



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ

---



Мирослав Малко

**Једно решење алгоритма за  
потискивање шума у говорном сигналу  
на платформи са ограниченим  
ресурсима**

МАСТЕР РАД

Нови Сад, 2012



УНИВЕРЗИТЕТ У НОВОМ САДУ • ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА  
21000 НОВИ САД, Трг Доситеја Обрадовића 6

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, <b>РБР</b> :		
Идентификациони број, <b>ИБР</b> :		
Тип документације, <b>ТД</b> :	Монографска документација	
Тип записа, <b>ТЗ</b> :	Текстуални штампани материјал	
Врста рада, <b>ВР</b> :	Дипломски – мастер рад	
Аутор, <b>АУ</b> :	<b>Мирослав Малко</b>	
Ментор, <b>МН</b> :	<b>Др. Јелена Ковачевић</b>	
Наслов рада, <b>НР</b> :	<b>Једно решење алгоритма за потискивање шума у говорном сигналу на платформи са ограниченим ресурсима</b>	
Језик публикације, <b>ЈП</b> :	Српски / латиница	
Језик извода, <b>ЈИ</b> :	Српски	
Земља публикавања, <b>ЗП</b> :	Република Србија	
Уже географско подручје, <b>УГП</b> :	Војводина	
Година, <b>ГО</b> :	<b>2012</b>	
Издавач, <b>ИЗ</b> :	Ауторски репринт	
Место и адреса, <b>МА</b> :	Нови Сад; трг Доситеја Обрадовића 6	
Физички опис рада, <b>ФО</b> : (поглавља/страна/ цитата/табела/слика/графика/прилога)	<b>7/36/11/9/13/0/0</b>	
Научна област, <b>НО</b> :	Електротехника и рачунарство	
Научна дисциплина, <b>НД</b> :	Рачунарска техника	
Предметна одредница/Кључне речи, <b>ПО</b> :	<b>ДСП, потискивање шума, SMNS, Speex, обрада говора, одузимање спектра</b>	
<b>УДК</b>		
Чува се, <b>ЧУ</b> :	У библиотеци Факултета техничких наука, Нови Сад	
Важна напомена, <b>ВН</b> :		
Извод, <b>ИЗ</b> :	<b>У овом раду имплементиран је једномикрофонски алгоритам за потискивање шума, SMNS (енг. Single Microphone Noise Suppression) алгоритам, на CS49834 процесору фирме Cirrus Logic. Квалитет имплементације алгоритма упоређен је са јавно доступним Speex алгоритмом исте намене, користећи метрику која се примењује приликом испитивања оваквих типова алгоритама. Циљ рада је био утврдити квалитет имплементације SMNS алгоритма, као и количину ресурса потребну за имплементацију на физичкој платформи.</b>	
Датум прихватања теме, <b>ДП</b> :		
Датум одбране, <b>ДО</b> :		
Чланови комисије, <b>КО</b> :	Председник: <b>Др. Иштван Пап</b>	
	Члан: <b>Др. Растислав Струхарик</b>	Потпис ментора
	Члан, ментор: <b>Др. Јелена Ковачевић</b>	



## KEY WORDS DOCUMENTATION

Accession number, <b>ANO</b> :		
Identification number, <b>INO</b> :		
Document type, <b>DT</b> :	Monographic publication	
Type of record, <b>TR</b> :	Textual printed material	
Contents code, <b>CC</b> :	Master Thesis	
Author, <b>AU</b> :	<b>Miroslav Malko</b>	
Mentor, <b>MN</b> :	<b>PhD Jelena Kovacevic</b>	
Title, <b>TI</b> :	<b>One solution of algorithm for suppression of noise in speech signal on an embedded platform</b>	
Language of text, <b>LT</b> :	Serbian	
Language of abstract, <b>LA</b> :	Serbian	
Country of publication, <b>CP</b> :	Republic of Serbia	
Locality of publication, <b>LP</b> :	Vojvodina	
Publication year, <b>PY</b> :	<b>2012</b>	
Publisher, <b>PB</b> :	Author's reprint	
Publication place, <b>PP</b> :	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, <b>PD</b> : <small>(chapters/pages/ref./tables/pictures/graphs/appendixes)</small>	<b>7/36/11/9/13/0/0</b>	
Scientific field, <b>SF</b> :	Electrical Engineering	
Scientific discipline, <b>SD</b> :	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, <b>S/KW</b> :	<b>DSP, noise suppression, SMNS, Speex, voice processing, spectral subtraction</b>	
<b>UC</b>		
Holding data, <b>HD</b> :	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, <b>N</b> :		
Abstract, <b>AB</b> :	<b>In this work single microphone algorithm for noise suppression is implemented, SMNS algorithm, on a CS49834 processor made by Cirrus Logic. Quality of implementation of this algorithm is compared with Speex algorithm, which is used for the same purpose, and that is by using standard metrics used to verify this types of algorithms. Goal of this work was to evaluate quality of implementation of SMNS algorithm, as well as amount of resources needed for implementation on a DSP platform.</b>	
Accepted by the Scientific Board on, <b>ASB</b> :		
Defended on, <b>DE</b> :		
Defended Board, <b>DB</b> :	President: <b>PhD Istvan Pap</b>	
	Member: <b>PhD Rastislav Struharik</b>	Menthor's sign
	Member, Mentor: <b>PhD Jelena Kovacevic</b>	

**SADRŽAJ**

1. Uvod.....	1
2. Teorijske osnove.....	3
2.1 Blok za transformaciju signala iz vremenskog u frekventni domen.....	4
2.2 Blok obrade.....	5
2.2.1 Opšti principi algoritama zasnovanih na metodi oduzimanja spektara.....	6
2.2.2 Nedostaci algoritama za oduzimanje spektara.....	8
2.2.3 Višeopsežni algoritam za oduzimanje spektara.....	9
3. Koncept rešenja.....	12
3.1 Model 0 .....	13
3.2 Model 1.....	13
3.3 Model 2.....	14
3.4 Model 3.....	14
3.4.1 Razvojno okruženje CLIDE.....	15
3.5 Finalna verzija koda.....	16
4. Implementacija.....	17
4.1 Detalji implementacije.....	17
4.1.1 Model 0.....	17
4.1.2 Model 1.....	18
4.1.3 Model 2.....	18
4.1.4 Model 3 – kod izvršiv na ciljnoj platformi.....	18
4.2 Tehničke specifikacije ciljne platforme.....	19
4.3 Iskorišćenost resursa datog rešenja na ciljnoj platformi.....	20
5. Ispitivanje i verifikacija.....	22

---

5.1 Subjektivne metode merenja.....	22
5.2 Objektivne metode merenja.....	24
5.3 Opis ispitnih vektora.....	25
5.4 Bit-identična ispitivanja.....	25
5.5 Rezultati ispitivanja objektivnim metodama merenja.....	27
6. Zaključak.....	33
7. Literatura.....	35

**SPISAK SLIKA**

Slika 2.1: Jednostavni blok dijagram redosleda obrade.....	4
Slika 2.2: Opšti oblik algoritma za oduzimanje spektara .....	8
Slika 3.1: Tok implementacije algoritma.....	12
Slika 3.2: Razvojno okruženje CLIDE.....	15
Slika 5.1: Prikaz BBT alata.....	26
Slika 5.2: PESQ odnos za ispitne vektore muškog govornika, odnos signal-šum 12dB.....	27
Slika 5.3: SNRE odnos za ispitne vektore muškog govornika, odnos signal-šum 12dB.....	28
Slika 5.4: PESQ odnos za ispitne vektore muškog govornika, odnos signal-šum 6dB.....	28
Slika 5.5: SNRE odnos za ispitne vektore muškog govornika, odnos signal-šum 6dB.....	29
Slika 5.6: PESQ odnos za ispitne vektore ženskog govornika, odnos signal-šum 12dB.....	29
Slika 5.7: SNRE odnos za ispitne vektore ženskog govornika, odnos signal-šum 12dB.....	30
Slika 5.8: PESQ odnos za ispitne vektore ženskog govornika, odnos signal-šum 6dB.....	30
Slika 5.9: SNRE odnos za ispitne vektore ženskog govornika, odnos signal-šum 6dB.....	31

**SPISAK TABELA**

Tabela 4.1: Količina raspoložive memorije unutar CS49834 čipa.....	19
Tabela 4.2: Iskorišćenost resursa datog rešenja.....	20
Tabela 4.3: Detaljna analiza potrošnje procesorskog vremena.....	21
Tabela 5.1: Skala subjektivne ocene kvaliteta govornog signala.....	23
Tabela 5.2: Tipovi smetnje korišćeni pri kombinaciji sa čistim signalom govora.....	25
Tabela 5.3: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore muškog govornika, odnos signal-šum 12dB.....	31
Tabela 5.4: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore muškog govornika, odnos signal-šum 6dB.....	32
Tabela 5.5: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore ženskog govornika, odnos signal-šum 12dB.....	32
Tabela 5.6: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore ženskog govornika, odnos signal-šum 6dB.....	32

## SKRAĆENICE

**SMNS** - *Single Microphone Noise Suppression*, Jednomikrofonski algoritam za potiskivanje smetnji u govornom signalu

**DSP** - *Digital Signal Processor*, Digitalni signal procesor

**FFT** - *Fast Fourier Transform*, Brza Furijeova transformacija

**SNR** - *Signal-to-Noise Ratio*, Odnos signal-šum

**FPU** - *Floating Point Unit*, Jedinica za rad sa aritmetikom u pokretnom zarezu

**CLIDE** - *Cirrus Logic Integrated Development Environment*, Integrisano razvojno okruženje namenjeno Crystal DSP familiji procesora firme Cirrus Logic

**SRS** - *Shift Round Saturate*, Jedinica sa aritmetičko pomeranje, zaokruživanje i saturaciju Crystal DSP familije procesora

**DMA** - *Direct Memory Access*, Direktan pristup memoriji

**SPDIF** - *Sony Philips Digital InterFace*, Sony Philips digitalni sprežni podsistem

**HDMI** - *High Definition Multimedia Interface*, Multimedijalni sprežni podsistem visoke definicije

**MIPS** - *Million Instructions Per Second*, Milion instrukcija u sekundi

**MOS** - *Mean Opinion Score*, Srednja vrednost mišljenja

**PSQM** - *Perceptual Speech Quality Measure*, Perceptualna mera za ocenu kvaliteta govora

**PESQ** - *Perceptual Evaluation of Speech Quality*, Perceptualna procena kvaliteta govora

**SNRE** - *Signal-to-Noise Ratio Enhancement*, Mera poboljšanja odnosa signal-šum

**BBT** - *Black Box Testing*, Ispitivanje metodom crne kutije

## 1. Uvod

U okviru ovog rada urađena je implementacija jednog algoritma za potiskivanje šuma u govornom signalu na platformi sa ograničenim resursima, odnosno DSP (*eng. Digital Signal Processor*) platformi. Implementacija je urađena na CS49834 čipu iz Crystal DSP familije procesora, firme Cirrus Logic. Polazna tačka ovog rada je bio jednomikrofonski algoritam za potiskivanje šuma SMNS (*eng. Single Microphone Noise Suppression*). Ovaj algoritam zasnovan na metodi oduzimanja spektara (*eng. spectral subtraction*) je razvijen na Katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu i realizovan u programskom jeziku C. Kvalitet realizovanog algoritma je upoređen sa jednim javno dostupnim algoritmom iste namene, Speex [1] algoritmom.

Razvoj telekomunikacija počeo još u doba prvih ljudskih civilizacija. Rane telekomunikacije podrazumevale su korišćenje dimnih signala i bubnjeva. Zanimljivo, ovakve primitivne telekomunikacije rane civilizacije koristile su za slanje više različitih informacija, kako za označavanje prisustva, tako i za pozive u pomoć i brojne druge poruke. Ipak, tek pronalaskom telegrafa i telefona utaban je put modernim telekomunikacijama. Brojni naučnici i pronalazači tokom 19. veka direktno su učestvovali u prvim uspešnim slanjima i prijemima telegrafskog i telefonskog signala, kako u granicama jednog kontinenta, tako i između dva kontinenta (prvi uspešni transatlantski kabl položen je 1866. godine i služio je za telegrafске primene između Severne Amerike i Evrope). Prenos govornog signala telefonom prvi put je uspešno obavljen 1878. godine i od tad kreće nezadrživ razvoj ovog vida komunikacije. U prvoj polovini 20. veka bežični prenos radio vezom je već prilično razvijen, a počinju i prvi eksperimenti sa televizijskim prenosom. Sa napretkom tehnologije u drugoj polovini 20. veka, dolazi do usavršavanja telekomunikacija, pre svega televizije i radija. Pronalaskom tranzistora, a zatim i razvojem poluprovodničke tehnologije, dolazi do pronalaska prvih računara koji će kako direktno, tako i indirektno uticati na dalji razvoj telekomunikacija. Internet, satelitske

komunikacije, mobilni telefoni, sve su to vidovi komunikacije koje savremeno društvo koristi svakodnevno i bez kojih današnji čovek gotovo da ne može da funkcioniše. U moderno doba, računari su postali neizostavni deo praktično svakog uređaja, a njihovo stalno poboljšavanje povlači sa sobom i stalno unapređivanje komunikacija u globalnom smislu te reči. Primena računara i digitalne obrade signala dovela je do minijaturizacije uređaja, povećanja kvaliteta kao i pristupačnosti.

Porastom kompleksnosti komunikacija, kao i različitih uslova u kojima komunikacije današnjice moraju da funkcionišu, dolazi i do porasta verovatnoće pojave smetnje prilikom slanja informacija. Ako je reč o govornoj komunikaciji, smetnjom se može smatrati sve što nije signal govornika. Današnji razvoj govornih komunikacija sastoji se upravo u odstranjivanju svega što nije koristan signal, odnosno težnja da se kvalitet komunikacije dovede na što viši nivo. Razumljivost govora je od ključnog značaja za kvalitet govorne komunikacije, te algoritmi za potiskivanje akustičkih smetnji imaju veliku primenu u telefonskim i radio komunikacijama, gde se koriste za potiskivanje ambijentalne buke. Pojava smetnji u govornim komunikacijama je najviše došla do izražaja pojavom mobilnih telefona i njihovim korišćenjem na najraznovrsnijim mestima, kao što su javni prevoz, koncerti, aerodrom, industrijski pogoni, gradilišta i slično. Na ovakvim mestima dolazi do otežane komunikacije upravo zbog povećanog prisustva buke, i jedan od izazova digitalne obrade signala je upravo da tu buku odstrani i da se vrši samo prenos korisnog signala. Uspešnom analizom buke i njenim odstranjivanjem iz ukupnog signala, prenos signala svodi se na prenos samo korisnog signala, što ima za posledicu rasterećenje prenosnog kanala.

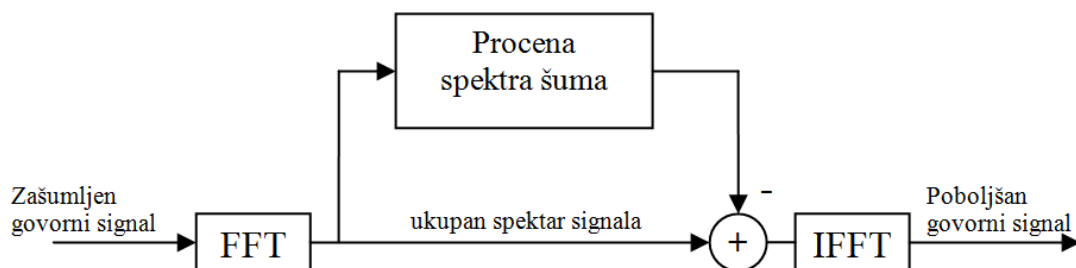
Rad je organizovan u sedam poglavlja. U drugom poglavlju opisane su teorijske osnove SMNS algoritma. U trećem poglavlju opisana je metodologija korišćena prilikom izrade ovog rada. U četvrtom poglavlju izneti su detalji implementacije algoritma na ciljnoj platformi, kao i tehničke specifikacije ciljne platforme. U petom poglavlju opisan je postupak ispitivanja i verifikacije datog rešenja. Šesto poglavlje sadrži zaključak, gde je ukratko sažeto sve je što je u okviru ovog rada urađeno, kao i zaključci do kojih se došlo prilikom izrade ovog rada. Poslednje, sedmo poglavlje sadrži spisak korišćene literature prilikom izrade ovog rada.

## 2. Teorijske osnove

Jednomikrofonski algoritam za potiskivanje buke (*eng. Single Microphone Noise Suppression*, u daljem tekstu SMNS) zasniva svoj rad na algoritmu oduzimanja spektara [3] (*eng. Spectral Substraction Algorithm*). Metod oduzimanja spektara vrši procenu spektra šuma i vrši oduzimanje ovako procenjenog spektra od ukupnog spektra signala. Ključni deo ovakvih algoritama je utvrđivanje govorne aktivnosti, odnosno određivanje kad je govor prisutan a kad ne. U odsustvu govora vrši se procena spektra šuma, a zatim se taj procenjeni spektar šuma oduzima od ukupnog spektra signala. Algoritmi koji zasnivaju rad na metodi oduzimanja spektara imaju prednost da su jednostavni za implementaciju. Ipak, postoje i druge vrste algoritama koje se primenjuju pri odstranjivanju šuma. Tzv. CASA (*eng. Computational Auditory Scene Analysis*) algoritmi zasnivaju svoj rad na modelovanju ljudskog slušnog aparata. Algoritmi ove vrste zasnivaju svoj rad na činjenici da ljudski slušni aparat i pored prisustva buke može da se usredsredi na govor, odnosno ljudski slušni aparat grupiše govor i šum u odvojene tokove. Ovakvi algoritmi pokušavaju da utvrde i odvoje tok šuma od toka govora (slično kao ljudsko uho) i ponište tok šuma. Kao poboljšanje jednomikrofonskog algoritma za oduzimanje spektara postoji i dvomikrofonski algoritam za oduzimanje spektara, kod kojeg je predviđeno da je jedan mikروفon postavljen bliže ustima, dok je drugi mikروفon tipično sa druge strane uređaja (npr. mobilnog telefona). Ovakvi algoritmi bolje detektuju govornu aktivnost, koja je ključna kod ovakvih tipova algoritama. Dodatno, signal sa drugog mikrofona može da se koristi za prostorno filtriranje (*eng. beam forming*). Prostorno filtriranje zasniva svoj rad na utvrđivanju pravca iz kojeg dolazi signal govora, dok se ostali pravci smatraju izvorom smetnje. Slično, postoji i inverzno prostorno filtriranje koje utvrđuje pravac izvora šuma, dok se ostali pravci smatraju izvorom govora [2]. Ipak, ova metoda podrazumeva korišćenje dva i više mikrofona, što često nije moguće unutar nekog uređaja. Stoga jednomikrofonski algoritmi za potiskivanje šuma predstavljaju interesantnu oblast istraživanja.

Kako i samo ime algoritma kaže, SMNS algoritam je namenjen sistemima sa jednim mikrofonom (*single microphone* – jedan mikrofoni). On ima za cilj da što bolje proceni spektr šuma, pri čemu se polazi od pretpostavke da je šum aditivan, odnosno da se ukupan signal na predajnoj strani može opisati kao zbir korisnog signala i signala šuma (koji potiče bilo od drugog govornika pored, bilo od muzike na koncertu, gradilišta itd.).

Sam redosled obrade je sledeći: najpre se ukupan signal pretvori iz vremenskog u spektralni domen. Zatim se vrši procena spektra šuma i oduzimanje ovog spektra od ukupnog spektra signala. Na ovaj način se dobija procenjeni spektr govora, koji se na kraju vraća iz frekventnog u vremenski domen, gde je kraj lanca obrade i signal je u nekoj meri poboljšan [3] (mere poboljšanja i rezultati u poglavlju ispitivanje i verifikacija). Jednostavni blok dijagram obrade prikazan je na slici 2.1.



Slika 2.1: Jednostavni blok dijagram redosleda obrade

Svaka od logičkih celina SMNS algoritma je detaljnije objašnjena u poglavljima koja slede.

## 2.1 Blok za transformaciju signala iz vremenskog u frekventni domen

Kako sam algoritam zahteva obradu u frekventnom domenu, na početku obrade ukupnog signala vrši se prevođenje signala iz vremenskog u frekventni domen. Zbog svoje jednostavnosti, kao i postojanju razvijenih brzih algoritama za njeno računanje, koristi se Furijeova transformacija, koja se definiše kao:

$$X(\Omega) = \sum_{n=0}^{\infty} x(n)e^{-jn\Omega} \quad (1)$$

U konkretnom rešenju, korišćen je FFT algoritam (*eng. Fast Fourier Transform*), koji je pogodan za primenu na platformama sa ograničenim resursima, koje podrazumevaju rad u realnom vremenu sa minimalnim kašnjenjem. Stoga, obrada signala se vrši po blokovima, pri

čemu je veličina bloka broj koji je stepen broja dva (zbog efikasnosti FFT algoritma). Kako se obrada vrši po blokovima, moguća je pojava diskontinuiteta u spektru (što su manji blokovi to je veća mogućnost pojave diskontinuiteta, ali je manje kašnjenje – i obrnuto). Da bi se umanjio ovaj efekat susedni okviri se preklapaju.

Pošto je rezultat diskretne Furijeove transformacije diskretan spektar, frekvencije čija perioda nije celobrojan umnožak dužine okvira ne mogu biti predstavljene jednim koeficijentom, već se njihova energija raspoređuje na više koeficijenata. Ova pojava se naziva efekat “curenja spektra”. Da bi se izbegao efekat „curenja spektra”, signal se pre primene Furijeove transformacije (dakle u vremenskom domenu) množi sa prozorskom funkcijom. Da bi se signal mogao ponovo vratiti iz frekventnog u vremenski domen, ista prozorska funkcija koja se koristi u analizi signala se obično koristi i u sintezi signala. Jedna od najčešće korišćenih prozorskih funkcija je Hannova funkcija, a ona je korišćena i u ovom rešenju. Razlog za njeno često korišćenje je što ispunjava uslov perfektno rekonstrukcije – što bi teoretski značilo da ukoliko bi uradili transformaciju iz vremenskog u frekventni domen, a zatim, bez ikakve obrade, uradili transformaciju iz frekventnog u vremenski domen, dobili bi potpuno isti signal kao i pre transformacija. Ovo u praksi naravno nije slučaj zbog grešaka nastalih prilikom aritmetičkih operacija, ali je jako blizu ovom idealnom slučaju.

## 2.2 Blok obrade

Kako je ranije napomenuto, SMNS algoritam zasniva svoj rad na metodi oduzimanja spektara. Ovaj algoritam je jedan od najstarijih algoritama osmišljenih za potrebe otklanjanja šuma na predajnoj strani. On zasniva svoj rad na vrlo jednostavnom principu. Pod pretpostavkom da je šum aditivan, spektar korisnog signala dobija se oduzimanjem procenjenog spektra šuma od ukupnog spektra signala, odnosno spektra signala i šuma. Ovakva pretpostavka je dobra ukoliko šum i koristan signal nisu korelisani, odnosno predstavljaju dva nezavisna slučajna signala.

Ovakav algoritam, u svojem najosnovnijem obliku ima jednu manu koja se sastoji u odluci u kojoj meri se oduzima procenjeni spektar šuma od ukupnog spektra signala. Naime, ukoliko se od ukupnog spektra oduzme previše, moguće je da će se izgubiti i neke korisne informacije, u ovom slučaju signal govora. Ukoliko se pak od ukupnog spektra oduzme premalo, pored signala govora ostaće prisutan i signal šuma. Mnogi radovi se bave ovom tematikom i kao rezultat toga, postoji veliki broj algoritama koji zasnivaju svoj rad na metodi oduzimanja spektara [5]. Svaki od tih algoritama ima svoje prednosti i mane i ne postoji univerzalno rešenje za svaki sistem. U nekom slučaju se zahtevaju visoke performanse sistema, dok u nekom drugom sistemu se zahteva što manja cena, čak i po cenu smanjenja kvaliteta. Konkretno, SMNS algoritam je

varijacija algoritma za oduzimanje spektara, i to višeopsežni (*eng. multiband*) algoritam za oduzimanje spektara. Slede najpre osnovni principi i formule koje su zajedničke za svaki algoritam za oduzimanje spektara, a zatim i specifični detalji vezani za višeopsežni algoritam za oduzimanje spektara.

### 2.2.1 Opšti principi algoritama zasnovanih na metodi oduzimanja spektara

Algoritmi za oduzimanje spektara polaze od pretpostavke da je šum aditivan, odnosno da se ukupan signal može smatrati zbirom signala govora (koristan signal) i signala šuma (nepoželjan deo ukupnog signala). Ovu pretpostavku možemo izraziti i formulom:

$$y(n) = x(n) + d(n) \quad (2)$$

pri čemu je  $y(n)$  ulazni (zašumljeni signal),  $x(n)$  signal govora i  $d(n)$  signal šuma. Ako uradimo prelazak iz vremenskog u frekventni domen primenom diskretne Furijeove transformacije, dobijamo:

$$Y(\omega) = X(\omega) + D(\omega) \quad (3)$$

gde su  $Y(\omega)$ ,  $X(\omega)$  i  $D(\omega)$  spektri ulaznog signala, signala govora i signala šuma respektivno. Ako izrazimo spektar ulaznog signala u polarnom obliku dobijamo:

$$Y(\omega) = |Y(\omega)|e^{j\phi_y(\omega)} \quad (4)$$

gde je  $|Y(\omega)|$  amplitudski spektar, a  $\phi_y(\omega)$  fazni spektar ulaznog signala. Spektar signala šuma se takođe može izraziti na sličan način kao i spektar ulaznog signala:

$$D(\omega) = |D(\omega)|e^{j\phi_d(\omega)} \quad (5)$$

Amplitudski spektar signala šuma ( $|D(\omega)|$ ) nije poznat, ali se može zameniti sa njegovom srednjom vrednošću izračunatom tokom delova signala gde signal govora nije prisutan (u pauzama tokom govora). Takođe, fazni spektar signala šuma  $\phi_d(\omega)$  može se zameniti sa  $\phi_y(\omega)$ , odnosno sa faznim spektrom ulaznog signala. Ovakva pretpostavka se temelji na činjenici da faza signala ne utiče na razumljivost govora, dok na sam kvalitet govora može uticati u određenoj meri. Uzimajući u obzir ove pretpostavke, izraz za procenjeni spektar signala govora postaje:

$$\hat{X}(\omega) = \left[ |Y(\omega)| - \left| \hat{D}(\omega) \right| \right] e^{j\phi_y(\omega)} \quad (6)$$

gde  $\left| \hat{D}(\omega) \right|$  predstavlja procenjeni amplitudski spektar šuma, izračunat tokom izostanka govorne aktivnosti. Stoga i simbol  $\hat{\phantom{x}}$  naglašava da je u pitanju procenjeni amplitudski spektar šuma. Ova formula je osnova algoritma za oduzimanje spektara. Najpre se izračuna amplitudski spektar ulaznog signala korišćenjem FFT algoritma. Zatim se vrši računanje procenjenog spektra

šuma u trenucima gde govorna aktivnost nije prisutna. U trenucima kada postoji govorna aktivnost ne vrši se računanje procenjenog spektra šuma, već se prethodno izračunata vrednost (tokom odsustva govorne aktivnosti) zadržava [4]. I tokom govorne aktivnosti, kao i u pauzama između delova govora, vrši se oduzimanje procenjenog spektra šuma od spektra ulaznog signala (otud i naziv za ovu metodu – oduzimanje spektara). Na kraju vrši se prevođenje signala iz frekventnog u vremenski domen, gde kao rezultat ove transformacije dobijamo poboljšan ulazni signal, odnosno šum je u određenoj meri potisnut.

Treba napomenuti da razlika između amplitudskog spektra ukupnog signala i procenjenog

amplitudskog spektra šuma,  $\left| \hat{X}(\omega) \right| = \left| |Y(\omega)| - \left| \hat{D}(\omega) \right| \right|$  može biti negativna. Ovo proizilazi iz nesavršenosti algoritma. Međutim, amplitudski spektar ne može biti negativan, i stoga se mora voditi pažnja kod oduzimanja procenjenog spektra šuma od spektra ulaznog (zašumljenog) signala da do ovoga ne dođe. Jedan od načina je se izbegnu negativne vrednosti amplitudskog spektra signala govora leži u tome da, ukoliko je procenjena vrednost spektra šuma veća od spektra ulaznog signala, spektar izlaznog signala odnosno spektar govora se u tom slučaju postavlja na nulu. Ovo je prikazano izrazom:

$$\left| \hat{X}(\omega) \right| = \begin{cases} \left| |Y(\omega)| - \left| \hat{D}(\omega) \right| \right| & , |Y(\omega)| > \left| \hat{D}(\omega) \right| \\ 0 & , |Y(\omega)| \leq \left| \hat{D}(\omega) \right| \end{cases} \quad (7)$$

Ovo je samo jedan od načina na koji se osigurava da spektar signala govora bude nenegativan. Različiti algoritmi rešavaju ovaj problem na različite načine, i između ostalog, i u tome se ogleda kvalitet jednog ovakvog algoritma za potiskivanje šuma (više o ovom problemu u narednom poglavlju).

Prethodno izlaganje o metodi za oduzimanje spektara korišćenjem spektra amplitude može se lako prošiti i na spektar snage. Neki algoritmi zahtevaju korišćenje spektra snage umesto amplitudskog spektra. Da bi se dobio spektar snage ulaznog (zašumljenog) signala, potrebno je  $|Y(\omega)|$ , odnosno amplitudski spektar ulaznog signala, pomnožiti sa njegovom konjugovano kompleksnom vrednošću. Urađivši to, izraz postaje:

$$\begin{aligned} |Y(\omega)|^2 &= |X(\omega)|^2 + |D(\omega)|^2 + X(\omega) \cdot D^*(\omega) + X^*(\omega) \cdot D(\omega) \\ &= |X(\omega)|^2 + |D(\omega)|^2 + 2 \operatorname{Re}\{X(\omega) \cdot D^*(\omega)\} \end{aligned} \quad (8)$$

Elementi jednačine,  $|D(\omega)|^2$ ,  $X(\omega) \cdot D^*(\omega)$  i  $X^*(\omega) \cdot D(\omega)$  se ne mogu odrediti direktno i umesto toga, date vrednosti se aproksimiraju očekivanim vrednostima,  $E\{|D(\omega)|^2\}$ ,

$E\{X(\omega) \cdot D^*(\omega)\}$  i  $E\{X^*(\omega) \cdot D(\omega)\}$  respektivno. Prvi se procenjuje tokom odsustva govorne aktivnosti, i obično se označava kao  $\left| \hat{D}(\omega) \right|^2$ . Ako pretpostavimo da šum nije korelisan sa govorom, onda se druga dva činioca aproksimiraju na nulu. Sa ovim pretpostavkama, izraz za procenu spektra snage čistog govornog signala postaje:

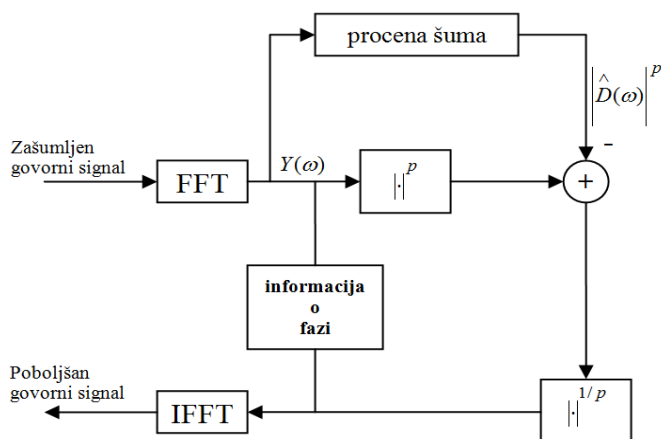
$$\left| \hat{X}(\omega) \right|^2 = |Y(\omega)|^2 - \left| \hat{D}(\omega) \right|^2 \quad (9)$$

Slično kao i kod izraza za procenu amplitudskog spektra čistog govornog signala, ni ovde ne postoji garancija da će procenjen spektar snage čistog govornog signala biti nenegativna vrednost. I u ovom slučaju, zavisno od algoritma, postoje različiti proračuni da bi se ovaj problem eliminisao.

Opšti oblik jednačine za oduzimanje spektara izgleda:

$$\left| \hat{X}(\omega) \right|^p = |Y(\omega)|^p - \left| \hat{D}(\omega) \right|^p \quad (10)$$

gde je  $p$  eksponent koji određuje tip spektra ( $p=1$  – amplitudski,  $p=2$  – spektar snage) a samim tim i varijaciju algoritma za oduzimanje spektara. Ova jednačina je i slikovito prikazana na slici 2.2.



Slika 2.2: Opšti oblik algoritma za oduzimanje spektara

### 2.2.2 Nedostaci algoritama za oduzimanje spektara

Iako je metoda oduzimanja spektara vrlo jednostavna za implementaciju i efektivno umanjuje šum u zašumljenom signalu, ona ima i neke nedostatke. Kako je napomenuto u prethodnom poglavlju, razlika između ukupnog i procenjenog spektra može biti negativna, te da bi se ovaj problem izbegao, ove vrednosti se izjednačavaju sa nulom. Ovakav metod ipak unosi i neke neželjene efekte. Naime, postavljanje negativne razlike između ukupnog i procenjenog

spektra na nulu ima za posledicu pojavu usamljenih vrhova na nasumičnim lokacijama unutar spektra. Kada se ovakav spektar pretvori u vremenski domen, posledica je zvuk koji zvuči kao nasumični tonovi, što je nepoželjan efekat. Ovakva pojava se u literaturi naziva muzička buka (*eng. musical noise*). U nekim slučajevima, muzička buka je čak neprijatnija nego šum koji je prisutan u ulaznom signalu. Kvalitet nekog algoritma za oduzimanje spektara se upravo ogleda u tome, u kojoj meri je uspešan u odstranjivanju šuma na predajnoj strani, a da pritom muzička buka bude što manja, pa ako je moguće i potpuno odsutna.

Drugi relativno manji nedostatak algoritama za oduzimanje spektara je direktna posledica korišćenja faze ukupnog zašumljenog signala. Iako na višim vrednostima ( $>5\text{dB}$ ) odnosa signal-šum (u daljem tekstu SNR – *eng. signal-to-noise ratio*) korišćenje faze ulaznog signala gotovo da nema uticaja na kvalitet rekonstruisanog signala, na manjim vrednostima SNR-a ( $<0\text{dB}$ ) korišćenje faze ulaznog signala ima za posledicu smanjenje kvaliteta rekonstruisanog signala. Ove vrednosti su proučavane u radovima isključivo eksperimentalno [3]. Procenjivanje faze čistog govornog signala pak nije lak zadatak, i doveo bi do daleko kompleksnijeg algoritma. Iz ovog razloga, praksa je da se kod većine ovakvih algoritama koristi faza ulaznog signala, koja ipak u većini slučajeva daje zadovoljavajuće rezultate.

### 2.2.3 Višeopsežni algoritam za oduzimanje spektara

Kako je u prethodnom poglavlju nagovešteno, kvalitet nekog algoritma za oduzimanje spektara se ogleda u tome, u kojoj meri je uspešan u odstranjivanju šuma na predajnoj strani, a da pritom muzička buka bude što manja. Višeopsežni algoritam za oduzimanje spektara [5] zasniva svoj rad na činjenici da, uglavnom, šum ne utiče na govorni signal u istoj meri čitavom širinom spektra. U zavisnosti od spektralnih osobina šuma, šum različito utiče na govorni signal unutar delova spektra.

SMNS algoritam je zasnovan na višeopsežnom algoritmu za oduzimanje spektara. U ovom algoritmu, ceo spektar se deli na više frekvencijskih podopsega, i to prozoriranjem u frekventnom domenu. Podela na podopsege se može vršiti i u vremenskom domenu korišćenjem filtara, ali je prozoriranje u frekventnom domenu daleko lakše i efikasnije za implementaciju. U svakom od ovako podeljenih podopsega se vrši oduzimanje spektara, što je, pokazalo se, dovelo do jako efikasnog umanjenja muzičke buke. Procenjeni spektar čistog govornog signala u i-tom opsegu dobija se kao:

$$\left| \hat{X}_i(\omega_k) \right|^2 = \left| \bar{Y}_i(\omega_k) \right|^2 - \alpha_i \cdot \delta_i \cdot \left| \hat{D}_i(\omega_k) \right|^2 \quad b_i \leq \omega_k \leq e_i \quad (11)$$

gde su  $\omega_k = 2\pi k/N$  ( $k = 0, 1, \dots, N-1$ ) diskretne frekvencije,  $\left| \hat{D}_i(\omega_k) \right|^2$  je procenjeni spektar snage šuma (izračunat u delovima signala bez prisustva govora),  $b_i$  i  $e_i$  su početna i krajnja frekvencija  $i$ -tog podopsega,  $\alpha_i$  je faktor koji određuje u kojoj meri će biti oduzet procenjeni spektar snage šuma, a  $\delta_i$  je dodatni faktor, koji omogućava podešavanje u kolikoj meri će biti oduzet procenjeni spektar snage šuma, i on se može podesiti za svaki podopseg ponaosob.  $\bar{Y}_i(\omega_k)$  predstavlja usrednjen spektar ulaznog signala u  $i$ -tom podopsegu. Negativne vrednosti razlike između spektra ulaznog signala i procenjenog spektra šuma se ne izjednačavaju sa nulom, već se računaju prema sledećoj formuli:

$$\left| \hat{X}_i(\omega_k) \right|^2 = \begin{cases} \left| \hat{X}_i(\omega_k) \right|^2 & , \left| \hat{X}_i(\omega_k) \right|^2 > \beta \left| \bar{Y}_i(\omega_k) \right|^2 \\ \beta \left| \bar{Y}_i(\omega_k) \right|^2 & , \left| \hat{X}_i(\omega_k) \right|^2 \leq \beta \left| \bar{Y}_i(\omega_k) \right|^2 \end{cases} \quad (12)$$

gde  $\beta = 0.002$ . Da bi se još umanjio uticaj muzičke buke, u ovako procenjeni spektar govora se vraća mala količina spektra ulaznog signala:

$$\left| \overline{\hat{X}}_i(\omega_k) \right|^2 = \left| \hat{X}_i(\omega_k) \right|^2 + 0.05 \cdot \left| \bar{Y}_i(\omega_k) \right|^2 \quad (13)$$

Faktor koji određuje u kojoj meri će biti oduzet procenjeni spektar snage šuma,  $\alpha_i$ , računa se na sledeći način:

$$\alpha_i = \begin{cases} 4.75 & SNR_i < -5 \\ 4 - \frac{3}{20}(SNR_i) & -5 \leq SNR_i \leq 20 \\ 1 & SNR_i > 20 \end{cases} \quad (14)$$

a SNR svakog podopsega se računa kao:

$$SNR_i(dB) = 10 \log_{10} \left( \frac{\sum_{\omega_k=b_i}^{e_i} \left| \bar{Y}_i(\omega_k) \right|^2}{\sum_{\omega_k=b_i}^{e_i} \left| \hat{D}_i(\omega_k) \right|^2} \right) \quad (15)$$

Dodatni faktor koji omogućava podešavanje u kolikoj meri će biti oduzet procenjeni spektar,  $\delta_i$ , dobijen je ispitivanjima, i dat je sledećim izrazom:

$$\delta_i = \begin{cases} 1 & f_i \leq 1kHz \\ 2.5 & 1kHz \leq f_i \leq \frac{Fs}{2} - 2kHz \\ 1.5 & f_i > \frac{Fs}{2} - 2kHz \end{cases} \quad (16)$$

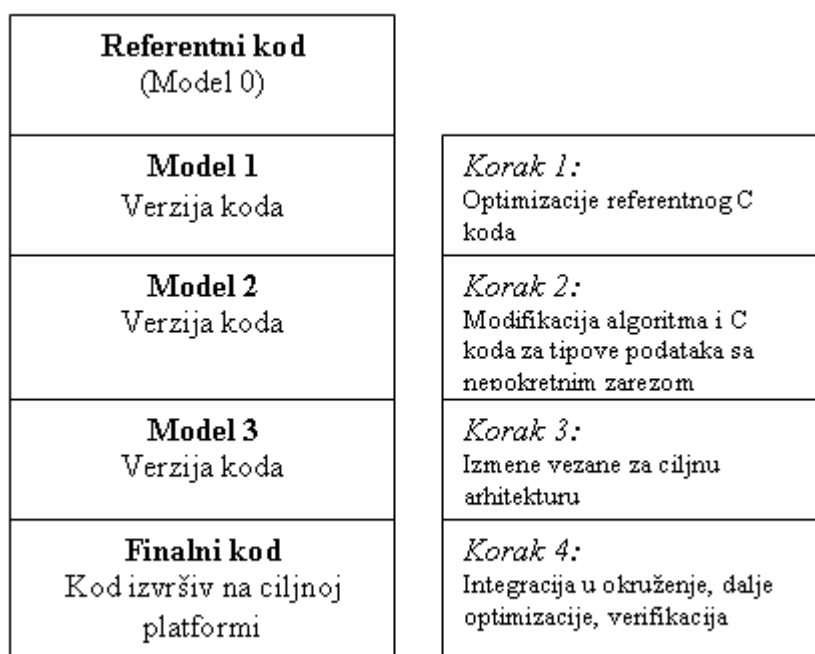
---

gde je  $f_i$  gornja frekvencija  $i$ -tog podopsega, a  $F_s$  frekvencija odabiranja.

Bitna napomena je da SMNS algoritam kao takav nije u potpunosti isti kao algoritam za višeopsežno oduzimanje spektara, ali princip rada je isti. Razlike se ogledaju u faktorima koji određuju u kojoj meri će biti oduzet procenjeni spektar šuma ( $\beta, \delta_i$ ) i oni su dobijeni eksperimentalno, prilikom razvoja samog algoritma. Kriterijum su bili naravno što bolji rezultati, a o merama koje se koriste prilikom razvoja i implementacije ovakvih algoritama će biti više reči u poglavlju ispitivanje i verifikacija.

### 3. Koncept rešenja

Prilikom razmatranja koncepta razvoja rešenja, a pritom razmatrajući brzinu i kvalitet izrade rešenja, odabrana je metodologija razvoja i implementacije algoritma [6] prikazana na slici 3.1. Ova metodologija predviđa korišćenje C kompajlera razvijenog na katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu, namenjenog Crystal DSP familiji procesora firme Cirrus Logic (CCC – Cirrus Logic C Compiler).



Slika 3.1: Tok implementacije algoritma

Ovakav pristup razvoja i implementacije algoritma predviđa niz koraka (modela), gde svaki predstavlja korak bliže ka krajnjem rešenju. Kako se razvoj algoritma vrši na računaru opšte namene (PC), prilikom razvoja algoritma nije bitno voditi računa o efikasnom iskorišćenju resursa, korišćenju aritmetike sa nepokretnim zarezom i sl., što je samo deo zahteva koji se stavljaju pred krajnje rešenje, implementirano na platformi sa ograničenim resursima, baziranoj na DSP (*eng. Digital Signal Processor*) tehnologiji. Prilikom razvoja ovakve vrste algoritama je najbitnije da algoritam obezbeđuje što bolje potiskivanje buke uz što manju degradaciju korisnog signala, odnosno signala govora. Tek u kasnijim fazama razvoja rešenja vodi se računa o efikasnom iskorišćenju resursa. Ovakva metodologija je razvijena na katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu, a iskustva u primeni ovakve metodologije na razvoj programske podrške namenjene audio procesorima firme Cirrus Logic pokazala su značajno brži razvoj krajnjeg rešenja, odnosno brže plasiranje proizvoda na tržište (*eng. time-to-market*).

### 3.1 Model 0

Model 0, odnosno referentni kod, predstavlja prvu verziju algoritma. U modelu 0, najbitnije je da algoritam zadovoljava postavljene kriterijume (ispravan rad algoritma, odnosno da je željeni efekat algoritma prisutan a da pritom taj efekat bude što je moguće kvalitetniji). U ovoj fazi nije bitno voditi računa o efikasnom iskorišćenju resursa, aritmetici sa nepokretnim zarezom itd. Model 0 se razvija na računaru opšte namene (PC), u programskom jeziku C (najčešće u programskom paketu Microsoft Visual Studio) ili programskom paketu MATLAB.

### 3.2 Model 1

Model 1 predstavlja model koji služi za pripremu za prelazak na aritmetiku sa nepokretnim zarezom. On podrazumeva optimizaciju C koda koji je napisan u aritmetici sa pokretnim zarezom. Ovakve optimizacije su vrlo bitne, pogotovo ako će se krajnji kod prevoditi kompajlerom (jer je i sama metodologija prvobitno osmišljena za korišćenje kompajlera za dobijanje koda izvršivog na ciljnoj platformi). Metodologija podrazumeva sledeće optimizacije:

- Zamena pristupanja elementima niza preko indeksa pristupanjem preko pokazivača
- Smanjivanje broja argumenata funkcije
- Redukovanje struktura

*Zamena pristupanja elementima niza preko indeksa pristupanjem preko pokazivača* je tip optimizacije čiji rezultat uglavnom zavisi od ciljne arhitekture i kvaliteta kompajlera ciljne platforme. Neke procene ukazuju na to da ovaj tip optimizacije donosi od 10% do 30%

smanjenje broja linija koda, a u nekim slučajevima korišćenjem CCC prevodioca pokazalo se čak i do 60% poboljšanja kod brzine izvršavanja koda.

**Smanjivanje broja argumenata funkcije** sastoji se u izbacivanju što većeg broja parametara funkcija na globalni nivo, u vidu promenljivih ili konstanti. Ovakav vid optimizacije takođe doprinosi poboljšanju kvaliteta koda, odnosno smanjenju broja linija, no ne u tolikoj meri koliko prethodno navedeni tip optimizacije.

**Redukovanje struktura** predstavlja vid optimizacije, gde se, slično kao i kod parametara funkcije, a iz faktički istog razloga, i strukture stavljaju na globalni nivo. Naime, neretko se u referentnom kodu funkciji prosleđuje struktura, i to najčešće preko njegovog pokazivača. Prevođenjem ovakvog koda korišćenjem CCC prevodioca dobija se neefikasni kod na ciljnoj platformi, te je praksa da se strukture postavljaju na globalni nivo, a elementima strukture se pristupa direktno unutar funkcija.

### 3.3 Model 2

Model 2 predstavlja prelazak sa aritmetike sa pokretnim na aritmetiku sa nepokretnim zarezom. Pošto neki DSP procesori ne poseduju jedinicu za rad sa tipovima podataka sa pokretnim zarezom (FPU – eng. *Floating Point Unit*), za ovakve DSP procesore potrebno je napisati C kod koji umesto tipova podataka sa pokretnim zarezom koristi tipove podataka sa nepokretnim zarezom. Neki prevodioci namenjeni DSP platformama očekuju da je kod u nepokretnom zarezu napisan korišćenjem celobrojnih tipova podataka, međutim, ovako napisan kod može dovesti do generisanja vrlo neoptimalnog koda za dati DSP procesor. Iz ovog razloga, na katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu, razvijene su posebne klase za emulaciju aritmetike sa nepokretnim zarezom, koje emuliraju tipove podataka prisutne na Crystal familiji procesora, firme Cirrus Logic. Ovakvim pristupom moguće je razvijati algoritam namenjen izvršenju na DSP platformi na računaru opšte namene (PC), korišćenjem programskog paketa Microsoft Visual Studio. Ovo znatno olakšava i ubrzava razvoj algoritma, doprinosi lakše uočavanje i otklanjanje grešaka, i to iz prostog razloga što su eliminisani faktori koji su vezani za implementaciju na ciljnoj platformi (konfiguracija ciljne platforme, snimanje izlaza, poravnavanje izlaza, bit-identično poređenje itd.).

### 3.4 Model 3

Model 3 predstavlja korak u kojem se C kod iz modela 2 prevodi u asemblerski kod ciljne platforme. Faktički, kod generisan u ovom modelu je u najvećem broju slučajeva izvršiv na ciljnoj platformi. Jedini slučaj u kojem kod generisan u ovom modelu nije izvršiv na ciljnoj platformi je u slučaju ako kod nije dovoljno prilagođen, odnosno prevazilazi ograničenja ciljne

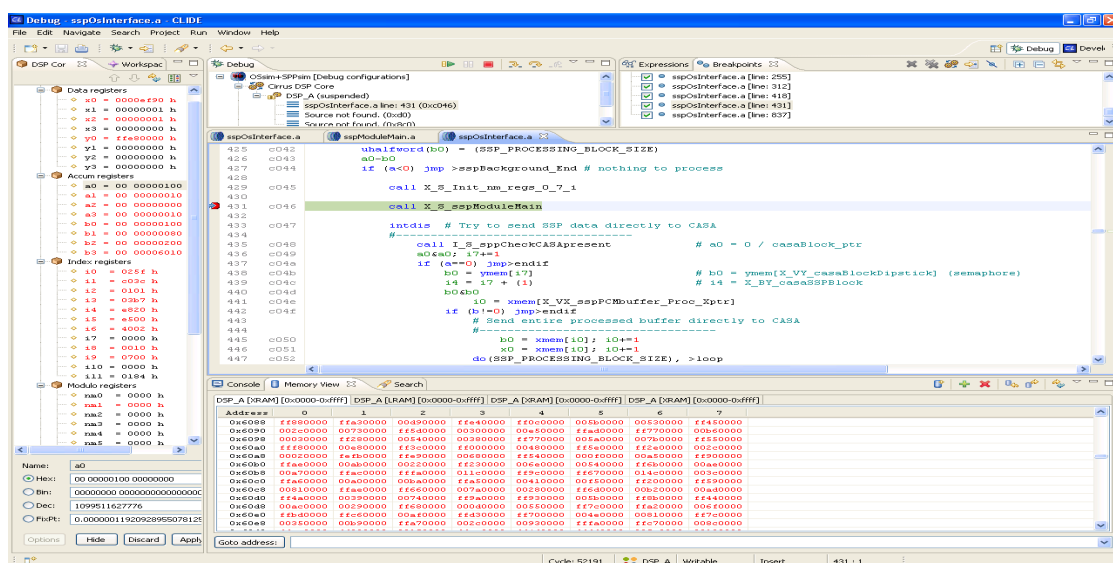
arhitekture (bilo u potrošnji memorije, što bi rezultovalo u neuspehom generisanju koda, bilo u potrošnji procesorskog vremena, što bi najčešće rezultovalo u pojavi smetnji u signalu). Dakle, samo uspešnim prevođenjem C koda iz modela 2 u asemblerski kod ciljne platforme dobija se kod koji je u potpunosti izvršiv na procesorima iz Crystal DSP familije firme Cirrus Logic.

### 3.4.1 Razvojno okruženje CLIDE

Na katedri za računarsku tehniku i računarske komunikacije na Fakultetu tehničkih nauka u Novom Sadu, razvijen je alat koji objedinjava kompajler (*eng. compiler*), povezivač (*eng. linker*), assembler i okruženje za nalaženje i otklanjanje grešaka, namenjen Crystal DSP familiji procesora firme Cirrus Logic. CLIDE (Cirrus Logic Integrated Development Environment) predstavlja alat, odnosno integrisano razvojno okruženje, realizovan na bazi Eclipse platforme, i kao takav poseduje sve potrebne funkcije bitne pri razvoju i otklanjanju grešaka, kao što su:

- kontrolisano izvršenje koda
- prikaz memorije
- prikaz registara
- praćenje vrednosti promenljivih (watch)
- prikaz datoteka koje sadrže izvorni kod
- tekstualni editor za pisanje programa

i mnoge druge. CLIDE takođe sadrži opciju da se kod izvršava u simulatorskom modu, odnosno u modu izvršenja na ciljnoj platformi. Simulatorski mod je vrlo koristan kod razvoja algoritama, jer ne obavezuje korišćenje ciljne platforme, a samim tim ni korišćenje mehanizama koje rad u realnom vremenu zahteva. Mod izvršenja na ciljnoj platformi posebno je koristan u procesu uočavanja i otklanjanja grešaka na ciljnoj platformi, ali isto tako i tokom razvoja. Prikaz jednog prozora CLIDE okruženja dat je na slici 3.2.



Slika 3.2: Razvojno okruženje CLIDE

### **3.5 Finalna verzija koda**

Kod generisan u modelu 3 je u potpunosti prilagođen izvršenju na ciljnoj platformi. Preostali zadaci koji kompletiraju dato rešenje su integracija u okruženje pritom poštujući konvencije koje data platforma zahteva. Takođe, u ovoj iteraciji se po potrebi mogu vršiti optimizacije koda. Optimizacije se vrše ukoliko dato rešenje troši preveliku količinu resursa, što radi ispravnog rada, što zbog oslobađanja resursa za neki drugi modul koji eventualno treba da radi u isto vreme na ciljnoj platformi sa datim algoritmom. Nakon integracije u okruženje i eventualne optimizacije koda, kod je u potpunosti izvršiv na ciljnoj platformi, te je spreman za ispitivanje i verifikaciju.

## 4. Implementacija

U ovom poglavlju izneti su detalji implementacije datog rešenja na ciljnoj platformi, iskorišćenost resursa, kao i neke tehničke osobine ciljne platforme na kojoj je dati algoritam implementiran.

### 4.1 Detalji implementacije

Kako je u poglavlju 3 izneto, implementacija je zasnovana na model-orijentisanoj metodologiji. Kako je za dato krajnje rešenje odabrano pisanje koda u asemblerskom jeziku (umesto korišćenja prevodioca), implementacija ovog rešenja odstupa od date metodologije u nekim nijansama. Pisanje izvršnog koda u asemblerskom jeziku je odabrano zbog efikasnosti koda, iako i sam kompajler ima dobre rezultate u efikasnosti prevedenog koda. Ručno napisani asemblerski kod ipak daje najbolje performanse ukoliko se poznaje instrukcijski set procesora i tehnike optimizacije, ali za razvoj asemblerskih kodova uglavnom treba više vremena nego pisanjem u jezicima višeg nivoa, u ovom slučaju u programskom jeziku C.

#### 4.1.1 Model 0

Model 0, kako je ranije u poglavlju 3 napomenuto, predstavlja referentni kod, koji je polazna osnova za implementaciju datog algoritma na ciljnoj platformi. Referentni kod razvijen je na katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu, koristeći za osnovu višeopsežni algoritam za oduzimanje spektara, i ovaj algoritam je korišćen kao polazna osnova u okviru ovog rada. Kod je implementiran u programskom jeziku C i predstavlja dobru polaznu osnovu za naredne modele (jer je referentni algoritam često realizovan u programskom paketu MATLAB, te je u tom slučaju prelazak na model 1 malo komplikovaniji).

### 4.1.2 Model 1

Kako je izneto u poglavlju 3.2, model 1 predstavlja iteraciju u kojoj se vrše neke opšte optimizacije, kao i optimizacije vezane za ciljnu platformu. Kako je odabrana realizacija pisanjem asemblerskog jezika (zbog optimalne potrošnje resursa ciljne platforme), optimizacije navedene u poglavlju 3.2 nisu implementirane u modelu 1. Ipak, u modelu 1 je urađeno profilisanje koda, koje se sastoji u određivanju opsega svih rezultata i međurezultata unutar algoritma. Dodatno, kako model 2 predstavlja implementaciju koda u nepokretnom zarezu [7], što podrazumeva da svi rezultati i međurezultati moraju biti u opsegu od -1 (odnosno 0x80000000, što je najmanji negativni broj koji je moguće predstaviti u 32 bita u aritmetici nepokretnog zareza) do približno 1 (odnosno 0x7FFFFFFF, što je najveći pozitivni broj koji je moguće predstaviti u 32 bita u aritmetici nepokretnog zareza), kod modela 1 je prilagođen da sve rezultate koristi u ovom opsegu, naravno, koristeći i dalje tipove podataka sa pokretnim zarezom. Ovim je znatno olakšana implementacija modela 2, a naročito otkrivanje i otklanjanje grešaka.

### 4.1.3 Model 2

Kako je ranije objašnjeno u poglavlju 3.3, model 2 predstavlja prelazak sa aritmetike pokretnog na aritmetiku nepokretnog zareza. Korišćenjem biblioteka za emulaciju aritmetike nepokretnog zareza za Crystal familiju DSP procesora, kod iz modela 1 prepravljen je da koristi aritmetiku nepokretnog zareza. Biblioteke za emulaciju aritmetike nepokretnog zareza za ciljnu platformu poseduju sve potrebne tipove podataka, koje emuliraju registre i akumulatore datih procesora, kao i mehanizme kao što su jedinica za logičko pomeranje, zaokruživanje i saturaciju (*SRS* modul date familije procesora, *Shift Round Saturate* – više u poglavlju 4.2) i dr. Kao olakšavajuća okolnost prilikom implementacije, kao i pronalaženja i otklanjanja grešaka u modelu 2, poslužila je činjenica da je kod u modelu 1 namešten da svi rezultati i međurezultati budu u opsegu od -1 do skoro 1.

### 4.1.4 Model 3 – kod izvršiv na ciljnoj platformi

Za razliku od predložene metodologije, s obzirom da je odabrana implementacija algoritma na ciljnoj platformi ručnim pisanjem asemblerskog koda, u ovom radu poslednja dva koraka sa slike 3.1 su objedinjena u jedan. Asemblerski kod je pisan direktno u radno okruženje, pri čemu je prilikom razvoja algoritma korišćena opcija CLIDE okruženja da radi u simulatorskom režimu, kao i specijalno razvijen *os\_simulator* alat (koji je razvijen u okviru katedre za računarsku tehniku i računarske komunikacije) koji simulira ponašanje systemske programske podrške date familije DSP procesora (Crystal familija firme Cirrus Logic). Najpre je urađena implementacija algoritma gde je bio akcenat na ispravnom radu algoritma, a verifikovan ispravan

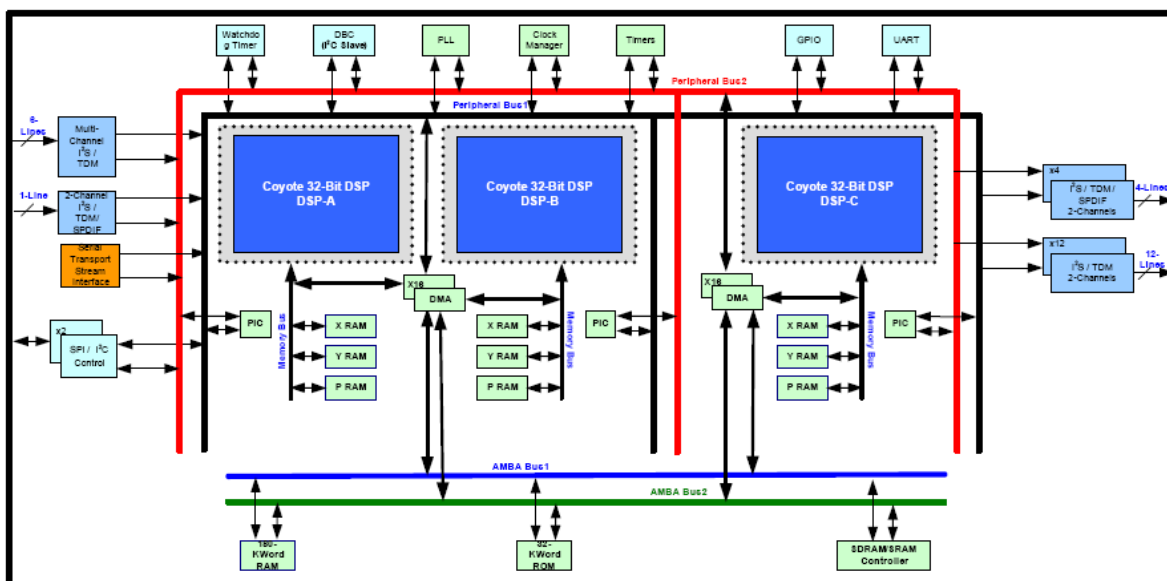
kod je zatim podvrgnut optimizacijama, vezanim za datu DSP platformu i instrukcijski set date familije DSP procesora. Iskorišćenost resursa na ciljnoj platformi data je u poglavlju 4.3.

## 4.2 Tehničke specifikacije ciljne platforme

Ciljni procesor, CS49834 [8] je najmoćniji digitalni signal procesor iz Crystal DSP familije procesora, firme Cirrus Logic. On predstavlja trojezgarni digitalni signal procesor, zasnovan na unapređenoj Harvard [9] arhitekturi procesora, gde je postoje dva odvojena tipa memorije, memorija za podatke i programska memorija, namenjena za smeštanje programskog koda. Dodatno, memorija za podatke je odvojena u dva segmenta, tzv. “X” i “Y” segment. Ovakva organizacija omogućava veliku iskorišćenost resursa datog procesora i do čak šest instrukcija u jednom taktu – dve MAC (*eng. Multiply and Accumulate* – pomnoži i saberi) instrukcije, dva čitanja ili upisa iz/u memoriju i dva povećanja/umanjenja adresnih registara. Svako jezgro datog čipa radi na taktu od 300 MHz. Takođe, svako jezgro sadrži osam 32-bitnih registara opšte namene (x0-x3 i y0-y3), osam 72-bitnih registara (tzv. akumulatora, a0-a3 i b0-b3), namenjenih za smeštanje međurezultata, dvostruku preciznost i dr. Svako jezgro takođe sadrži dvanaest tzv. indeksnih registara (i0-i11), veličine 16 bita, namenjenih za adresiranje kod čitanja i upisa iz/u memoriju, kao i za druge namene, kao što su brojači iteracija petlji i slično. Svakom od indeksnih registara odgovara i tzv. modulo registar, koji omogućava da odgovarajući registar radi u modulo režimu (korisno kod kružnih bafera) kao i bit-reverznom režimu (vrlo korisno kod algoritama koji zahtevaju izuzetno brzo izvršenje, kao što su FFT algoritmi i dr.). Svako od jezgara sadrži tzv. SRS jedinicu (*eng. Shift Round Saturate*) koja upravlja aritmetičkim pomeranjem, zaokruživanjem i saturacijom između akumulatora i registara, kako joj i samo ime kaže. Dati čip sadrži i jedinicu za direktni pristup memoriji, DMA (*eng. Direct Memory Access*), koji služi kako za prenos podataka između jezgara, tako i za prenos podataka između spoljne memorije i svakog od tri jezgra. Povezivanje sa periferijama omogućeno je preko SPDIF (*eng. Sony Philips Digital InterFace*), HDMI (*eng. High Definition Multimedia Interface*) kao i analognih ulaza/izlaza. Količina raspoložive memorije je prikazana u tabeli 4.1, dok je blok dijagram CS49834 čipa dat na slici 4.1.

Tip memorije	DSP A	DSP B	DSP C	Ostala memorija
X	60k SRAM	60k SRAM	60k SRAM	180k SRAM
Y	60k SRAM	60k SRAM	60k SRAM	
P	60k SRAM	60k SRAM	60k SRAM	

Tabela 4.1: Količina raspoložive memorije unutar CS49834 čipa



Slika 4.1: Blok dijagram CS49834 čipa

### 4.3 Iskorišćenost resursa datog rešenja na ciljnoj platformi

U ovom delu poglavlja iznete su vrednosti iskorišćenosti memorije datog DSP procesora, kao i utrošenog procesorskog vremena. Iskorišćenost memorije se obično izražava u količini zauzetih “reči”, što u suštini predstavlja broj zauzetih 32-bitnih memorijskih lokacija. Iskorišćenost procesorskog vremena se obično meri u MIPS-ima, odnosno milionima instrukcija u sekundi (*eng. Million Instructions Per Second*). Brojanjem utrošenih vremenskih ciklusa (taktova) procesora, računanje potrošnje procesorskog vremena vrši se datom formulom:

$$MIPS = \frac{nCiklusa \cdot \frac{Fs}{BLOCK\_SIZE}}{1000000} \quad (17)$$

gde  $Fs$  predstavlja frekvenciju odabiranja, u ovom slučaju 16kHz, a  $BLOCK\_SIZE$  predstavlja veličinu bloka obrade, koji je u ovom rešenju 128 odbiraka.

U tabeli 4.2 je data potrošnja resursa date platforme, memorije podataka i memorije za kod, kao i maksimalna potrošnja procesorskog vremena. U tabeli 4.3 data je detaljna analiza potrošnje procesorskog vremena za dato rešenje.

XMEM	YMEM	LMEM	PMEM	MIPS
1105	1107	1922	1267	11.128

Tabela 4.2: Iskorišćenost resursa datog rešenja

Funkcija	Minimalna vrednost broja ciklusa	Srednja vrednost broja ciklusa	Maksimalna vrednost broja ciklusa	Minimalna vrednost MIPS	Srednja vrednost MIPS	Maksimalna vrednost MIPS
X_S_sspModuleMain	0000bf35	0001495d	00015bc2	6.119	10.540	11.128
X_S_nslibFFT	00001cb0	00001cb5	00001cba	0.918	0.919	0.919
X_S_sspProcess	00006b05	0000f52e	00010793	3.425	7.846	8.434
X_S_nslibDivide	00001b00	000062f6	000068d4	0.864	3.167	3.355
X_S_nslibLog10	00000000	00002281	0000229f	0.000	1.104	1.108
X_S_nslibIFFT	00002b85	00002b59	00002b5d	1.393	1.387	1.388

Tabela 4.3: Detaljna analiza potrošnje procesorskog vremena

U tabeli 4.3 dat je prikaz potrošnje procesorskog vremena svake od funkcija unutar koda. **X\_S\_sspModuleMain** funkcija predstavlja glavnu funkciju ovog algoritma i ona ima maksimalnu potrošnju od 11.128 MIPS. Ova funkcija poziva tri funkcije, i to **X\_S\_nslibFFT** koja vrši FFT transformaciju, **X\_S\_sspProcess** koja vrši oduzimanje spektara, kao i **X\_S\_nslibIFFT** koja vrši inverznu FFT transformaciju. Potrošnja procesorskog vremena za svaku od ovih funkcija je takođe data tabelarno. Dodatno, **X\_S\_sspProcess** funkcija poziva jednom **X\_S\_nslibLog10** funkciju i četiri puta **X\_S\_nslibDivide** funkciju, i ukupna potrošnja ovih funkcija data je takođe unutar tabele. Razlika između ukupnog broja i zbira pojedinačnih potrošnja procesorskog vremena predstavlja potrošnju ostalog koda (oduzimanja spektara, prozoriranja itd.).

## 5. Ispitivanje i verifikacija

Prilikom razvoja algoritma, kao i u svakoj fazi implementacije rešenja, bitno je pratiti efikasnost algoritma za potiskivanje smetnji. U slučaju polaznog (referentnog) algoritma, praćenje efikasnosti je veoma bitno da bi se odabrala verzija algoritma koja je najefikasnija. Kada se odredi najbolja verzija algoritma, koja zadovoljava zadate kriterijume (u konkretnom slučaju što bolje potiskivanje buke uz što manju degradaciju korisnog signala), pristupa se razvoju ostalih faza algoritma, kako je opširnije opisano u prethodnim poglavljima. Vrlo je bitno da se u svakoj iteraciji prati efikasnost, da ne bi došlo do degradacije efikasnosti algoritma. Mere, u kojima se ogleda efikasnost ovakvih vrsta algoritama su kvalitet i razumljivost. Kvalitet i razumljivost govornog signala nisu jednaki, iako utiču jedan na drugog. U kvalitet spadaju prirodnost govora, koliko šum smeta pri slušanju i sl. Razumljivost predstavlja meru, u kojoj je moguće razumeti signal govora. Na primer, veštački generisani govorni signal, koji se sastoji moduliranih sinusoida i šumova, većini slušalaca će zvučati veoma neprirodno, iako njegova razumljivost može biti veoma visoka. S druge strane, u govornoj komunikaciji preko interneta (VoIP) često dolazi do gubljenja paketa, što znatno smanjuje razumljivost govora usled gubljenja pojedinih slogova ili celih reči, dok sam govorni signal može biti vrlo visokog kvaliteta.

Mere kojima se ocenjuje efikasnost ovakvih vrsta algoritama mogu se podeliti u dve osnovne grupe:

1. subjektivne metode merenja
2. objektivne metode merenja

### 5.1 Subjektivne metode merenja

U subjektivne metode merenja spada slušno ispitivanje ispitnih vektora. Ispitni vektori se puštaju određenom broju ispitanika, koji ocenjuju kvalitet sekvenci na skali od 1-5, gde 1

predstavlja najgoru ocenu, a 5 predstavlja najbolju ocenu. Tačno značenje ocena prikazano je u tabeli 5.1.

Ocena	Nivo kvaliteta
1	Loš
2	Slab
3	Zadovoljavajući
4	Dobar
5	Odličan

Tabela 5.1: Skala subjektivne ocene kvaliteta govornog signala

Pre sprovođenja subjektivnog ispitivanja bitna je faza obuke osoba koje vrše ispitivanje. Obuka se sastoji iz predstavljanja niza referentnih signala, od kojih svaki odgovara nekoj od ocena iz tabele 5.1. Na ovaj način ispitanici stiču osećaj i bivaju obučeni za ispitivanje algoritma za otklanjanje smetnji u govornom signalu. Sam postupak obuke slušalaca je definisan standardom (ITU-R BS.562-3), i definiše broj slušalaca, proceduru za sprovođenje kao i trajanje ispitivanja, kao i izbor opreme za reprodukciju. Nakon izvršenog ispitivanja, računa se srednja vrednost zbira svih ocena, odnosno MOS (*eng. Mean opinion score* – srednja vrednost mišljenja). Iako je ovakav vid ispitivanja prilično pouzdan, on sam po sebi nije praktičan za čestu upotrebu, npr. kod razvoja algoritama, jer zahteva dosta vremena, kao i dosta ljudskih resursa. Iz ovakvog razloga, ovakvo ispitivanje nije sprovedeno u okviru ovog rada.

Drugu vrstu subjektivnog ispitivanja predstavlja ispitivanje razumljivosti (*eng. intelligibility*). Kao i MOS ispiti, i ona uključuje slušno ispitivanje, ali je razlika u ispitnim vektorima. Ispitni vektori kod ispitivanja razumljivosti mogu da budu niz reči kao i rečenice. U prvom slučaju, reči nemaju nikakvog značenja i sastavljene su od dva sloga. Ovakve reči se još nazivaju i logatomima. Svaki logatom, odnosno dvosložna reč bez ikakvog značenja, sastavljena je od dva sloga, gde svaki slog sadrži jedan samoglasnik i jedan suglasnik. Iako logatomi koji se koriste u srpskom jeziku nisu definisani standardom, postoji skup logatoma koji se često koristi. Ovakav skup logatoma sastavljen je od strane laboratorije za akustiku Elektrotehničkog fakulteta u Beogradu i sastoji se od 500 logatoma podeljenih u 10 grupa po 50 logatoma. Korišćenjem ovog skupa logatoma rezultati ispitivanja su merodavni, jer je skup reči koje se koriste u ispitivanju uvek isti.

Kod ispitivanja razumljivosti koje uključuje rečenice, sličan princip postoji kao i kod logatoma. Naime, kod logatoma same reči nemaju nikakvog značenja, dok u slučaju rečenica, rečenice same po sebi nemaju nikakvog smisla. Svaka ponaosob reč unutar rečenice ima svoje značenje, ali rečenice su sastavljene od reči koje jedna sa drugom ne daju nikakav smisao.

Ovakav koncept doprinosi verodostojnosti rezultata ispitivanja, jer eliminiše mogućnost da ispitanik nasluti narednu reč iz konteksta ili da nauči napamet rečenice.

Ovakav vid ispitivanja daje još merodavnije rezultate od MOS ispitivanja, ali kao i kod MOS ispitivanja, ono zahteva vreme i dosta ljudskih resursa. Iz tog razloga ni ovakvo ispitivanje nije sprovedeno u okviru ovog rada.

## 5.2 Objektivne metode merenja

Kako subjektivno ispitivanje daje verodostojne rezultate, ali zahteva vreme i veliku količinu ljudskih resursa, postojala je potreba da se osmisli metod ispitivanja koji bi davao rezultate slične verodostojnosti kao i subjektivne metode ispitivanja, ali uz što manje angažovanje ljudskih resursa, pa čak i eliminaciju ljudskog faktora u potpunosti. Kao alternativa subjektivnim metodama ispitivanja, početkom devedesetih godina prošlog veka pojavila se PSQM mera (skraćeno od *Perceptual Speech Quality Measure*). Ipak, ubrzo se pokazalo da ova mera nije davala dobre rezultate za neke ispitne slučajeve, te je ona ubrzo zamenjena sa PESQ [10] merom (*eng. Perceptual Evaluation of Speech Quality*), koja je ove nedostatke ispravila. Za računanje PESQ mere potrebno je najpre vremenski poravnati dva signala, pri čemu je jedan od signala originalni signal a drugi je obrađeni signal. Zatim se ovako vremenski poravnati signali prevode iz vremenskog u perceptualni domen, koji odgovara načinu na koji ljudsko uho predstavlja zvučne signale. Na kraju, vrši se poređenje dveju perceptualnih domena, pri čemu se meri tzv. “simetrično” i “asimetrično” odstupanje. Simetrično odstupanje jednako vrednuje sve vrednosti odstupanja, dok asimetrično pozitivne vrednosti odstupanja posmatra kao pojavu šuma u obrađenom signalu, dok negativnu vrednost odstupanja posmatra kao degradaciju signala. Mere odstupanja usrednjavaju se po vremenu i frekvenciji, a zatim se računa krajnja PESQ ocena prema sledećoj formuli:

$$PESQ = 4.5 - 0.1 \cdot d_{sym} - 0.0309 \cdot d_{asym} \quad (18)$$

Druga često korišćena objektivna mera kod implementacije ovakvih tipova algoritama je mera poboljšanja odnosa signal-šum, odnosno SNRE (*eng. Signal-to-Noise Ratio Enhancement*). Ona se računa kao razlika procenjenog SNR-a između obrađenog i neobrađenog signala, što odgovara odnosu snage odgovarajućih signala smetnje:

$$SNRE = 10 \log_{10} \frac{E(X_{S,out}^2)}{E(X_{N,out}^2)} - 10 \log_{10} \frac{E(X_{S,in}^2)}{E(X_{N,in}^2)} = 10 \log_{10} \frac{E(X_{N,in}^2)}{E(X_{N,out}^2)} \quad (19)$$

Pošto tokom govorne aktivnosti nije moguće pouzdano odrediti snagu signala smetnje, procena SNRE mere se vrši tokom odsustva govorne aktivnosti.

Ovakve dve prethodno opisane mere za objektivno ispitivanje su pogodne za višestruku primenu jer eliminišu ljudski faktor i daju merodavne rezultate. Shodno tome, ove dve mere su

korišćene prilikom izrade ovog rada za verifikaciju kvaliteta svake faze implementacije algoritma, a rezultati su dati u poglavlju 5.5.

### 5.3 Opis ispitnih vektora

Prilikom izrade ovog rada korišćena su 32 ispitna vektora. Kao i metode ispitivanja, i organizacija ispitnih vektora koji se koriste u ispitivanju i verifikaciji algoritama za potiskivanje smetnji u govoru definisana je standardom. Ispitne sekvence traju osam sekundi, unutar kojih su dve rečenice na engleskom jeziku, sa pauzom između rečenica. U ovom radu korišćena su dva tipa ispitnih vektora, pri čemu jednu izgovara muški a jednu ženski govornik. U svakom ispitnom vektoru su po dve rečenice na engleskom jeziku, gde rečenice same po sebi nemaju nikakvog smisla. Ovakva dva ispitna vektora su zatim kombinovana sa osam različitih signala smetnje, i to u dva različita odnosa SNR-a, 6dB i 12dB. Signali smetnje su takođe prepisani standardom i oni simuliraju različite situacije u kojim se jedan sistem za potiskivanje akustičkih smetnji može naći. Svi tipovi smetnje sa kojima su čisti signali govora kombinovani dati su u tabeli 5.2.

Redni broj	Ime datoteke	Tip smetnje
1	jackhammer.wav	Pneumatski čekić
2	outside_traffic.wav	Buka ulice
3	pub_noise.wav	Veći broj istovremenih razgovora
4	inside_train.wav	Unutrašnjost voza
5	midsize_car.wav	Unutrašnjost automobila
6	1812.wav	Muzika
7	single_voice.wav	Drugi govornik
8	chord_pause.wav	Multiton signal sa pauzama

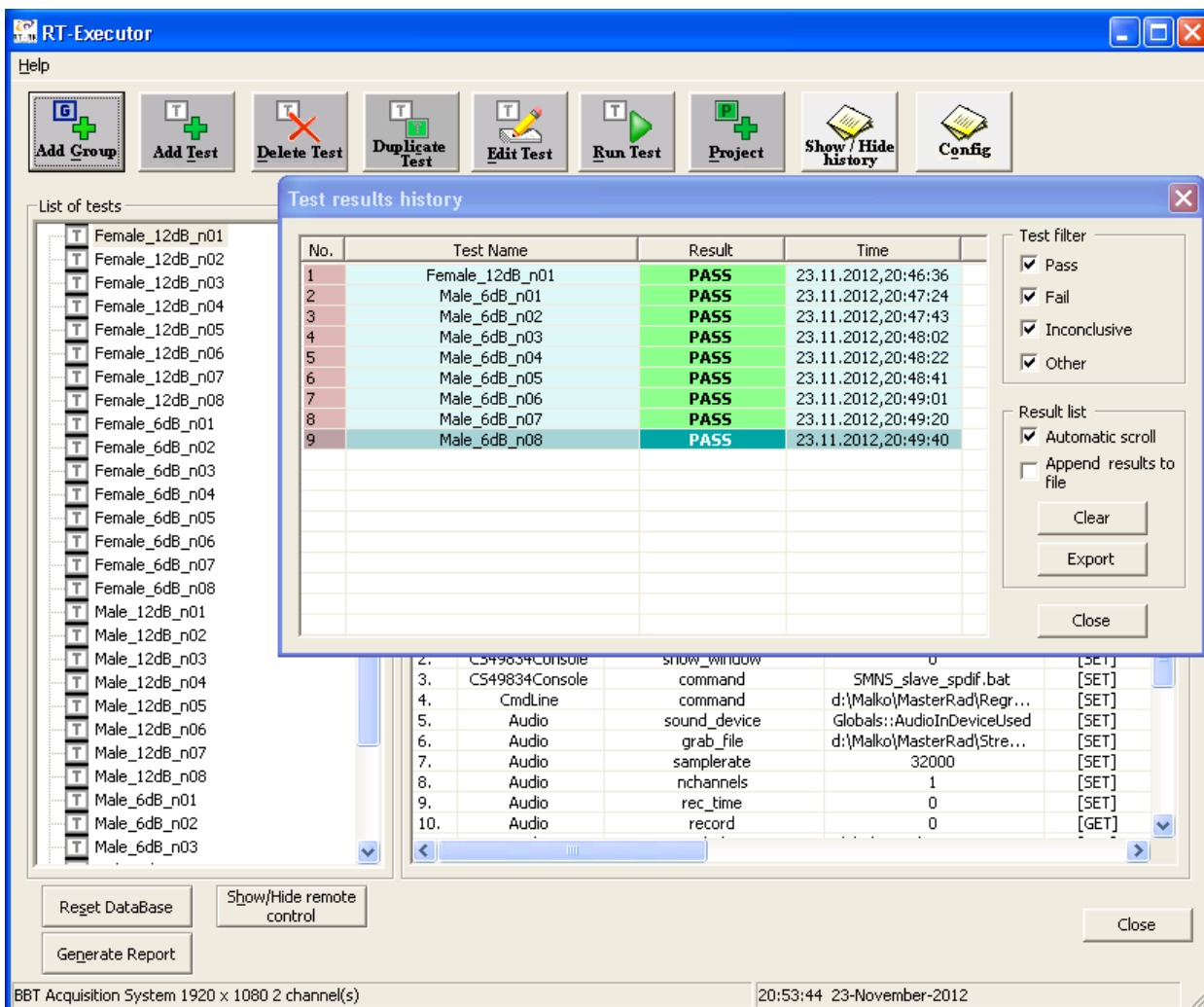
Tabela 5.2: Tipovi smetnje korišćeni pri kombinaciji sa čistim signalom govora

Svi ispitni vektori su odabirani frekvencijom odabiranja od 16kHz.

### 5.4 Bit-identična ispitivanja

Iako bit-identična ispitivanja ne nose informaciju o kvalitetu algoritma, vrlo su korisna u prelascima između dveju faza. Naime, sama model-orijentisana metodologija razvoja audio algoritama pretpostavlja bit-identičnost između modela 0 i 1, kao i između modela 2 i 3. Na prelasku između modela 1 i 2 nije jednostavno postići bit-identičnost, iz prostog razloga što se u toj fazi vrši prelazak sa aritmetike pokretnog zarezna na aritmetiku nepokretnog zarezna. Ovakva ispitivanja su dakle jako korisna, jer mogu vrlo lako da ukažu na grešku, ukoliko je bitska razlika

van okvira predviđenih model-orijentisanom metodologijom. Bit-identično ispitivanje može se izvršavati batch skriptama, kao i korišćenjem posebnog alata, BBT [11] (Black Box Testing – ispitivanje metodom crne kutije). Izgled ovog alata prikazan je na slici 5.1.



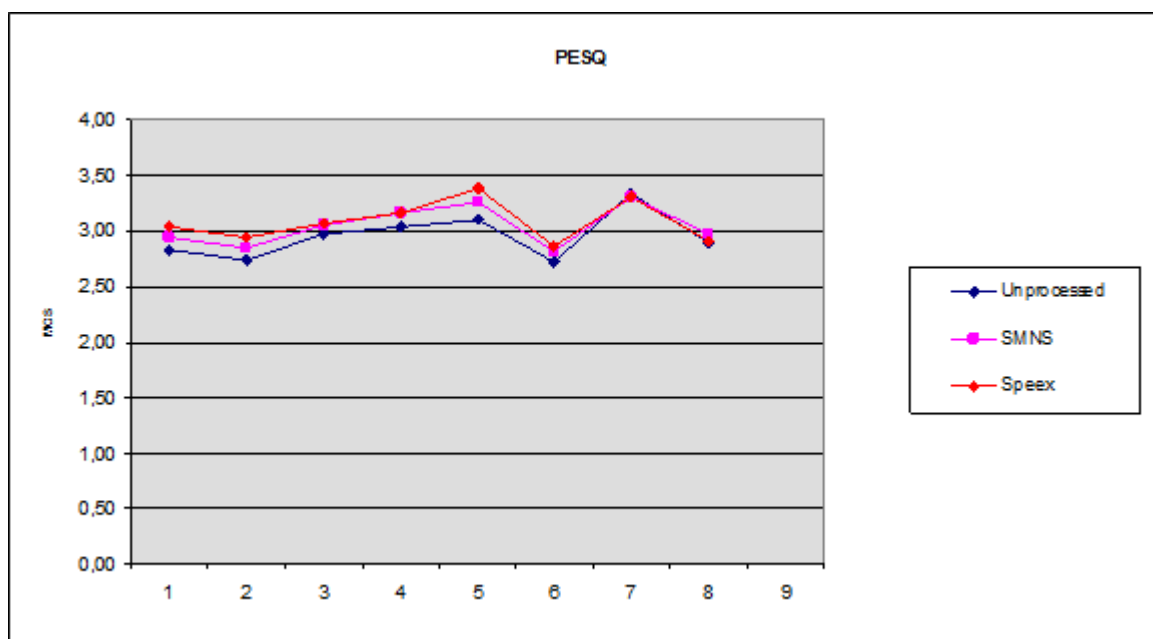
Slika 5.1: Prikaz BBT alata

BBT alat je razvijen na katedri za računarsku tehniku i računarske komunikacije Fakulteta tehničkih nauka u Novom Sadu i predstavlja moćan alat za ispitivanje i verifikaciju. Generalno rečeno, BBT alat je daleko naprednija verzija batch skripti. Prednosti BBT alata nad batch skriptama su brojne, od relativno jednostavne sintakse za pisanje ispitnih slučajeva (a samim tim i razvoj istih je relativno brz), do lakog korišćenja, prenosivosti, mnoštva ugrađenih alata koji pojednostavljaju proces ispitivanja, sve do preglednih izveštaja o rezultatima. Ovaj alat dakle omogućava brz razvoj ispitnih slučajeva, a ugrađeni alati omogućavaju izvršavanje širokog spektra operacija. Standardna procedura za ispitivanje izlaza sa ciljne platforme (konfiguracija, snimanje izlaza, vremensko poravnanje i bit-identično poređenje), korišćenjem BBT alata, prilično je jednostavna i jednom razvijeni ispitni slučajevi mogu se ponoviti neograničeni broj puta, kao i na proizvoljnom računaru (što je ponekad problem kod običnih batch skripti).

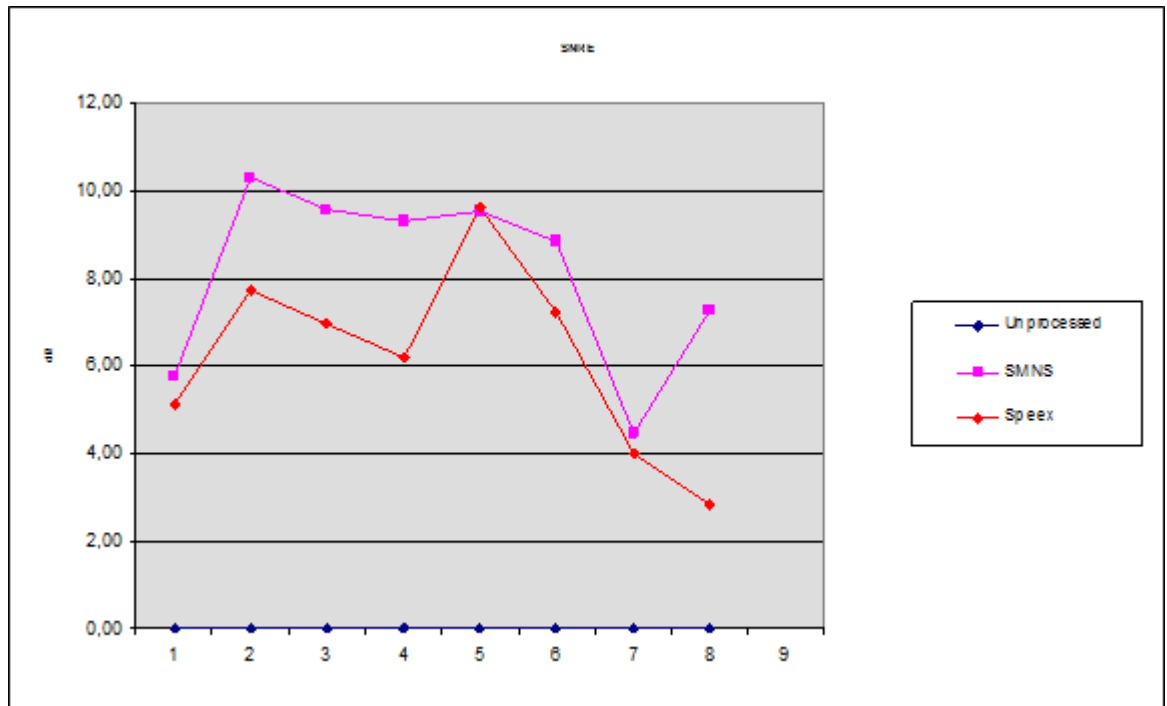
U okviru ovog rada urađeno je ispitivanje korišćenjem BBT alata, a dobijeni rezultati odgovaraju zahtevima koje predstavlja model-orijentisana metodologija. Dakle, za ceo skup ispitnih vektora, izlazi modela 0 i modela 1 su međusobno bit-identični, kao i izlazi modela 2 i modela 3. Razlika između izlaza modela 1 i modela 2 je 1 bit, što je prihvatljivo, obzirom da se prilikom prelaska iz modela 1 u model 2 vrši prebacivanje aritmetike pokretnog zarez u aritmetiku nepokretnog zarez.

## 5.5 Rezultati ispitivanja objektivnim metodama merenja

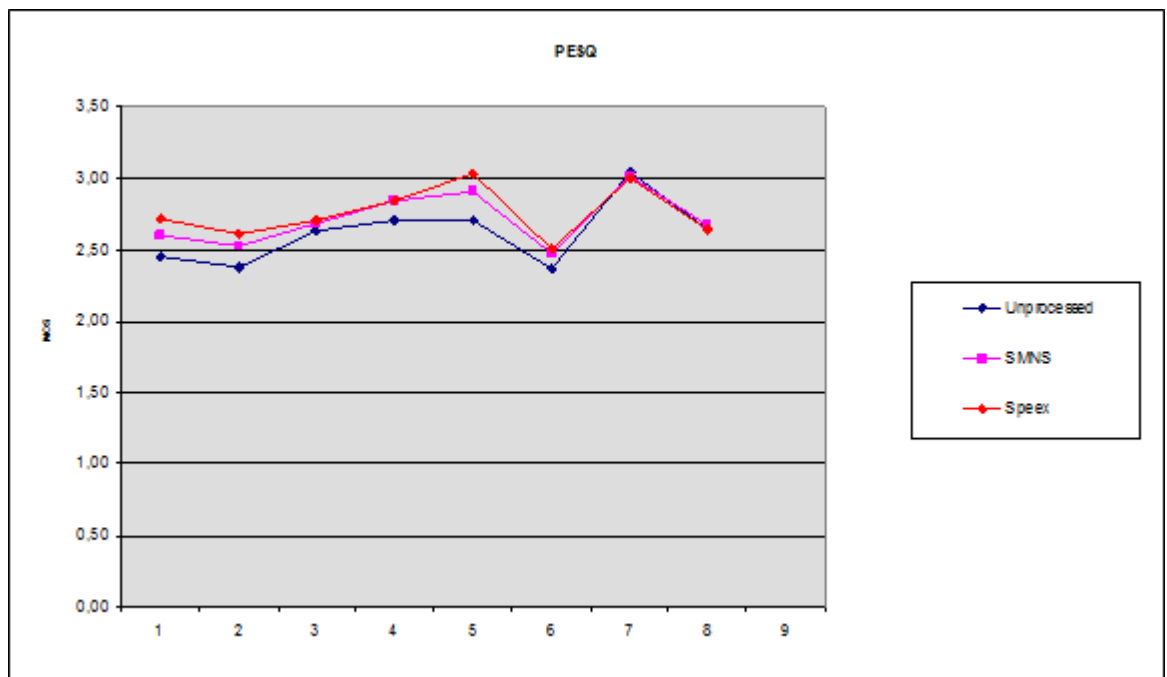
U ovom delu poglavlja su predstavljeni rezultati ispitivanja SMNS algoritma objektivnim metodama merenja, bez analize rezultata, koja je urađena u poslednjem poglavlju. Kao referenca za poređenje kvaliteta rezultata, poslužio je jedan algoritam iste namene. U pitanju je Speex [1] algoritam, koji je, kao i SMNS algoritam, namenjen sistemima sa jednim mikrofonom. Rezultati merenja PESQ i SNRE za različite tipove smetnje, u različitim odnosima signal-šum, za oba algoritma, dati su na slikama 5.2 – 5.9 , odnosno tabelama 5.3 – 5.6. Na slikama na x-osi brojevima su predstavljeni tipovi smetnji iz tabele 5.2, dok su na y-osi date vrednosti za PESQ odnosno SNRE.



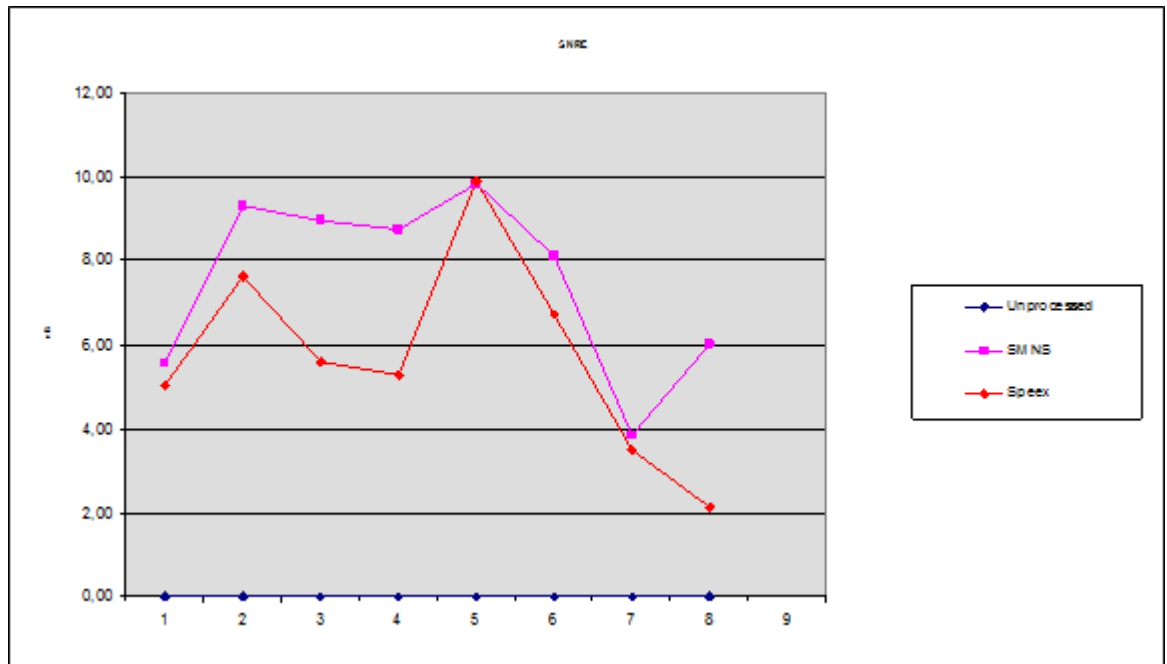
Slika 5.2: PESQ odnos za ispitne vektore muškog govornika, odnos signal-šum 12dB



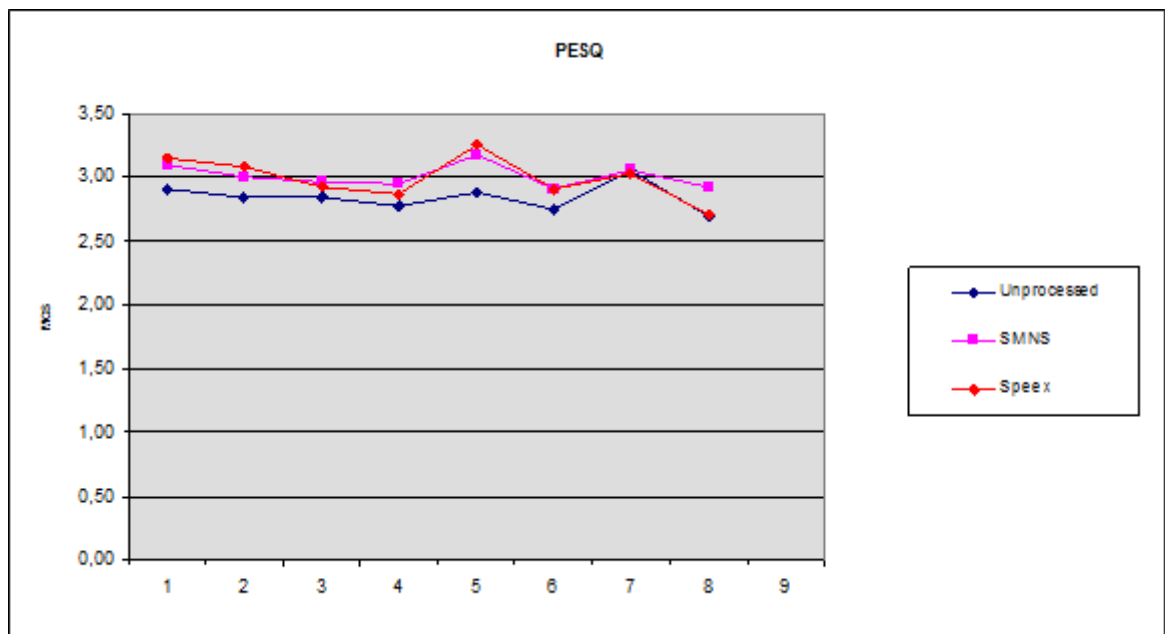
Slika 5.3: SNRE odnos za ispitne vektore muškog govornika, odnos signal-šum 12dB



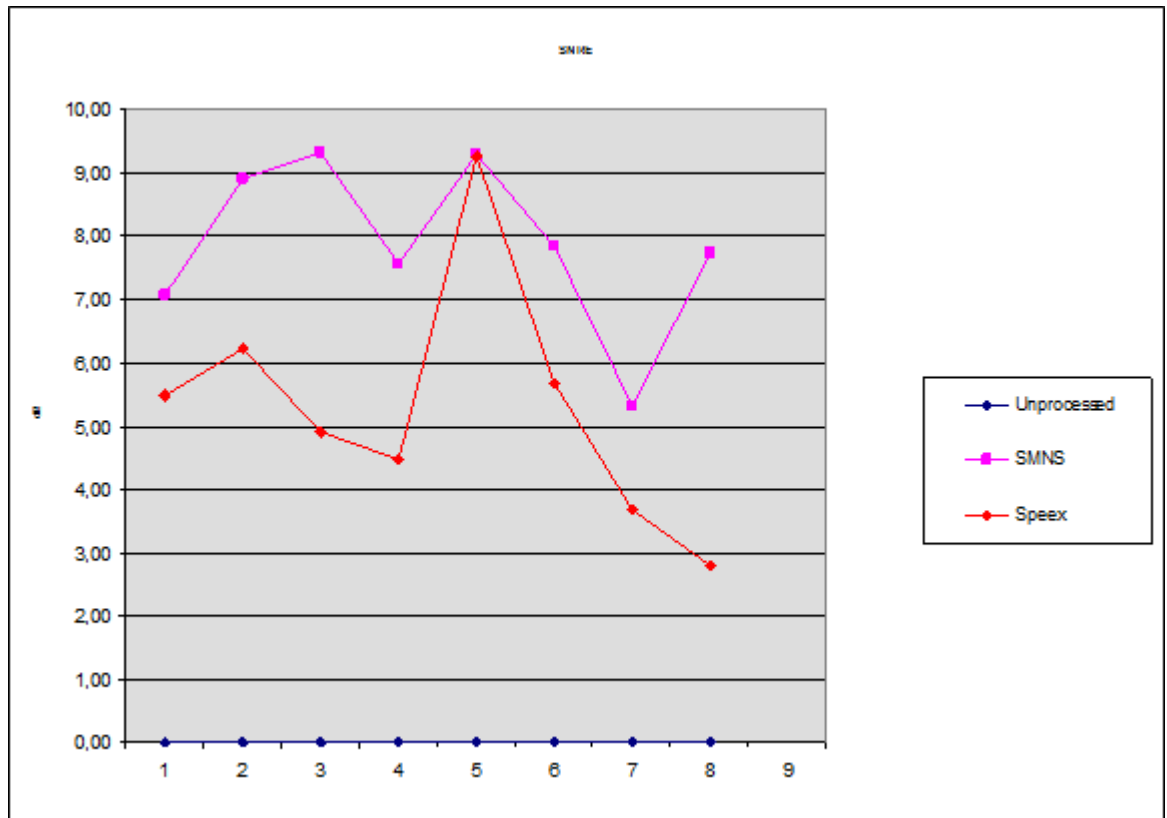
Slika 5.4: PESQ odnos za ispitne vektore muškog govornika, odnos signal-šum 6dB



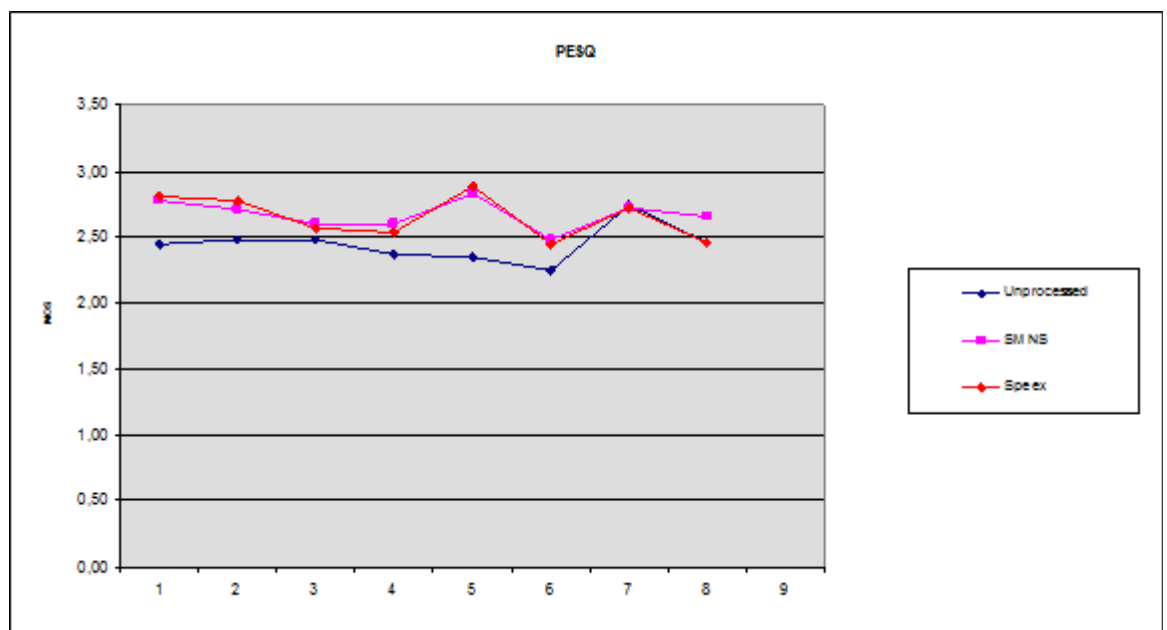
Slika 5.5: SNRE odnos za ispitne vektore muškog govornika, odnos signal-šum 6dB



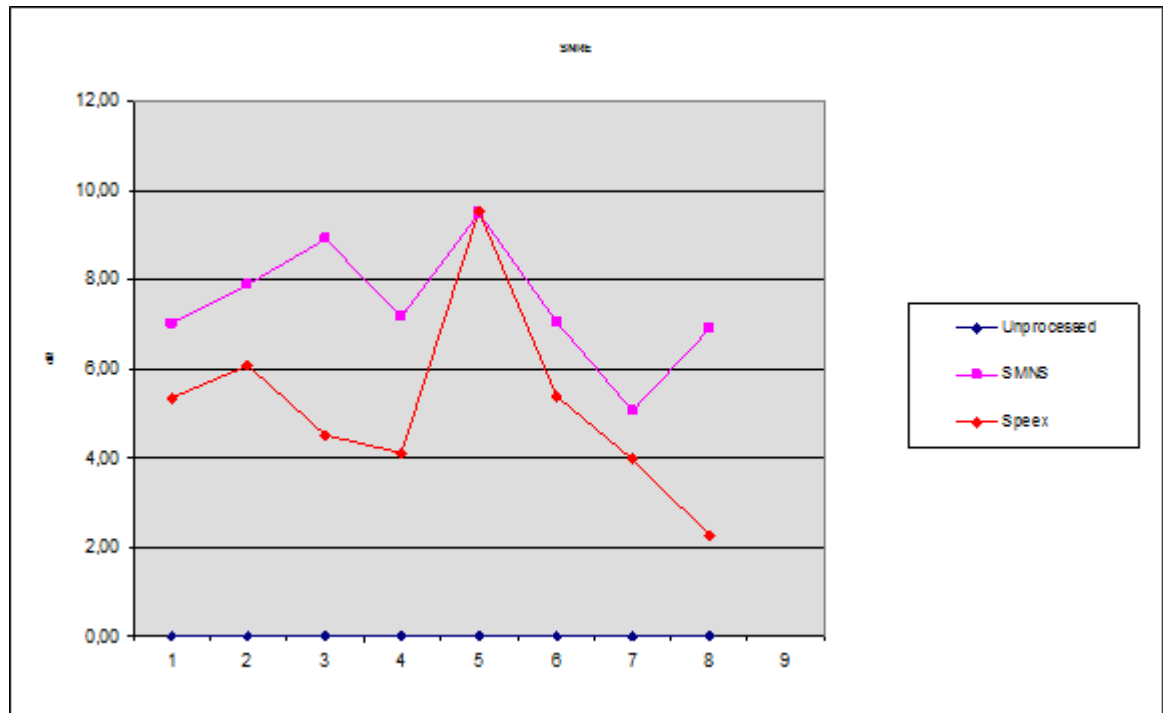
Slika 5.6: PESQ odnos za ispitne vektore ženskog govornika, odnos signal-šum 12dB



Slika 5.7: SNRE odnos za ispitne vektore ženskog govornika, odnos signal-šum 12dB



Slika 5.8: PESQ odnos za ispitne vektore ženskog govornika, odnos signal-šum 6dB



Slika 5.9: SNRE odnos za ispitne vektore ženskog govornika, odnos signal-šum 6dB

	Type	Noise source	Unprocessed		SMNS		Speex	
			PESQ	SNRE	PESQ	SNRE	PESQ	SNRE
1	Street	Work_Noise_Jackhammer_binaural	2,83	0,00	2,93	5,75	3,04	5,11
2	Street	Outside_Traffic_Road_binaural	2,73	0,00	2,85	10,30	2,94	7,72
3	Babble	Pub_Noise_Binaural_V2	2,97	0,00	3,05	9,57	3,07	6,94
4	Pink	Inside_Train_Noise3_binaural	3,04	0,00	3,17	9,30	3,16	6,21
5	Car	Midsize_Car2_130Kmh_binaural	3,09	0,00	3,25	9,54	3,39	9,62
6	Music	Some_music	2,73	0,00	2,82	8,84	2,86	7,23
7	Single voice	Speech	3,34	0,00	3,30	4,45	3,31	4,00
8	Tonal	sine wave - beep	2,89	0,00	2,97	7,27	2,90	2,84

Tabela 5.3: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore muškog govornika, odnos signal-šum 12dB

	Type	Noise source	Unprocessed		SMNS		Speex	
			PESQ	SNRE	PESQ	SNRE	PESQ	SNRE
1	Street	Work_Noise_Jackhammer_binaural	2,45	0,00	2,60	5,57	2,71	5,05
2	Street	Outside_Traffic_Road_binaural	2,37	0,00	2,53	9,31	2,61	7,63
3	Babble	Pub_Noise_Binaural_V2	2,64	0,00	2,69	8,96	2,71	5,61
4	Pink	Inside_Train_Noise3_binaural	2,71	0,00	2,85	8,76	2,84	5,27
5	Car	Midsize_Car2_130Kmh_binaural	2,71	0,00	2,90	9,82	3,04	9,92
6	Music	Some_music	2,37	0,00	2,47	8,10	2,50	6,71
7	Single voice	Speech	3,05	0,00	3,01	3,85	3,01	3,50
8	Tonal	sine wave - beep	2,64	0,00	2,67	6,01	2,64	2,15

Tabela 5.4: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore muškog govornika, odnos signal-šum 6dB

	Type	Noise source	Unprocessed		SMNS		Speex	
			PESQ	SNRE	PESQ	SNRE	PESQ	SNRE
1	Street	Work_Noise_Jackhammer_binaural	2,91	0,00	3,10	7,08	3,15	5,49
2	Street	Outside_Traffic_Road_binaural	2,85	0,00	3,00	8,90	3,08	6,23
3	Babble	Pub_Noise_Binaural_V2	2,84	0,00	2,97	9,32	2,92	4,90
4	Pink	Inside_Train_Noise3_binaural	2,77	0,00	2,96	7,56	2,88	4,47
5	Car	Midsize_Car2_130Kmh_binaural	2,89	0,00	3,17	9,30	3,26	9,26
6	Music	Some_music	2,75	0,00	2,91	7,83	2,92	5,68
7	Single voice	Speech	3,06	0,00	3,05	5,31	3,04	3,66
8	Tonal	sine wave - beep	2,70	0,00	2,93	7,73	2,72	2,79

Tabela 5.5: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore ženskog govornika, odnos signal-šum 12dB

	Type	Noise source	Unprocessed		SMNS		Speex	
			PESQ	SNRE	PESQ	SNRE	PESQ	SNRE
1	Street	Work_Noise_Jackhammer_binaural	2,45	0,00	2,77	7,03	2,81	5,34
2	Street	Outside_Traffic_Road_binaural	2,48	0,00	2,71	7,89	2,78	6,06
3	Babble	Pub_Noise_Binaural_V2	2,48	0,00	2,61	8,90	2,57	4,50
4	Pink	Inside_Train_Noise3_binaural	2,37	0,00	2,60	7,16	2,53	4,10
5	Car	Midsize_Car2_130Kmh_binaural	2,35	0,00	2,83	9,50	2,88	9,54
6	Music	Some_music	2,25	0,00	2,48	7,05	2,45	5,39
7	Single voice	Speech	2,74	0,00	2,73	5,04	2,72	3,95
8	Tonal	sine wave - beep	2,46	0,00	2,65	6,92	2,46	2,27

Tabela 5.6: Brojevne vrednosti PESQ i SNRE mere za neobrađene, sekvence obrađene SMNS i Speex algoritmom za ispitne vektore ženskog govornika, odnos signal-šum 6dB

## 6. Zaključak

U ovom radu urađena je implementacija algoritma za potiskivanje šuma u govornom signalu na platformi sa ograničenim resursima, CS49834 DSP procesorom firme Cirrus Logic. Polazna tačka, odnosno referentni kod (model 0), je najpre prilagođen za prebacivanje na aritmetiku sa nepokretnim zarezom (model 1). Zatim je urađeno prebacivanje sa aritmetike pokretnog na aritmetiku nepokretnog zarezom (model 2), pritom koristeći biblioteke za emulaciju aritmetike u nepokretnom zarezu za Crystal DSP familiju procesora. Kod je napisan u asemblerskom jeziku, integrisan u radno okruženje procesora i optimizovan. Rešenje je verifikovano na dva načina. Prvi način verifikacije je bio vršenjem bit-identičnih ispitivanja prilikom prelazaka između modela, gde je postignuta bit-identičnost između modela 0 i 1 kao i između modela 2 i 3. Između modela 1 i 2 postojala je razlika od jednog bita za sve ispitne vektore, a to pripisujemo upravo prelasku sa aritmetike u pokretnom na aritmetiku u nepokretnom zarezu. Drugi način verifikacije je bio vršenjem ispitivanja korišćenjem objektivnih metoda ispitivanja, PESQ i SNRE. Kao referenca za poređenje poslužio je Speex algoritam, koji je takođe namenjen jednomikrofonskim sistemima, kao i implementirani SMNS algoritam.

Rezultati ispitivanja su pokazali da ovako implementirani algoritam daje slične rezultate kao i Speex algoritam u pogledu objektivne mere PESQ, dok je SNRE u većini slučajeva bolji od Speex algoritma. Što se tiče potrošnje resursa, procesorskog vremena i memorije, ovaj algoritam zadovoljava uslove rada u realnom vremenu, a potrošnja resursa je daleko manja od ukupno dostupne količine resursa na datoj platformi (ukupna iskorišćenost resursa data u tabeli 4.2, dok je ukupno dostupna količina resursa na CS49834 čipu data u tabeli 4.1).

Iz svega navedenog, može se zaključiti da je ovakav algoritam primenljiv na sisteme sa jednim mikrofonom, kao npr. mobilne telefone i da doprinosi značajnom poboljšanju kvaliteta signala.

Pravci daljeg razvoja ovog rešenja bi se ticali poboljšanja detekcije govorne aktivnosti datog algoritma, kao i eventualno procena faze smetnje, umesto korišćenja faze ukupnog signala. Ovo bi gotovo izvesno doprinelo poboljšanju rezultata ispitivanja, odnosno efikasnosti algoritma. Drugo moguće poboljšanje bi se ticalo dodatnih optimizacija potrošnje procesorskog vremena, odnosno smanjivanje broja instrukcija.

## 7. Literatura

- [1] Jean-Marc Valin, „*The Speex Codec Manual Version 1.2 Beta 3*“, December 8, 2007, [www.speex.org/docs/manual/speex-manual.pdf](http://www.speex.org/docs/manual/speex-manual.pdf)
- [2] Ištvan Pap, „*Prilog rešenju obrade govornog signala korišćenjem mikrofonskog niza*“, Doktorska disertacija, Novi Sad 2009
- [3] P. C. Loizou „*Speech Enhancement: Theory and Practice*“, CRC Press, 2007
- [4] J. Thiemann „*Acoustic Noise Suppression for Speech Signals Using Auditory Masking Effects*“, Master thesis, Department of Electrical & Computer Engineering, McGill University, Montreal, Canada, 2001
- [5] S. D. Kamath, P. C. Loizou „*A Multi-Band Spectral Subtraction Method for Enhancing Speech Corrupted By Colored Noise*“
- [6] M. Djukic, N. Cetic, J. Kovacevic, M. Popovic, „*A C compiler based methodology for implementing audio DSP applications on a class of embedded systems*“, IEEE International symposium on consumer electronics, 2008
- [7] Randy Yates, „*Fixed-Point Arithmetic: An Introduction*“, July 7, 2009, [www.digitalsignallabs.com/fp.pdf](http://www.digitalsignallabs.com/fp.pdf)
- [8] „*CS49834 Data Sheet*“, Cirrus Logic, Inc., May 2012
- [9] V. Kovačević, M. Temerinac, M. Popović, N. Teslić, *Arhitekture i algoritmi DSP-a I*, Novi Sad, FTN, 2004
- [10] ITU-T Recommendation P.862 „*Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*“, International Telecommunication Union, 2001

- [11] M. Popovic, J. Kovacevic, „*A Statistical Approach to Model-Based Robustness Testing*“, 14<sup>th</sup> Annual IEEE International Conference and Workshop on Engineering of Computer Based Systems, 2007