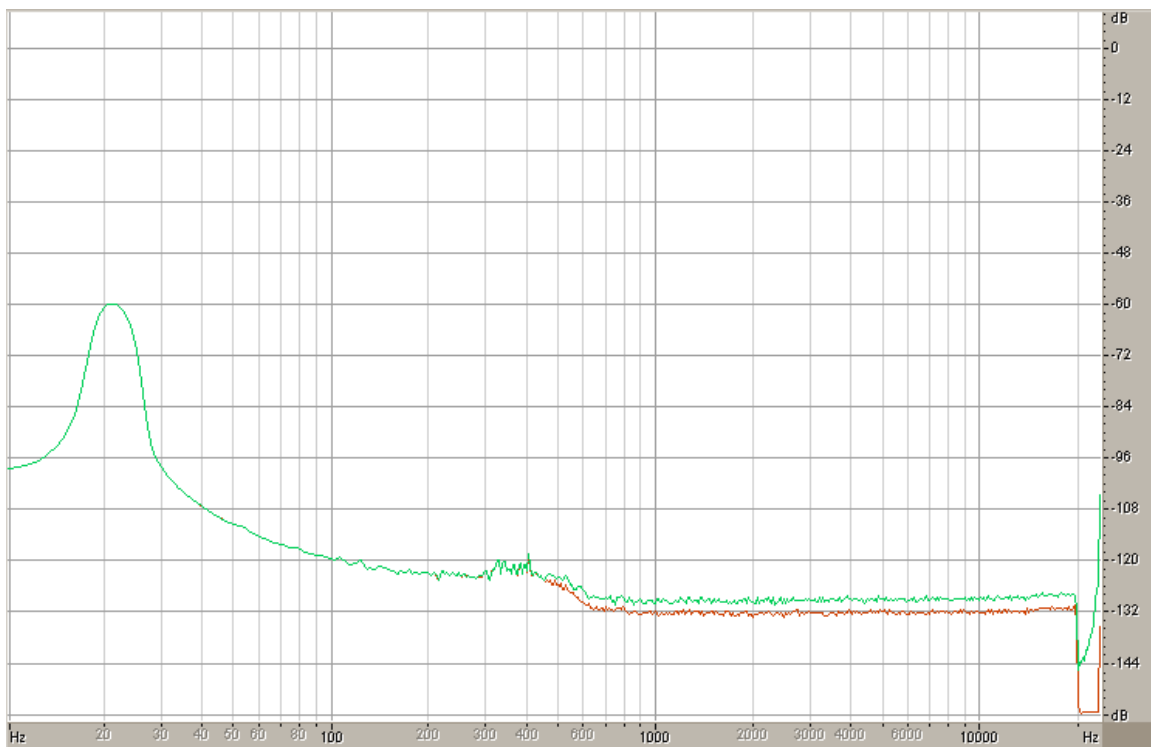
Slika 5.7 Sinusni signal frekvencije 20kHz, nivoa -60dB



Slika 5.8 Sinusni signal frekvencije 2kHz, nivoa -60dB

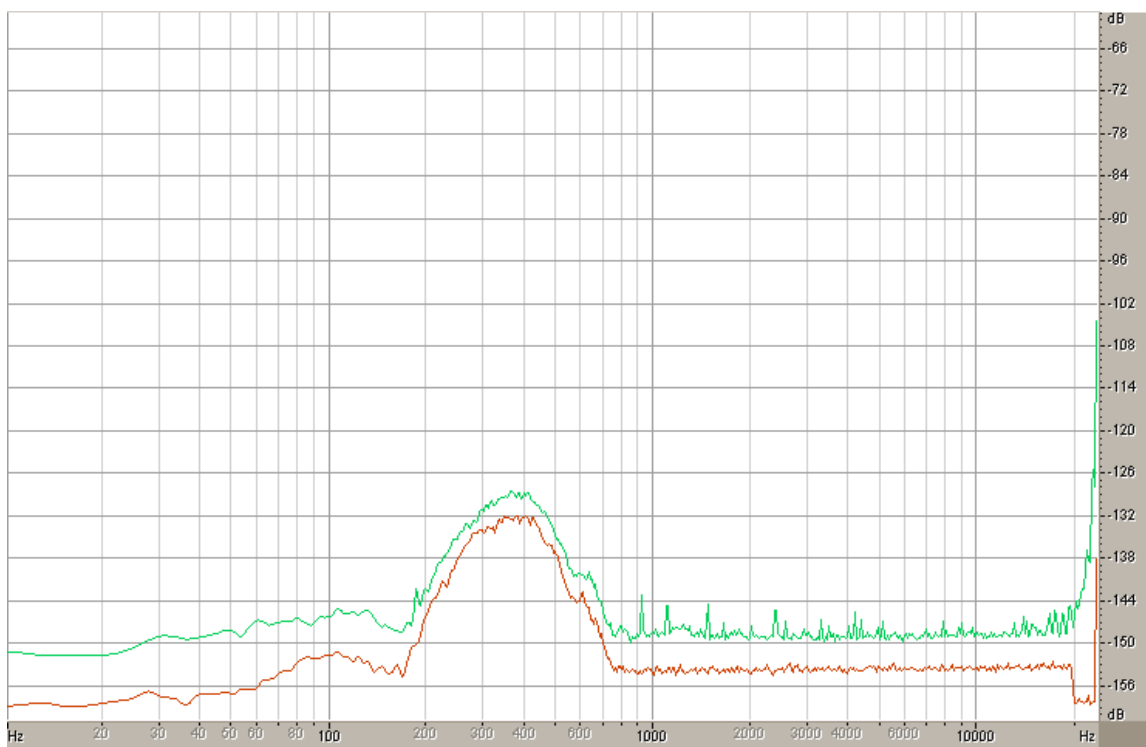


Slika 5.9 Sinusni signal frekvencije 200Hz, nivoa -60dB



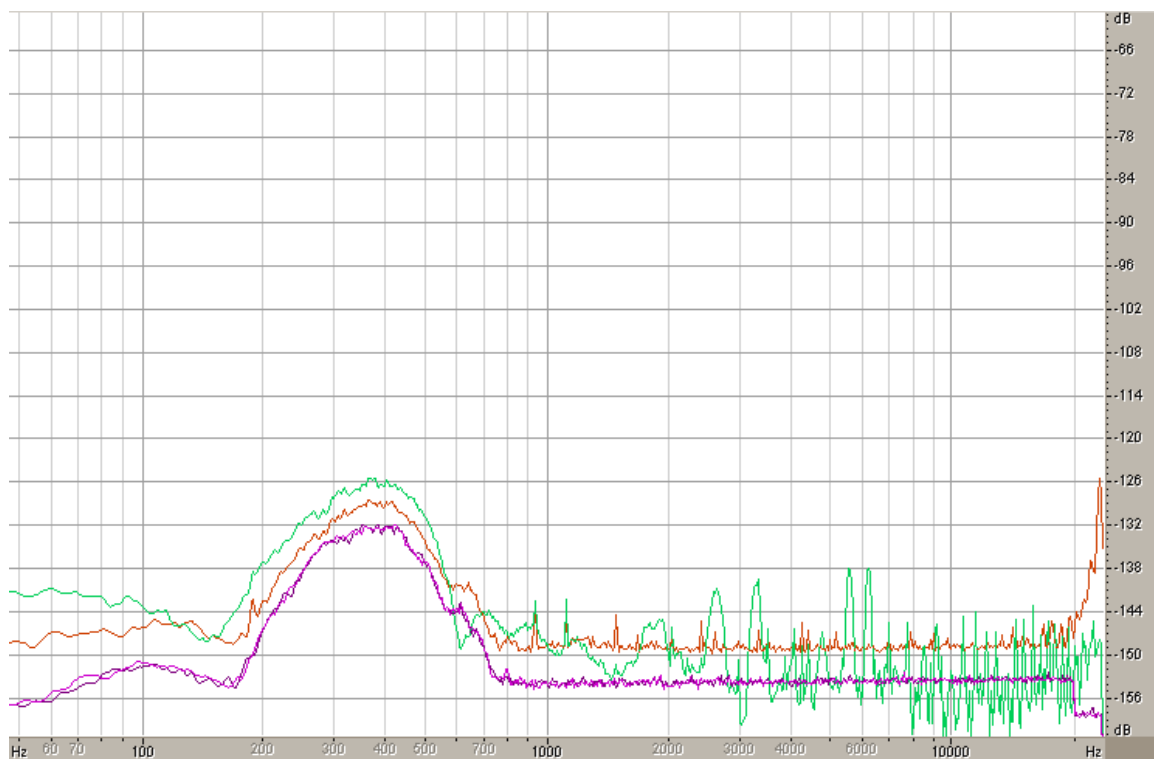
Slika 5.10 Sinusni signal frekvencije 20 Hz, nivoa -60dB

U trećem testu se koristi signal u kojem je prisutan šum niskog nivoaa lokalizovan u uskom dijelu spektra nastao variranjem bita najmanje težine u 20-bitnom PCM signalu. Na slici se može uočiti odstupanje nivoaa signala koje se kreće u opsegu od 3-7 dB kao i pojava spektralnih vrhova (engl. “spectral peaks”) male amplitude. Spektralni vrhovi ukazuju na pojavu tonalnih komponenti u signalu niskog nivoaa. U slučaju da je nivoaa tonalnih komponenti viši bilo bi potrebno povećati tačnost testiranog dekodera.



Slika 5.11 Uskofrekventni šum niskog nivoaa

Na slici 5.12 je zajedno sa signalima sa slike 5.11 prikazan signal napravljen dekoderom koji ne posjeduje potrebnu tačnost. Ukupna količina šuma unijeta ovim dekoderom je malo veća ako ne i približna količini šuma koje je unio realizovani dekoder ali je taj šum raspoređen tako da se može otkriti slušanjem (lako primjetna razlika u odnosu na prethodna dva signala). Ovaj primjer pokazuje da mjera kvaliteta iskazana odnosom signal-šum nije u potpunosti primjenljiva prilikom ocjene kvaliteta realizacije dekodera algoritma za kopresiju audio signala koji se zasniva na perceptualnim metodama.



Slika 5.12 Uskofrekventni šum niskog nivoa – “tonalni šum”

6. Zaključak

U okviru ovog magistarskog rada izvršena je analiza metoda za izračunavanje transformacija (MDCT, DCT, DFT) primjenom brzih algoritama koji se zasnivaju na brzom Furijeovoj transformaciji. Greške nastale prilikom izračunavanja transformacija u aritmetici nepokretnog zareza analizirane su na primjeru radix-2 algoritma brze Furijeove transformacije.

Izvršena je analiza prostiranja greške i predložene su metode za proširenje operacija kompleksnog sabiranja i kompleksnog množenja koje omogućuju sprečavanje eventualnog prekoračenja uz postizanje minimalne greške i uz minimalan broj dodatnih operacija potrebnih za popravke. Analiza je izvršena teoretski uz korištenje statističkog modela grešaka kao i simulacijom analiziranih metoda na primjerima nekih karakterističnih signala.

Predložene metode predstavljaju kompromis između zahtjevane brzine obrade i raspoloživih resursa i moguće ih je prilagoditi ciljnoj aplikaciji odnosno primjeniti na DSP platformi što je pokazano na primjeru realizacije AC3 dekodera na digitalnom signalnom procesoru MAS firme Micronas.

Naredna istraživanja bi trebala dovesti do stvaranja parametrizovanog modela za određivanje tačnosti implementacije brze Furijeove transformacije u zavisnosti od primjenjenih rješenja i arhitekture sistema (širina memorijske riječi, registara, akumulatora, kao i raspoloživosti i vrsti mehanizama za praćenje i kontrolu porasta maksimalne vrijednosti podržanih arhitekturom).

7. Literatura

- [1] M. Temerinac and B. Edler, "Ovelaping Block Transform: Window Design, Fast algorithm, and Image Coding Experiment," *IEEE Trans. Commun.*, September 1995.
- [2] M. Temerinac and B. Edler, "LOT"
- [3] H. Malvar, Lapped Transforms, *Elektronik Letters*, 1992
- [4] J. G. Proakis and D. G. Manolakis, "Digital Signal Processing"
- [5] G.R.L. Sohie and W. Chen, "Implementation of Fast Furier Transforms on Motorola's Digital Signal Processors"
- [6] M. Dietz, "AAC LC Decoder source code for LSI development", Fraunhofer Institut Integrierte Schaltungen, Erlangen, April 1999
- [7] , "MPEG2 AAC Decoder Low Complexity Implementation for the MASC 20 bit DSP", Fraunhofer Institut Integrierte Schaltungen, Erlangen
- [8] MAS 35xx Manual, March 1999
- [9] J. P. Princen and A. B. Bredley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-34, pp. 1153-1161, Oct. 1986.
- [10] C. C. Todd, G. A. Davidson, M. F. Davis, L. D. Fielder, B. D. Link and S. Vernon, "AC-3: Flexible Perceptual Coding for Audio Transmission and Storage," *Dolby Laboratories*, San Francisco, 1994
- [11] H. Malvar, "A Modulated Complex Lapped Transform and its Applications to Audio Processing," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ, pp. 1421-1424, March 1999.
- [12] ISO/IEC 11172, *Coding of Moving Pictures and Asociated Audio for Digital Storage Media up to about 1.5 Mbit/s: Part 3 Audio*, International Standards Organisation
- [13] ISO/IEC 13818, *Coding of Moving Pictures and Asociated Audio: Part 3*, International Standards Organisation
- [14] *Digital Audio Compression Standard (AC-3)*, United States Advanced Television Systems Committee (ATSC), Doc. A/52, Dec 1995. Internet adresa: <http://www.atsc.org/Standards/A52/>.
- [15] ISO/IEC, 13 818-7, "Information technology—Generic coding of moving pictures and associated audio—Part 7: Advanced audio coding," 1997.
- [16] Overview of the MPEG-4 standard, Internet adresa <http://www.cselt.it/mpeg/standards/mpeg-4/mpeg-4.html>, July 1998.
- [17] B. Edler, "Very Low Bit Rate Audio Coding Development"
- [18] Dolby Laboratories, "Test Procedure for AC-3 Decoders Using Test Files," *Dolby Document S96/11283*
- [19] T. Thong, "Fixed-Point Fast Fourier Transform Error Analysis," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-24, No. 6, December 1976.
- [20] F. O. Witte, M. Dietz and W. Sinnhofer, "Single Chip Implementation of an ISO/MPEG Layer III Decoder," *AES 96 convention*, Amsterdam, March 1994
- [21] V. Britanak and K. R. Rao, "A new fast algorithm for the unified forward and inverse MDCT/MDST computation," *Signal Processing* 82, 433-459 (2002)
- [22] G. Soulodre et al., "Subjective evaluation of state-of-the-art twochannel audio codecs," *J. Aud. Eng. Soc.*, vol. 46, no. 3, pp. 164-177, Mar. 1998.

-
- [23] J. C. Abel, M. A. Julier, “*Implementation of a High-Quality Dolby Digital Decoder Using MMXTM Technology*,” Intel Technology Journal Q3 ‘97
 - [24] Clifford Paris, Danny Wong and Steven Roberts: “*Minimal Word-width implementation of Dolby AC-3 and ISO MPEG Audio Decompression Standards*”
 - [25] G.A. Souldre, T. Grusec, M. Lavoie, and L. Thibault, “*Subjective Evaluation of State-of-the-Art Two-Channel Audio Codecs*,” Journal Audio Eng. Soc., Vol.46, No 3, 1998 March
 - [26] C. Liu and W. Lee, “*A Unified Fast Algorithm for Cosine Modulated Filter Banks in Current Audio Coding Standards*,”
 - [27] K. Ramkishor and U. Satyanarayana, “*Methods to Improve Accuracy in Fixed-point Implementation of IDCT*”
 - [28] R. M. Gray and D. L. Neuhoff, “*Quantization*”, IEEE Trans. On Information Theory, Vol. 44, No. 6, October 1998
 - [29] A. V. Oppenheim and C. J. Weinstein, “*Effects of Finite Register Length in Digital Filtering and Fast Fourier Transform*,” Proceedings of the IEEE, Vol. 60, No. 8, August 1972
 - [30] M.. Jeong, S. Kim, J. Sohn, J. Kang, and W. Sung, “*Finite Wordlength Effects Evaluation of the MPEG-2 Audio Decoder*”
 - [31] Y. Wang, L. Yaroslavski, M. Vilermo, and M. Väänänen, “*Some Peculiar Properties of the MDCT*”
 - [32] B. Widrow, I. Kolar and M. Liu, “*Statistical Theory of Quantization*”
 - [33] T. Painter, A. Spanias, “*Perceptual coding of digital audio*,” Proc. IEEE 88 (4) (April 2000) 451–513.
 - [34] P. P. Vaidyanathan and S. Akkarakaran, “*A Review of the Theory and Applications of Optimal Subband and Transform Coders*”, Applied and Computational Harmonic Analysis 10, 2001
 - [35] S. Jeng, “*Audio Signal Processing - Time to Frequency Mapping*”
 - [36] R. Gluth, “*Regular FFT-Related Transform Kernels for DCT/DST-Based Polyphase Filter Banks*,” IEEE, 1991
 - [37] V. Nikolajevic and G. Fettweis, “*New Recursive Algorithms for the Forward and Inverse MDCT*,” in Proc. IEEE Workshop Signal Process.Syst., Antwerp, Belgium, Sept. 2001.