SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 2708

SUSTAV ZA ZAŠTITU GOVORNE INFORMACIJE

Igor Mijić

Zagreb, lipanj 2012.

Sadržaj

Uvod

Današnja tehnologija omogućava prikupljanje i spremanje golemih količina informacija. Vlade, vojske, korporacije, financijske instituticije, te privatne tvrtke skupljaju mnogo povjerljivih informacija o svojim zaposlenicima, kupcima, proizvodima, istraživanja i financijskom statusu. Većina tih podataka bivaju prikupljeni, obrađeni i pohranjeni na elektroničkim računalima i prenose se preko mreže na druga računala. Tolika količina povjerljivih podataka za sobom povlači i sisteme zaštite istih. Zaštita podataka pri prijenosu čini jednu važnu cijelinu u sigurnosti tih podataka. Kao primjeri mogu se navesti različiti sigurnosni telefoni, načini enkripcije podataka prije slanja, te dekripcije na prijamnoj strani i slično.

Tema ovog rada je zaštita govorne informacije sa naglaskom na obradu govorne informacije korištenjem kvadraturnih filtarskih slogova (eng. Quadrature Mirror Filter Banks).

Sama ideja rada je da prolaskom kroz analizirajući dio pseudoQMF sloga, dijelimo signal na komponente signala ovisno o frekvencijskoj karakteristici pojasa. Te potpojasne komponente tada tretiramo kao pojedinačne signale, djeleći ih na blokove uzoraka u vremenu. Nad blokovima vršimo pseudoslučajnu permutaciju, te ih ispremješane šaljemo u sintetizirajući dio QMF sloga. Prolaskom kroz slog za sintezu, sve komponente signala se spajaju, te signal ne gubi uzorke, ali je ispremješan i kao takav u njemu ne prepoznajemo nikakvu govornu informaciju.

Predstavljena metoda daje zadovoljavajuć stupanj zaštite, uz relativno ne kompliciranu implementaciju na DSP sustavima, naspram nekih od osnovnih metoda zaštite govora poput vremenske inverzije ili obične permutacije uzoraka govora, koji možda jesu jednostavniji ali nemaju dovoljan stupanj zaštite.

Struktura rada nakon uvoda polako nas vodi kroz povijest i osnovne načine zaštite govorne informacije. Drugo poglavlje se osvrće na pseudoQMF slogove i njihovo projektiranje, dok treće poglavlje prikazuje simulaciju sustava u programskom okružju MATLAB. Implementacija sustava na TMS320C5505 DSP-u opisana je u četvrtom poglavlju, dok se svi MATLAB kodovi potrebni za projektiranje i simulaciju filtarskih slogova i permutatora, nalaze u privitku.

1. Ukratko o zaštiti govorne informacije

Potreba za zaštitom informacija je vjerojatno stara koliko i potreba za komunikacijom između ljudi. Puno različitih tehnika razvijene su tijekom prošlih stoljeća, počevši od Cezarovog šifrarnika do današnjih modernih standarda poput Advenced Encryption Standard-a. S razvojem telekomunikacija, početkom prošlog stoljeća nastala je potreba za enkripcijskim sustavima za zaštitu glasovnih poruka., te je veliki udio prvotnih digitalnih dostignuća razvijeno pokušavajući izvesti sisteme za zaštitu govornih informacija.

Prvi pokušaji enkripcije govora proizlaze iz Bell Labs-a, u vremenu između dva svijetska rata, i naravno proizašli su iz snažnih vojnih potreba. Prvotni sistemi su spajali dva signala, jedan koji je sadržavao govornu informaciju, a drugi šum. Osobe koje bi prisluškivale mogle bi razaznati samo šum, koji je maskirao govor. Na prijemnoj strani govorna informacija bi se izvlačila na račun toga što je sastav šuma bio poznat prijemnom aparatu.

Malo jači sustav razvijen je u vrijeme drugog svijetskog rata te se zvao Sigsaly. Među mnogim od pionirskih svojstava tog 50 tonskog uređaja su prvi primjer kvantizacije govornog signala, prvo korištenje pulsno kodne modulacije, te FSK modulacije.

Neki od hardverskih primjera enkripcijskih sustava su :

- Digital Voice Protection (Motorola)
- STU III (Secure Telephone Unit, Generation III) – pretežito za potrebe telefonske komunikacije unutar vlade SAD-a
- STEs (Secure Terminal Equipment) - pretežito za potrebe video komunikacije visoke brzine unutar vlade SAD-a

Neki od softverskih primjera enkripcijskih sustava su :

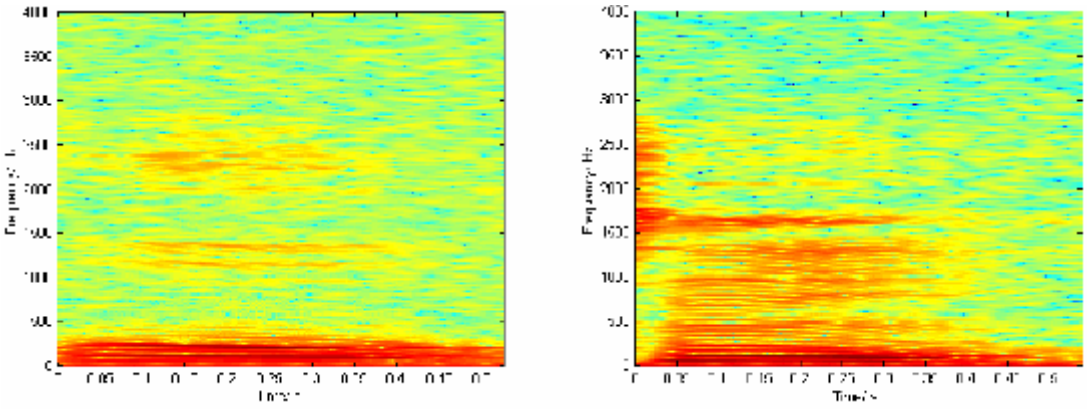
- PGPfone
- Nautilus
- Speak Freely

1.1. Ukratko o ljudskom glasu

Za razvoj i razumijevanje govornih enkripcijskih sustava je vitalno znati neke osnovne parametre ljudskog glasa. Ljudski glas je jednostavno zvuk ili audio signal, generirana od strane čovjeka sa svrhom komunikacije s drugim ljudima.

Dakle, ljudi koriste svoje glasnice koje moduliraju tok zraka, koji dolazi iz pluća, u vibracije, uz nemale utjecaje od strane usne šupljine jezika i usni. Različitim utjecajim tih sustava stvaraju se različiti zvukovi, čija osnovna podjela bi bila na vokale i konsonante.

Stoga, uz razlike u anatomiji između svih ljudi, možemo govoriti o svojstvu glasa da je pomalo različit kod svakog čovjeka. Najveća razlika je naravno najočitija, tj. razlika između glasova muškaraca, žena i djece. Ona postoji radi različitih temeljnih frekvencija, koje nastaju na glasnicama, različitih radi veličina samih glasnica. Glasnice prosječnog muškarca su otprilike 17 do 25 mm u dužini što čini temeljnu frekvenciju između 85 i 155 Hz-a, dok su kod žena između 12.5 i 17.5 mm dužine što daje temeljnu frekvenciju između 165 i 250 Hz-a. Djeca imaju još manje dimenzije glasnica te su im temeljne frekvencije oko 440 Hz-a, kod dojenčadi čak i do 500 Hz-a.



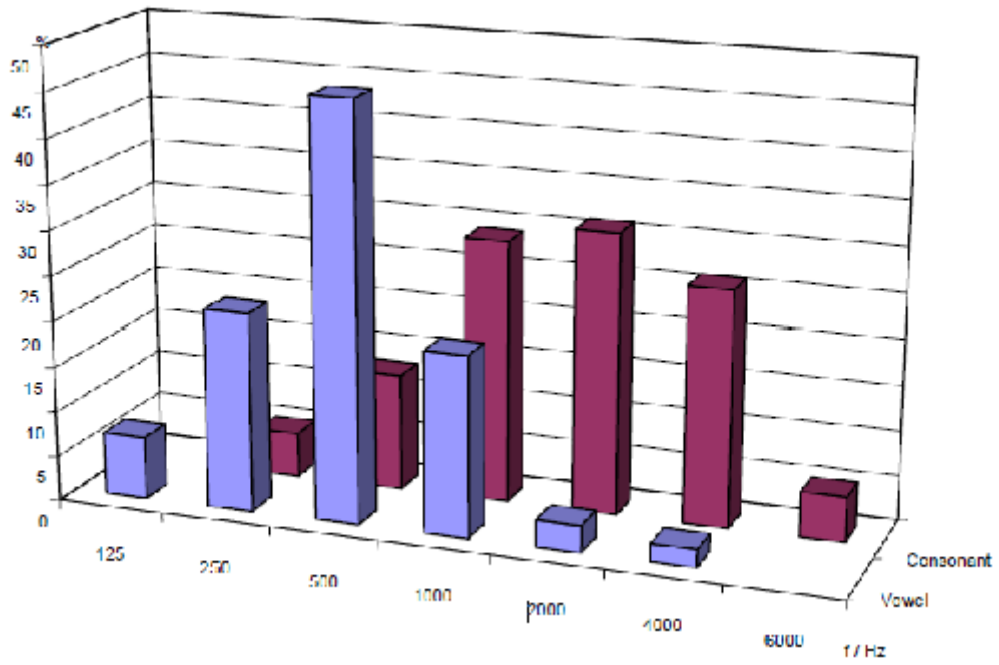
Slika 1.1 : Spektrogram vokala "O" s lijeve strane i konsonanta "T" s desne strane

Slika 1.1 prikazuje spektrogram vokala 'O' sa lijeve strane i konsonanata 'T' sa desne. Možemo primjetiti da je vokal dominantno u niskom području.

Frekvencijsko područje u kojem su sadržane baš sve komponente ljudskog glasa bilo bi između 80 Hz-a i 12 kHz-a s tim da bi većina snage bila sadržana u području između 100 Hz-a i 4 kHz-a. U tim potpojasevima koji su predominantni po snazi nalazi se i većina podataka potrebnih za raspoznavanje govora.

Radi manje potrošnje frekvencijskog pojasa kod telefonije se koristi područje izeđu 340 i 3400 Hz-a, u koje spadaju temeljne frekvencije većine ljudi ili barem osnovni harmonici temeljnih frekvencija koji sadržavaju najviše snage, tako da je odašiljanom govoru uvijek dovoljno spektralnih komponenti koje slušatelju daju osjećaj da uz njih čuje i temeljnu

frekvenciju. Uz tu širinu frekvencijskog pojasa, te par zaštitnih podpojaseva opravdana je frekvencija otipkavanja govora od 8 kHz-a.



Slika 1.2 : Distribucija frekvencija prosječnog ljudskog glasa uz razlikovanje konsonanata i vokala

Na slici 1.2 vidimo distribuciju frekvencija prosječnog ljudskog glasa, uz razlikovanje konsonanata i vokala.

1.2. Tehnike zaštite govorne informacije

Premetanje ili izokretanje (eng. scrambling) je riječ koju koristimo kada se referiramo na namjerni utjecaj na signal sa svrhom maskiranja njegove poruke pri njegovom prijenosu. U stručnoj literaturi kada govorimo o zaštiti informacija izokretanje spominjemo u značenju analognog utjecaja dok riječ enkripcija koristimo sa značenjem digitalnog utjecaja na signal. U daljnjem tekstu su spomenuti i ukratko objašnjeni najvažniji primjeri oba utjecaja.

1.2.1. Izokretanje u vremenskoj domeni

Jedna od osnovnih tehnika izokretanja signala je dijeljenje snimljenih govornih podataka na manje blokove, te njihovo odašiljanje u promjenjenom redoslijedu. Ovisno o duljini odaslanog signala, te veličini i broju samih blokova zaštita postaje sve veća. Na primjer, signal duljine 1 s, podijeljen na 10 blokova ima oko 3600 mogućih permutacija. No, loša strana ovakvog sustava je da se signal još uvijek sastoji od istih frekvencijskih komponenti,

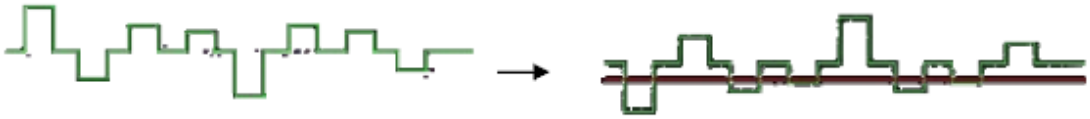
te se neke osnovne informacije poput spola govornika ili čak i nekih izgovorenih glasova mogu vrlo lako izvući analizom.

1.2.2. Izokretanje u frekvencijskoj domeni

Druga, još raširenija osnovna tehnika izokretanja signala je izokretanje frekvencija signala. U osnovi, spektar visokih frekvencija se prebacuje u niske, i obrnuto. Ova tehnika se najčešće naziva inverzijom govora jer se govorni signal izokreće oko jedne frekvencije.

Postoje tri osnove verzije izokretanja u frekvencijskoj domeni:

- Inverzija u osnovnom pojasu – spektar se izokreće oko jedne nepromjenjive frekvencije, radi čega metoda ima veoma nisku kriptološku jačinu
- Inverzija u promjenjivim pojasevima – (rolling phase inversion) signal se izokreće oko stalno promjenjive frekvencije, tako da dekripcija postaje moguća samo ako su poznate sve frekvencije oko koje se signal mijenja.
- Inverzija u više pojaseva - inverzija u osnovnom pojasu ali primjenjena na signalima dobivenim dijeljenjem signala u komponente koje pripadaju različitim pojasevima frekvencija sadržanim u početnom signalu



Slika 1.3 : Izokretanje frekvencije inverzijom u osnovnom pojasu

1.2.3. Standardi digitalne enkripcije

Princip kod enkripcije je uvijek jednak, niz digitalnih uzoraka, ili blokova uzoraka prolazi kroz algoritam koji ovisno o matematičkim funkcijama na kojima se baziraju, unose bitove u signal čineći ga manje čitljivim, ili različitijim od originala, što je više moguće. Pojednostavljeno, dobar enkripcijski algoritam mijenja originalni signal u tolikoj mjeri da niti jedna od karakteristika originalnog signala nije raspoznatljiva ili sačuvana. Važno svojstvo enkripcijskih algoritama je potreba ključa za dekripciju, tj. to da osiguraju nemogućnost dekripcije bez poznavanja ključa u slučaju da je poznat sam algoritam enkripcije i njegov inverz.

Primjeri najpoznatijih bi bili :

- DES (Data Encryption Standard)
- AES (Advanced Encryption Standard)
- FEAL (Fast Data Encryption Algorithm)
- IDEA (International Data Encryption Algorithm)
- Safer (Secure and Fast Encryption Routine)
- RC5 (Rivest's Code 5) and RC6 (Rivest's Code 6)

1.2.4. VOCODER enkripcija

U digitalnim telekomunikacijskim sustavima, koderi govora se često koriste da se smanji potrebna širina pojasa kod odašiljana. Vocoder je uređaj koja analizira zvuk kojeg generira ljudski govorni trakt i iz njega izvlači informacije koji se onda šalju, a iz kojih se na drugom kraju komunikacijskog kanala može uspješno sintetizirati signal što sličniji originalnom govoru. Čineči to vocoderi uvelike smanjuju količinu informacija potebnu za spremanje ili prijenos govornih informacija. Vocoderi se najčešće implementiraju korištenjem linearne predikcije. Primjenom bilo kojeg od spomenutih načina enkripcije na izlazni signal vocodera, koji već sam po sebi sadrži informacije koje se razlikuju od osnovnog govornog signala, dobivamo signal visoke kriptološke snage.

Neki od standarda vocoder su :

- LPC-10, FIPS Pub 137
- Code Excited Linear Prediction, (CELP), Federal Standard 1016
- Continuously Variable Slope Delta-modulation (CVSD)
- Mixed Excitation Linear Prediction (MELP)
- Adaptive Differential Pulse Code Modulation (ADPCM), former ITU-TG.721

2. Digitalni filtarski slogovi

Digitalni filtarski slogovi (eng.filter banks) predstavljaju skup digitalnih filtara s zajedničkim ulazom ili izlazom pomoću kojih se jedan signal može rastaviti na više različitih frekvencijski ograničenih signala ili više frekvencijski ograničenih signala sastaviti u jedan zajednički signal. Slogovi za analizu (eng. analysis bank) rastavljaju ulazni signal, a slogovi za sintezu ga rekonstruiraju, zbrajanjem signala nastalih u pojedinim pojasevima sloga. Najvažnije svojstvo gore opisane strukture jest da filtriranjem ne gubimo značajne količine informacija, tj. u idealnom slučaju niti jedan dio korisne informacije ulaznog signala nebi bio izgubljen.

Jedna od prednosti korištenja digitalnih filtarskih slogova iz porodice kvadraturnih zrcalnih slogova je poboljšanje razumljivosti podataka pri dekripciji radi relativno ravne amplitudno-frekvencijske karakteristike samih slogova čime se izbjegava gubljenje informacija sadržanih u frekvencijama koje bi se u drugim filtrima izgubile.Nadalje, pojačava se kriptološka snaga uređaja za kodiranje jer se omogućuje malim količinama signala da se obrađuju, tj. blokovi koji se obrađuju mogu korespondirati sample-ima AD pretvornika.

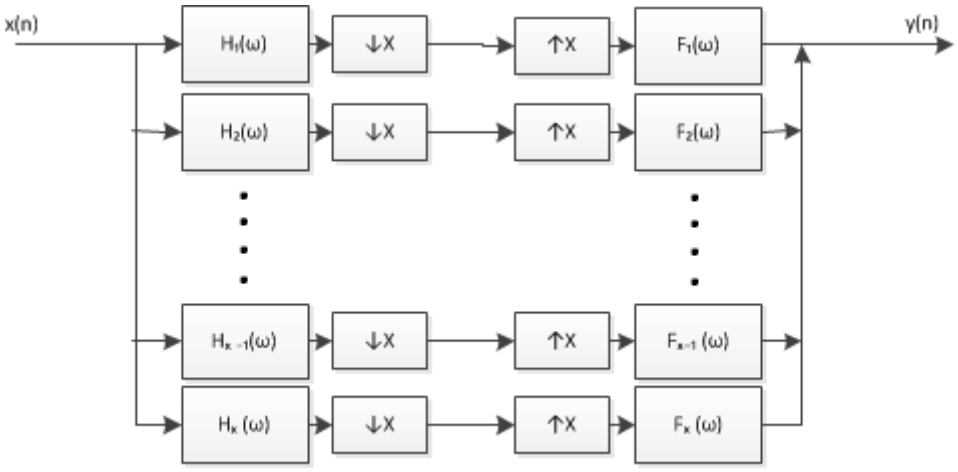
U ovom radu koristiti će se pseudo kvadraturni zrcalni filtarski slogovi opisani u slijedećem potpoglavlju.

2.1. PseudoQMF filtarski slogovi

Kao posebna porodica kadraturnih zrcalni filterskih slogova dani su pseudo-kvadraturni zrcalni maksimalno decimirani filtarski slogovi s M pojaseva (eng. M-band pseudoQMF). Oni se sastoje od slogova za analizu i sintezu. Slog za analizu maksimalno decimiranog pseudoQMF-a sastoji se od M paralelnih pojasno propusnih filtara H koje slijede blokovi za smanjenje frekvencije otipkavanja po bazi M (eng. decimator) dok je svaki slog za sintezu analogno slogu za analizu sastavljen od blokova za povećanje frekvencije otipkavanja po bazi M (eng. interpolator), te pojasno propusnih filtara F. Struktura pseudoQMF filtarskog sloga prikazana je na slici 2.1.

Kod pseudoQMF filtarskih slogova ne možemo govoriti o svojstvu perfektne rekonstrukcije, nego moramo osigurati što kvalitetniju rekonstrukciju, te ih tada zovemo slogovima sa skoro perfektnom rekonstrukcijom. Izvori izobličenja rekonstruiranog signala

kod pseudoQMF slogova su amplitudno izobličenje, fazno izobličenje te preklapanje spektra (eng. aliasing).



Slika 2.1 : Struktura N pojasnog pseudoQMF filtarskog sloga

Općenito, sve vrste filtarskih slogova ovise o ravnoj amplitudnoj karakteristici pojedinih filtara koja osigurava nepojavljivanje amplitudnih izobličenja, te o linearnoj faznoj karakteristici koja jamči nepojavljivanje faznih izobličenja. Također, važno svojstvo je ortogonalnost modulacijskih funkcija koje se primjenjuju na prototipnom niskopropusnom filtru da bi se stvorili ostali filtri sloga, gdje ortogonalnost osigurava kvalitetnu rekonstrukciju faze i amplitude. Da bi se izbjegla izobličenja koja unosi preklapanje spektra, pseudoQMF filtarski slogovi poništavaju izobličenja između susjednih pojasaeva, te pretpostavljaju da ne dolazi do preklapanja između nesusjednih pojaseva, tj. da je gušenje u zaustavnom pojasu bilo kojeg filtra naspram svih nesusjednih pojaseva beskonačno. Iako kod realnih izvedbi filtara to nije točno, pseudoQMF filtarski slogovi dovoljno dobro funkcioniraju ako je atenuacija zaustavnog pojasa dovoljno visoka u odnosu na izlaznu rezoluciju rekonstruiranog signala.

Kod filtara sa skoro perfektnom rekonstrukcijom amplitudna karakteristika filtarskog sloga je skoro potpuno ravna, problem preklapanja spektra je isto skoro rješen, te jedino značajno izobličenje koje filtarski slog unosi signalu je kašnjenje.

2.2. Projektiranje pseudoQMF filtarskih slogova

Visoka kvaliteta rekonstrukcije signala kod pseudoQMF filtarskih slogova ovisi o tome u kolikoj mjeri prototipni niskopropusni filtar zadovoljava izraze (1) i (2). Izraz (1) eliminira

amplitudna izobličenja nastala prolaskom signala kroz slogove za analizu i sintezu,

$$|H(\omega)|^2 + \left|H\left(\omega - \frac{\pi}{M}\right)\right|^2 = 1, za\ 0 < \omega < \frac{\pi}{M} \quad (1)$$

dok izraz (2) osigurava da neće doći do preklapanja spektra između nesusjednih pojaseva

$$|H(\omega)|^2 = 0, za\ \omega > \frac{\pi}{M} \quad (2)$$

Preklapanje spektra između susjednih pojaseva rješava se koristeći prikladne modulacijske faktore. Impulsni odzivi moduliranih filtara u slogu za analizu su dani izrazom (3)

$$h_k(n) = 2h(n)cos\left[(2k+1)\frac{\pi}{M}\left(n - \frac{N-1}{2}\right) + (-1)^k\frac{\pi}{4}\right] \quad (3)$$

dok su filtri sloga sa sintezu dani izrazom (4)

$$f_k(n) = 2h(n)cos\left[(2k+1)\frac{\pi}{M}\left(n - \frac{N-1}{2}\right) - (-1)^k\frac{\pi}{4}\right] \quad (4)$$

za n i k

$$0 \le n \le N-1 \quad (5)$$

$$0 \le k \le M-1 \quad (6)$$

gdje su N i M redovi filtara koji se koriste u filtarskom slogu, odnosno broj pojaseva u filtarskom slogu.

Konstrukcija filtra sa konačnim impulsnim odzivom, tj. FIR filtra, koji u potpunosti zadovoljava uvjete savršene rekonstrukcije nije moguća, tako da se okrećemo metodama koje aproksimativno zadovoljavaju uvjete (1) i (2), te se koristeći njih dizajniraju filtarski slogovi sličnih svojstava slogovima sa savršenom rekonstrukcijom.

Koristeći metodu predloženu u radovima C. Creuserea i S. Mitre problem rješavamo dizajnom niskopropusnog filtra kojeg onda provlačimo kroz algoritam koji pomiče rub propusnog pojasa filtra, ispitujući u kolikoj mjeri filtar zadovoljava uvjete (1) i (2) sve dok ne dođe unutar postavljenih tolerancija. Metoda koju Creusere i Mitra predlažu za dizajn prototipa je Parks-Mcclellanov algoritam kojeg koristimo i mi pošto omogućava kostrukciju filtara sa poprilično ravnim karakteristikam i laticama u zaustavnom pojasu, te je u MATLAB programskom paketu već izveden kao `firpm` funkcija.

Pri početku izvedbe algoritma postavaljamo fiksni red filtra ovisno o broju pojaseva M, odnosno N=8*M, da bi postigli gušenje u stopbandu veće od 40 dB koliko nam je

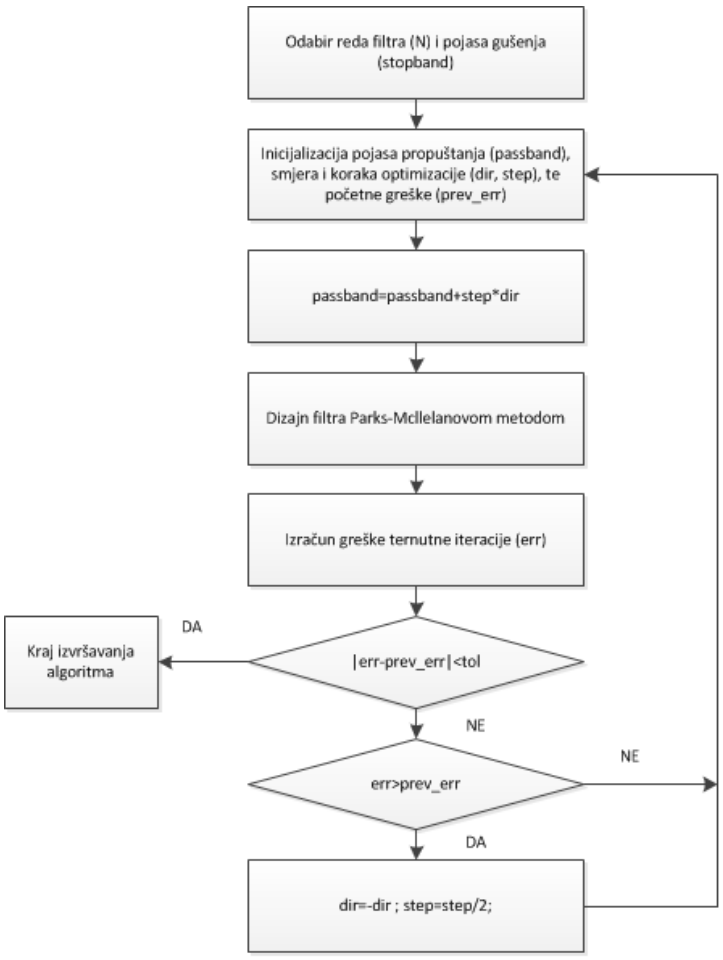
potrebno za kvalitetnu rekonstrukciju signala koje nose glas. Kada bi prenosili audio signale sa širim frekvencijskim područjem, izlazna rezolucija tih signala bi bila veća (>100dB) te bi trebali povećati i gušenje u nepropusnim djelovima filtra, tj. red filtara.

Algoritam jednostavno pomiče rub pojasa propuštanja ovisno o veličini varijable **step**, dizajnira novi filtar, te izračuna pogrešku **err**. Kad god pogreška pređe vrijednost pogreške prošle iteracije **prev_err**, vrijednost **step**-a se prepolovi i smjer pomaka **dir** promjeni. Optimizacijski proces završava kada razlika između pogreške trenutne iteracije i prošle iteracije ulazi u vrijednost unaprijed zadane tolerancije **tol** .

Pogreška **err** koju se optimizira računa se po izrazu 7.

$$\emptyset = max_{\omega}\left\{|H(\omega)|^2 + \left|H\left(\omega - \frac{\pi}{M}\right)\right|^2 - 1\right\} \tag{7}$$

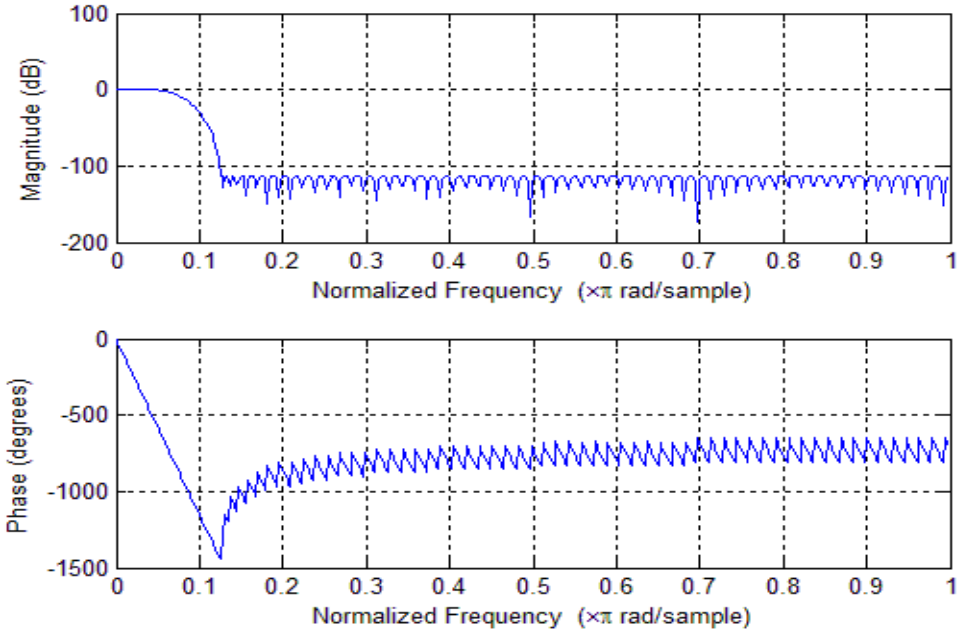
Dijagram toka na slici 2.2. pokazuje rad algoritma nakon izbora reda filtra te frekvencija na kojim počinje gušenje.



Slika 2.2 : Prikaz algoritma za projektiranje niskopropusnog prototipnog filtra

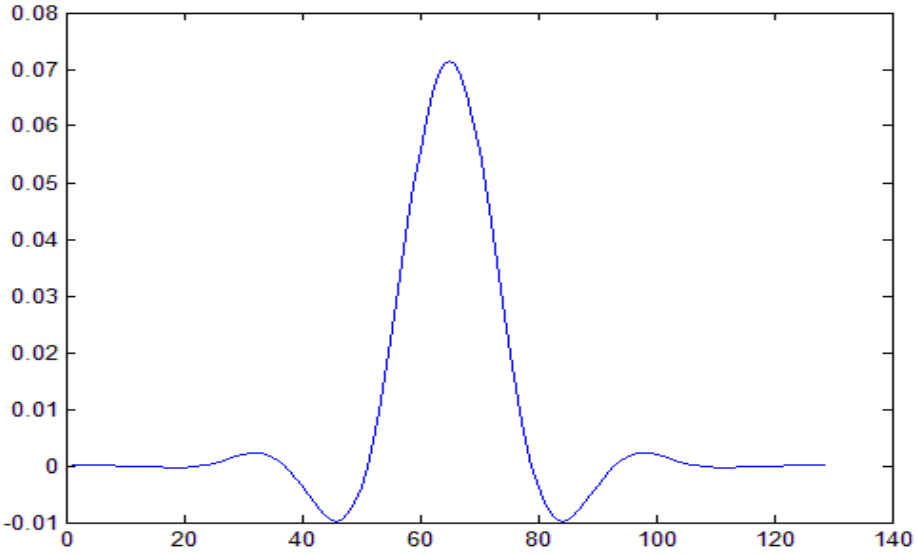
2.3. Primjer pseudoQMF filtarskog sloga

Primjer niskopropusnog prototipnog filtra biti će prikazan za M=8 pojaseva, te N=16*M red filtra. Slika 2.3 prikazuje amplitudno-frekvencijsku i fazno frekvencijsku karakteristiku prototipnog filtra,



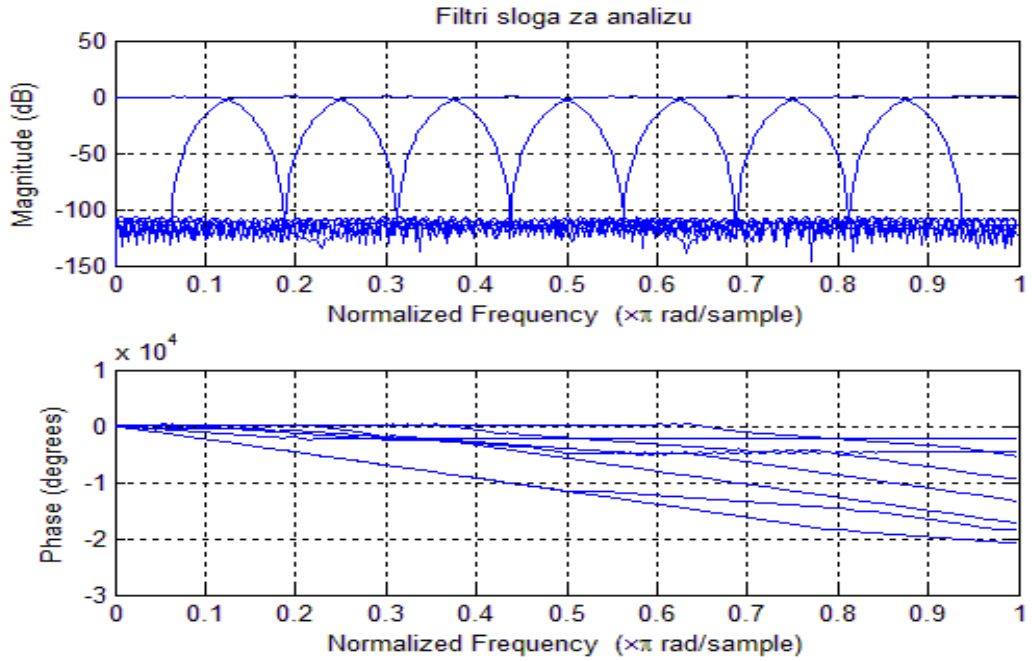
Slika 2.3 : Frekvencijska karakteristika prototipnog niskopropusnog filtra

dok slika 2.4 prikazuje impulsni odziv

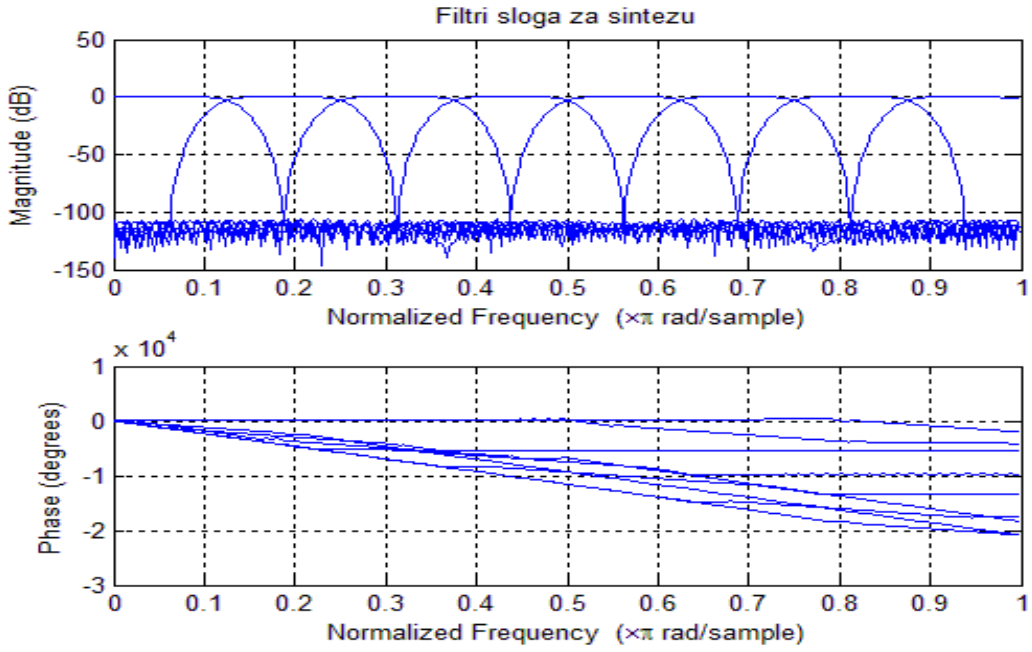


Slika 2.4 : Impulsni odziv prototipnog niskopropusnog filtra

Ostale filtre u slogovima dobivamo koristeći izraze (3) i (4), te slike 2.5 i 2.6 prikazuju frekvencijske karakteristike svih filtara slogova za analizu i sintezu.

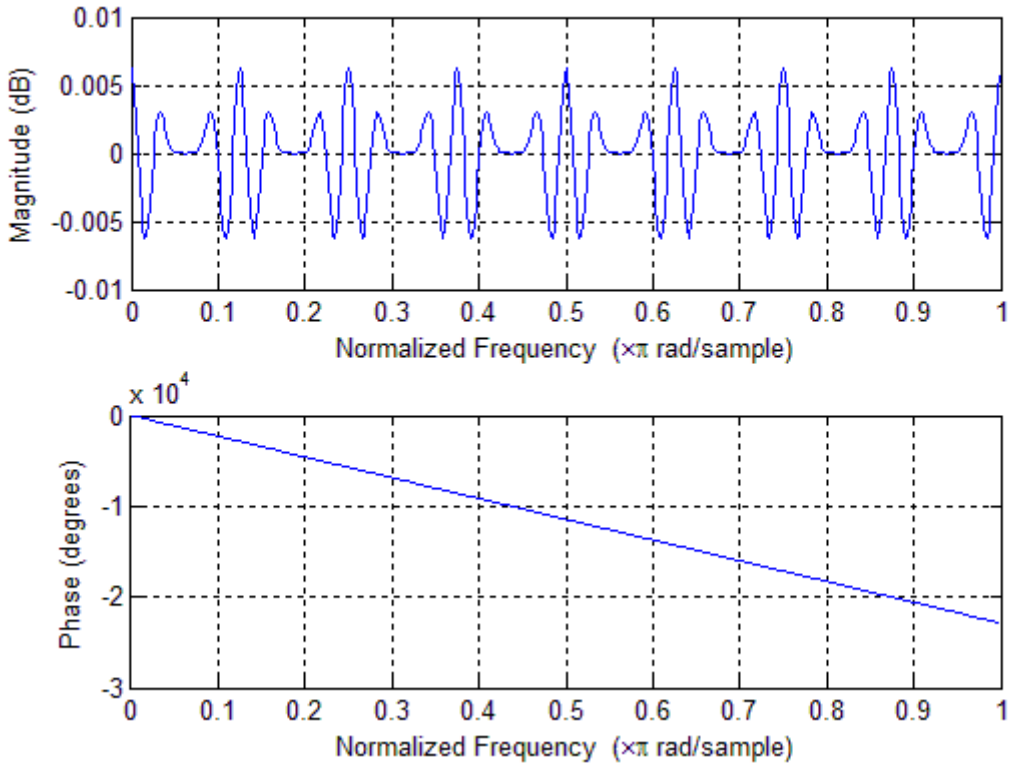


Slika 2.5 : Frekvencijske karakterstike filtara sloga za analizu



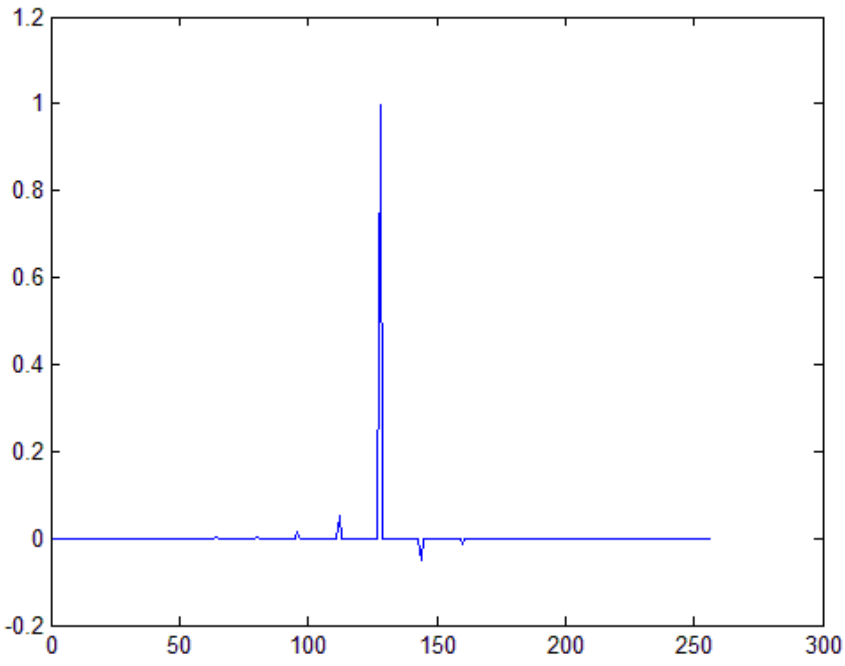
Slika 2.6 : Frekvencijske karakteristike filtara sloga za sintezu

Kaskadirajući date slogove dobivamo amplitudnu i frekvencijsku karakteristiku cijelog sloga prikazanu na slici 2.7. , te njen impulsni odziv na slici 2.8.



Slika 2.7 . Frekvencijska karakteristika kaskadiranih slogova za sintezu i analizu

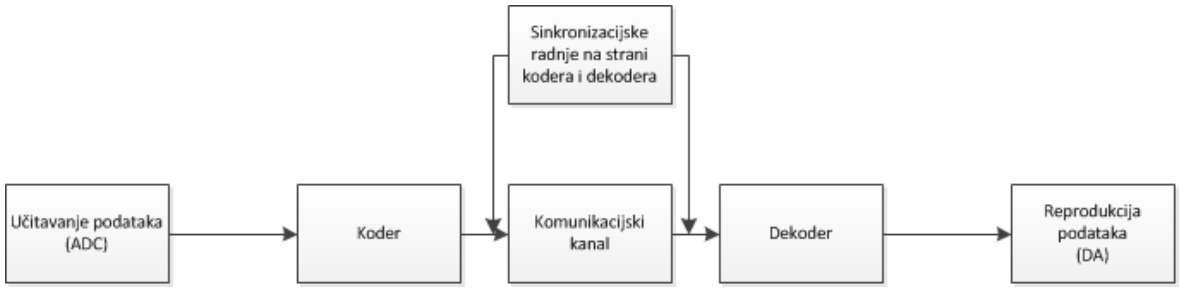
Iz slike 2.7 se vidi da cijeli filterski slog unosi jako malu grešku u sam signal.



Slika 2.8 Impulsni odziv kaskadiranih slogova za sintezu i analizu

3. Simulacija sustava u programskom okružju MATLAB

Blok shema sustava za zaštitu prikazana je slikom 3.1.

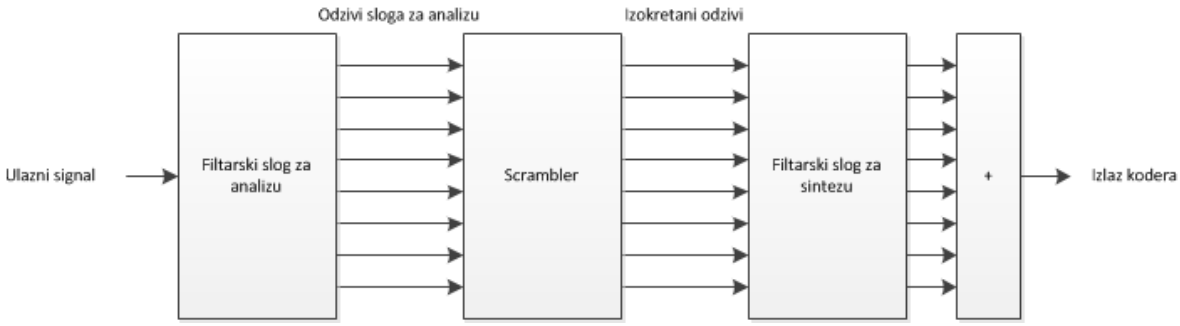


Slika 3.1 : Blok shema sustava

Govorni signal sa mikrofona uzorkujemo te polje uzoraka šaljemo na koder. Koder potpojasno izokreće signal te konstruira vremenski i frekvencijski izobličen signal. Na ulazu i izlazu komunikacijskog kanala vrše se radnje potrebne za sinkronizaciju između kodera i dekodera. Dekoder potpojasno izokreće ulazni signal, te poznavajući ključ po kojem je originalni signal izobličen rekonstruira originalni signal. Izlaz dekodera se nakon DA pretvorbe dovodi na izlaz zvučnika.

3.1. Koder

Blok shema kodera prikazana je slikom 3.2.



Slika 3.2 : Blok shema kodera

Koder, kao i dekoder koristi koeficijente filtara izračunate optimizacijskim algoritmom opisanim u potpoglavlju 3.2. Koristimo filterske slogove sa M=16 potpojaseva, sa redovima filtara N=16*16. Iako bi manji redovi filtara omogućili jednostavniju i bržu izvedbu kodiranja i dekodiranja, viši redovi nam omogućavaju bolju rekonstrukciju, te za potrebe simulacije nemoramo razmišljati o optimizaciji sustava.

Funkcija `koder.m` dakle prima koeficijente filtara, te uzorke govornog signala učitanog naredbom `wavread`. Uzorci se dovode na ulaz analizirajućeg sloga, gdje se signal konvoluira sa impulsnim odzivima filtarskog sloga, što nam daje 16 odziva koji korespondiraju potpojasnim komponentama ulaznog signala. Odzivi su duljine (N+1)+(length(signal)-1 gdje je (N+1) duljina impulsnih odziva filtara u slogu. Svih 16 odziva se tada poduzorkuju, te se provode kroz funkciju scrambler.

Funkcija scrambler funkcionira tako da ulazno polje uzoraka y rascijepa na n blokova, te ih ispremješta ovisno o ključu r. Pri simulaciji smo koristili n=16 blokova.

Da bi postigli dovoljnu enkripcijsku snagu, svaki od pojaseva ispremještamo po različitom ključu, no pošto pri komunikaciji ne šaljemo ključ u poruci, moramo koristiti ključeve koji su odprije poznati i koderu i dekoderu. U najednostavnijem slučaju spremimo u koder i dekoder jednu slučajnu permutaciju brojeva između 1 i 16.

Tu permutaciju koristimo kao ključ za prvi potpojas, a za svaki slijedeći pomićemo r za jedno mjesto u desno. Čineći to dobivamo i izokretanje u frekvencijskoj domeni, što nebismo dobili kada bi sve potpojase ispremetali sa istim ključem. Primjer polja koje sadrži ključeve za 8 pojaseva s podjelom u 16 blokova prikazan je slikom 3.3.

r[1]	3	11	8	2	4	9	16	15	1	6	13	5	7	12	10	14
r[2]	14	3	11	8	2	4	9	16	15	1	6	13	5	7	12	10
r[3]	10	14	3	11	8	2	4	9	16	15	1	6	13	5	7	12
r[4]	12	10	14	3	11	8	2	4	9	16	15	1	6	13	5	7
r[5]	7	12	10	14	3	11	8	2	4	9	16	15	1	6	13	5
r[6]	5	7	12	10	14	3	11	8	2	4	9	16	15	1	6	13
r[7]	13	5	7	12	10	14	3	11	8	2	4	9	16	15	1	6
r[8]	6	13	5	7	12	10	14	3	11	8	2	4	9	16	15	1

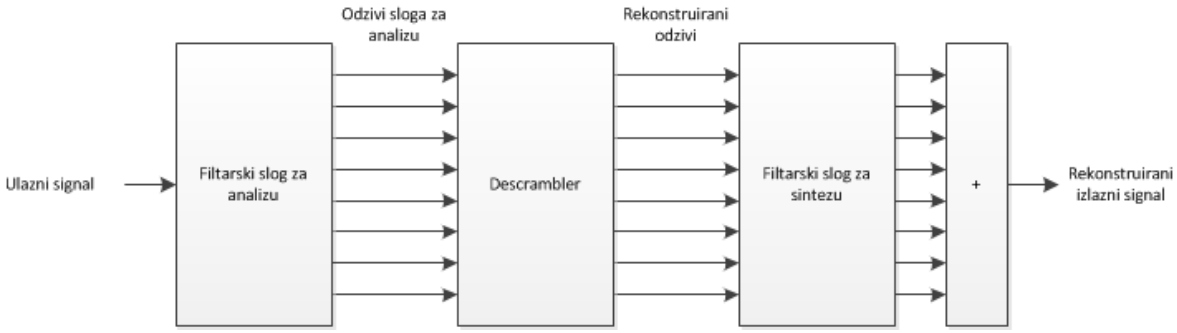
Slika 3.3 : Prikaz ključeva izokretanja za različite pojaseve – 8 pojaseva/16 blokova

Ispremetani uzorci se tada naduzorkuju da dobijemo isti broj uzoraka kao i na izlazu analizirajućeg sloga, te se provode kroz sintetiziraući slog, konvoluirajući ih sa pripadnim filtrima f. Odzivi filtara sintetizirajućeg sloga se sumiraju, te dobijemo vremenski i frekvencijski izokretan signal koji još mora proći kroz modulaciju sa dva signala bliskih

frekvencija da bi se mogla omogućit sinkronizacija na strani dekodera. Nakon modulacije dobijemo signal spreman za slanje u komunikacijski kanal.

3.2. Dekoder

Funkcionalnost dekodera je veoma slična koderu. Filtarski slogovi za analizu i sintezu su jednaki. Koriste se jednaki koeficijenti FIR filtara kao i kod kodera. Nakon razlaganja signala na potpojasne komponente u slogu za analizu, funkcija descrambler, poznavajući ključ kojim su originalne potpojasne komponente kodirane, preslaguje blokove u pravom redu. Blok shema dekodera prikazana je slikom 3.4.

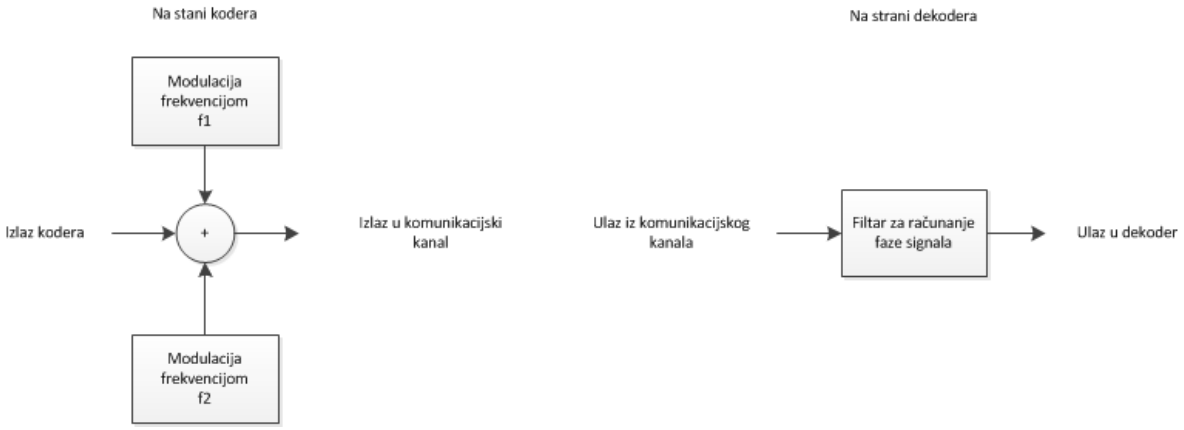


Slika 3.4 : Blok shema dekodera

3.3. Sinkronizacija

Sinkronizacija odašiljača i prijamnika je vitalan dio svake komunikacije. Da bi kvalitetno dekodirao podatak, dekoder se mora postaviti u početni uzorak, dakle mora prepoznati fazu signala kojeg prima. Sama sinkronizacija nije implementirana u sustav kod simulacije, ali pošto znamo kolika je njena važnost kod izvedbe izveden je skripta koja daje primjer sinkronizacije. Na izlazu iz kodera, signal se modulira sa dva signala bliskih frekvencija postavljenim po sredini dva gornja pojasa. Govorni signal uzorkujemo sa 8000 Hz-a, tako da teoretski očuvamo frekvencije do 4000 Hz-a. Pošto je većina informacija potrebna da se očuva razumljivost i prepoznatljivost ljudskog glasa sačuvana u pojasu do 3500 Hz-a, pojasevi od 3500 do 3750, te 3750 do 4000, nisu nam bitni u vidu govorne informacije, tako da signal moduliramo sa frekvencijama f1=3625 Hz i f2=3875 Hz. Takav modulirani signal već je izokretan, pa ga daljnja modulacija sa dva signala iz područja visokih frekvencija čini još neprepoznatljivijim.

Blok shema radnji vezanih uz sinkronizaciju prikazana je slikom 3.5.



Slika 3.5 : Blok shema sinkronizacije

Na strani dekodera, računaju se faze na frekvencijama f1 i f2, te se iz toga faza cijelog signala, čime se dekoder može postaviti u početni uzorak, te se signal može kvalitetno dekodirati.

Pošto na raspolaganju nemamo pravi komunikacijski kanal, simuliramo ga tako da iz polja govornih informacija uzemo N uzoraka na proizvoljnoj poziciji u polju. Na polju od takvih N proizvoljnih uzoraka nepoznajemo relativni odnos sa početnim uzorkom, te smo u sličnoj situaciji kao kad bi prijemnik primio polje uzoraka za koje nebi bio siguran da li je cijelovito ili početno. Kada bi računali fazu samo jednog signala, dobili bi informaciju samo o tome kolika je faza od 0 do 2Π. Računajući faze dva signala bliskih frekvencija možemo izračunati koliko su punih okretaja od 2 Π prošli od trenutka odašiljanja što nam omogućava rekonstrukciju faze cijelog signala.

Faze na frekvenciji računamo Goertzelovim algoritmom, pošto je mnogo jednostavnije i mnogo manje procesorski zahtjevno nego računanje FFT-a za cijeli spektar. Goertzelov algoritam olakšava računanje faze za frekvencije koje očekujemo u spektru, te je opisan u slijedećem potpoglavlju.

3.3.1. Göertzelov algoritam

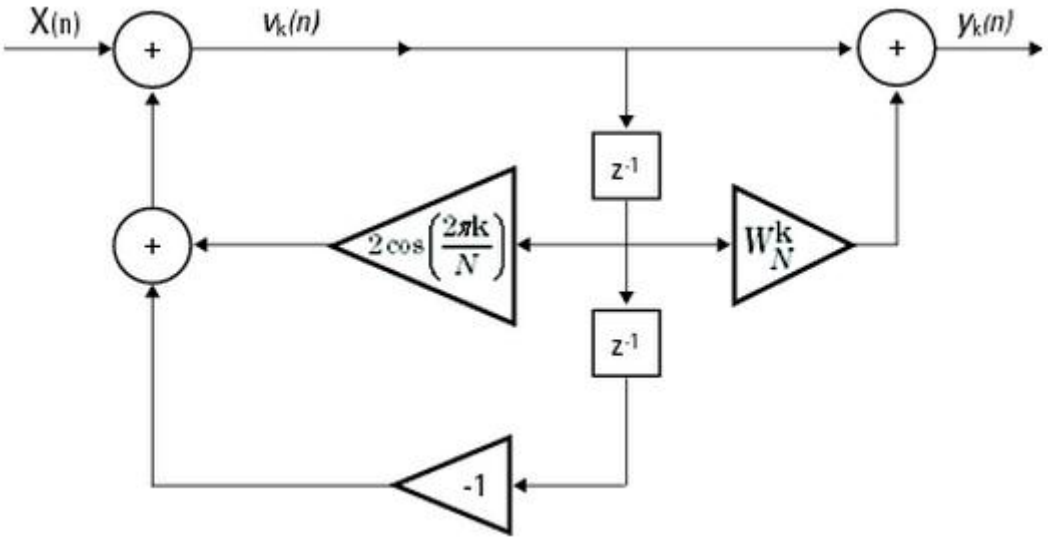
Göertzelov algoritam koristimo jer je efikasniji od FFT-a ako nam treba DFT od samo nekoliko frekvencija, što je veoma važno za rad u stvarnom vremenu na DSP-u sa ograničenim procesorskim resursima.

Göertzelov algoritam implementira DFT kao rekurzivnu diferencijsku jednadžbu. Da dođemo do te jednadžbe, DFT izražavamo kao kovoluciju ulaza *x(n)* sa impulsnim odzivom $h(n) = W_N^{-kn} u(n)$.

gdje je $W_N^{-kn} = e^{-i2\pi k/N}$. Z-transformacija impulsnog odziva sustava je dana sa

$$H(z) = \frac{1 - W_N^k z^{-1}}{1 - 2cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}} \quad (8)$$

te je njegova izvedba kao IIR filtar prikazana na slici 3.8.



Slika 3.8 : IIR izvedba Goertzelovog algoritma

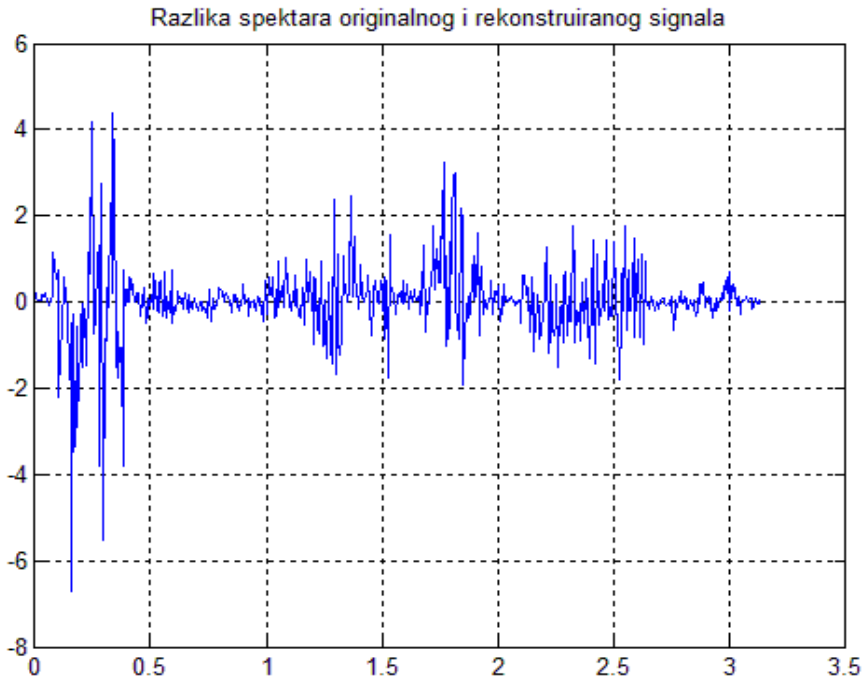
Za izvedbu Goertzlovog algoritma koristimo `goertzel` funkciju ugrađenu u MATLAB-ov Signal Processing Toolbox.

3.4. Rezultati i primjeri

U ovom potpoglavlju prikazati ćemo rezultate simulacija provedenih korištenjem funkcija opisanih u prošlim potpoglavljima. Dati ćemo primjere rada sustava na dva testna signala :

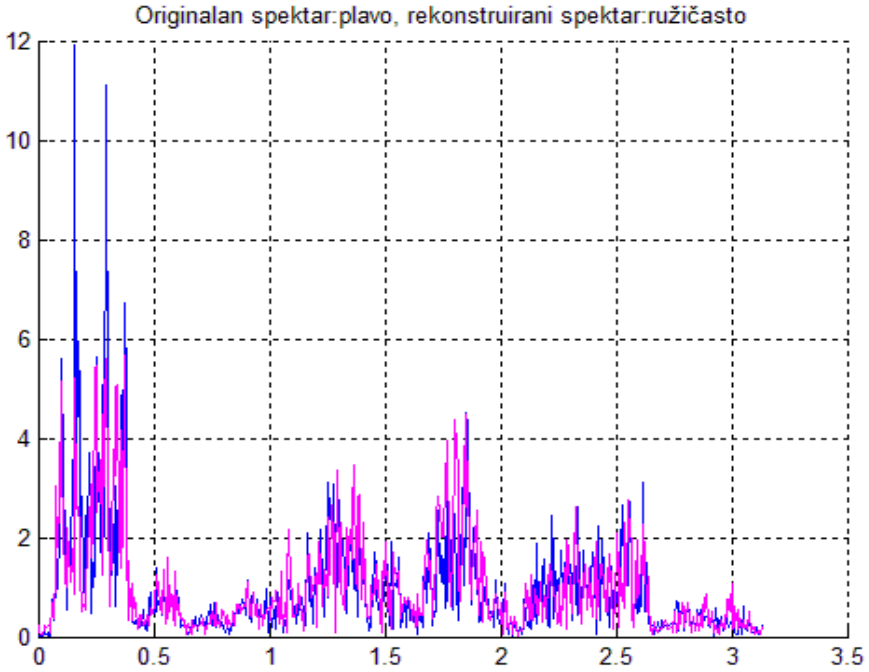
- test_signal_1.wav - primjer govora na hrvatskom jeziku, tekst govora "Ježurka ježić živi u šumi."
- test_signal_2.wav – primjer govora na engleskom jeziku, tekst govora "How much wood, would a woodchuck chuck, if a woodchuck would chuck wood."

Prvi primjer po količini izobličenja na izlazu dekodera spada u srednju kategoriju među ostalim testnim signalima, dok je test_signal_2.wav primjer koji je polučio najveća izobličenja od svih testnih signala sa gotovo 13 dB razlike na par frekvencija, te sa prosječnim izobličenjem od 5 dB na par frekvencijskih područja.



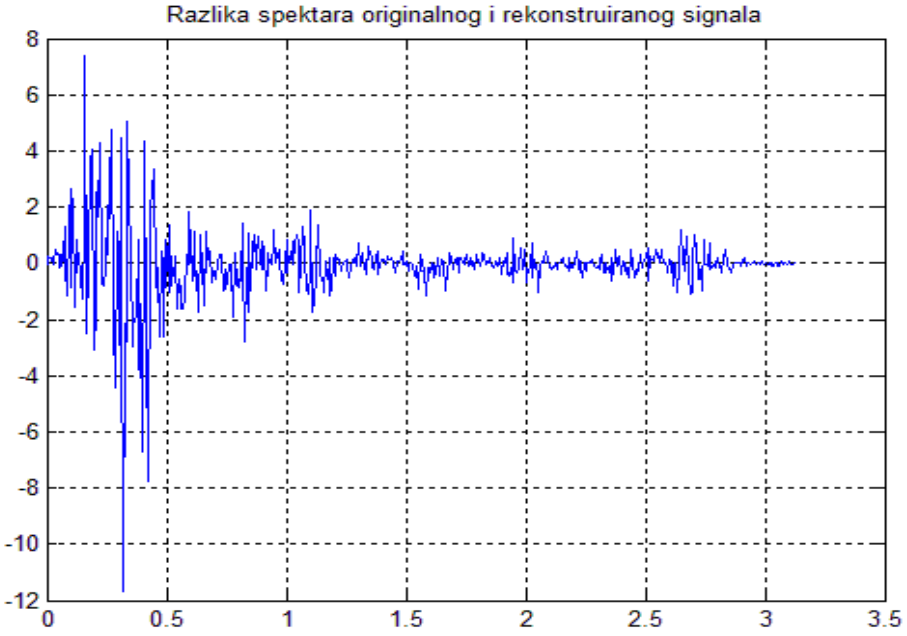
Slika 3.9 : Razlika spektara testnog signal i dekodiranog signala na izlazu sustava

Slika 3.10 prikazuje spektre originalnog testnog signala test_signal_1.wav, te signala na izlazu sustava, a slika 3.9 prikazuje razliku spektara ta 2 signala.

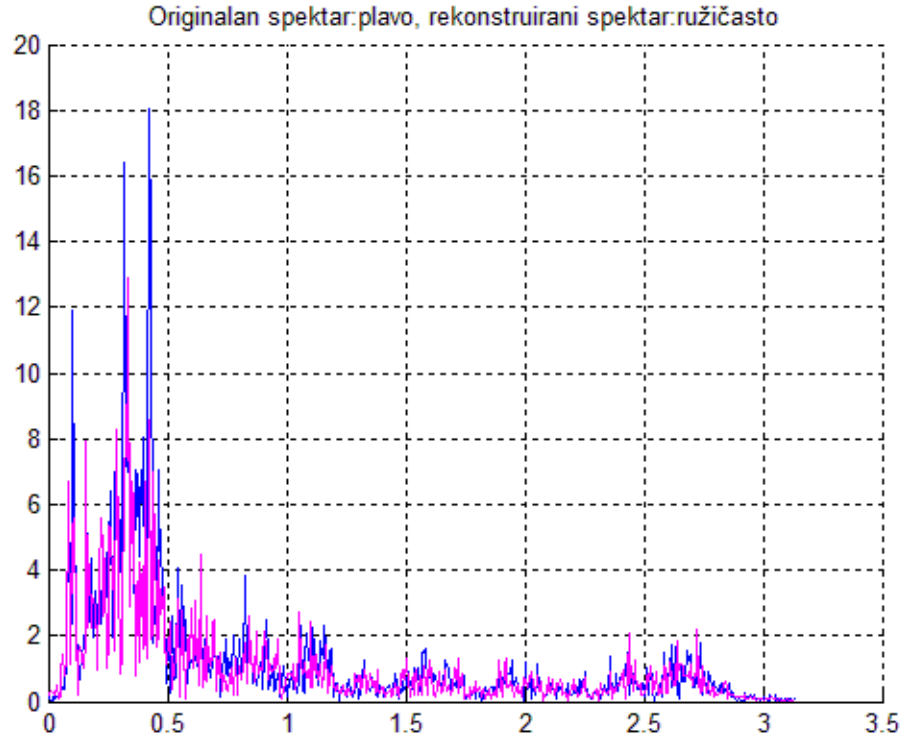


Slika 3.10 : Spektar testnog signala u plavoj boji i dekodiranog signala na izlazu u ružičastoj boji

Subjektivni slušni opis izlaznog signala za test_signal_1 je da je rekonstruirani signal na izlazu razumljiv, te da je govor prepoznatljiv, tj. da se prepoznaje specifičan govor osobe koja govori. U pozadini se čuju različite smetnje koje neutječu na razumljivost govora.



Slika 3. 11: Razlika spektra testnog signala i dekodiranog signala na izlazu sustava



Slika 3.12 : Spektar test_signal_2 signala i dekodiranog signala na izlazu sustava

Slika 3.12 prikazuje spektre originalnog testnog signala test_signal_2.wav, te signala na izlazu sustave, a slika 3.11 prikazuje razliku spektara ta 2 signala. Kao i kod prvog testnog signala, govor na izlazu dekodera je razumljiv, ali su sintetizirane smetnje jače.

Oba testna signala, te njihove kodirane i dekodirane verzije nalaze se u privitku.

Treba primjetiti da postoje tri veličine kojima se utječe na rad i rezultate algoritma:

- red FIR filtara u pseudoQMF slogovima
- broj FIR filtara u pseudoQMF slogovima
- broj blokova na koje se dijele odzivi analizirajućih slogova

Povećanjem bilo koje od te tri vrijednosti povećavamo vrijeme vrijeme izvođenja algoritma, te moramo izabrati optimalne vrijednosti. Povećanjem reda filtara smanjujemo grešku, a povećanjem broja potpojasnih komponenti u rastvu pojačavamo kriptološku snagu algoritma, pa za praktičnu primjenu preporučamo 8 filtara u slogu, te 8*16=128 red filtara da postignemo što manju grešku. Povećanjem broja blokova na koje razlažemo odzive analizirajućih slogova u koderu, povećavamo grešku, te izabiremo brojku od 16 blokova, koja je u različitim mjerenjima i na različitim testnim signalima polučila najbolje rezultate, tj. najmanju prosječnu grešku.

4. Implementacija sustava korištenjem programskog jezika C, te na TMS320VC5505 eZdsp evaluacijskom modulu

4.1. ANSI C izvedba

ANSI C izvedbu provodimo isključivo radi lakše pretvorbe MATLAB koda u C kod iskoristiv na TMS320VC5505 ezDSP modulu, te kao međukorak između simulacije i implementacije sustava. Potrebne parametre sustava tj, koeficijente FIR filtara u filtarskim slogovima spremamo u header datoteke, a testne signale, u MATLAB-u spremamo u tekstualne datoteke iz kojih C može čitati brojeve. Rezultati se na kraju spremaju u tekstualne datoteke iz kojih MATLAB može čitati polja brojeva i verificirati da li izvedba u C-u odgovara onoj u MATLAB-u.

Izveden je samo koder, te se njegov kod može naći u privitku.

4.2. Izvedba na TMS320VC5505 eZdsp modulu

Pri izvedbi bilo koje funkcionalnosti na DSP modulima postoji određen broj razlika naspram klasične ANSI C izvedbe.

Prva razlika je u tome što je TMS320VC5505 "fixed point" procesor, tj. nije optimiziran za rad sa "floating point" brojevima. Compiler i debuger u CCS programskoj podršci mogu interpretirati ANSI C kod koji koristi float varijable za izvedbu na procesoru, ali promjene koje mora izvesti pri prilagodbi koda mogu značajno utjecati na efikasnost i vrijeme izvođenja programa. Radi toga je bitno program preraditi da radi sa “fixed point“ varijablama.

Najlakši način da to postignemo je sve koeficijente FIR filtara, a i ostale brojeve s kojima operiramo, prebaciti u cijele vrijednosti, s tim da njihov apsolutne vrijednost nisu toliko bitne, koliko su odnosi između vrijedosti varijabli. Int16 vrsta varijable, pretpostavlja cijeli broj sa predznakom spremljen u 16 bita. 1 bit za predznak i 15 bitova za vrijednost može prikazati brojeve u intervalu od -32768 do 32767. Radi toga sve brojeve koje koristimo množimo sa 32767, te zaokružujemo na najbližu cijelobrojnu vrijednost. Množenjem sa 32767 omogućavamo najveću moguću razliku između dva broja koja su reprezentirana na ovaj način, čime maksimiziramo ralučivost koju gubimo prelasko sa float varijabli.

Druga razlika je vezana za samu funkcionalnost koda. Pri ANSI C simulaciji nemoramo razmišljati o samom učitavanju podataka, te njihovom prikazu na izlazu. Testne signale, koeficijente FIR filtara i izlaze sve čitamo ili pišemo iz raznih headera ili datoteka. Kod

DSP izvedbe, očitavamo prave vrijednosti signala, te ih na izlazu moramo pripremiti za reprezentaciju na zvučniku ili nekom spremištu poput CD-a ili flash memorije.

Na TMS320VC5505 eZdsp evaluacijskom modulu, za svaku od potrebnih funkcija, dakle, A/D pretvorbu, prijenos podataka od/do procesora, te D/A pretvorbu koristi se ugrađeni AIC3204 audio kodek. AIC3204 je modul sa programibilnim stereo ulazima i izlazima, te A/D i D/A pretvornicima sa brzinama do 48 ksampl-a/s.

Za pravilan rad sustava moramo inicijalizirati analogne ulaze i izlaze, I2C i I2S protokole za prijenos podataka između ulaza/izlaza i kodeka, te kodeka i procesora. Dodatno, moramo postaviti funkciju koja stereo ulaze pretvaraju u mono ulaze, funkcije koje inicijaliziraju PLL, te programabilni clock za AIC3204 kodek.

Za to koristimo funkcije već pripremljene od strane proizvođača, te su predefinirane u slijedećim header datotekama :

- `usbstk5505.h`
- `aic3204.h`
- `PLL.h`
- `stereo.h`

Funkcije se mogu koristiti u svojoj originalnoj formi ili kao u ovom radu modificirane uz priznavanje svih autorskih prava Texas Instruments Inc., te je u zaglavljima svih korištenih datoteka ostavljen *"copyright notice"*.

Za primjer osnovnog modela kodera, očitavamo određeni broj uzoraka sa analognih ulaza korištenjem funkcije `aic3204_codec_read.c`. Očitavamo samo 2000 uzoraka radi ograničenosti memorije na modulu. Očitane uzorke konvoluiramo sa koeficijentima filtra sadržanima u header datoteci `filtri.h` koristeći funkciju `FIR_filters_asm.c`. Funkcija je pisana u assembleru radi bržeg rada i velikog broja množenja s akumulacijom koja su potrebna za operaciju linearne konvolucije. Korištenje assemblera nam omogućuje višestruko ubrzanje radi mogućnosti korištenja `MACM` (multiply and accumulate) koja koristi sklopovska množila unutar samog modula i po specifikacijama omogućava do 200 milijuna množenja u sekundi. Odzivi se tada ispremještaju kao i kod simulacije, korištenjem poznatih ključeva sadžanih u datoteci `filtri.h`. Sintetizirajući slog dobijemo kao i analizirajući, koristeći `FIR_filters_asm.c`, te se dobiveni signal dovodi na zvučnički izlaz koristeći funkciju `aic3204_codec_read.c`.

Zaključak

Radom je prikazana mogućnost izvedbe jednostavnog sustava za zaštitu govorne informacije korištenjem pseudo kvadraturnih zrcalnih filtarskih slogova. Rezultati dobiveni simulacijom u programskom okružju MATLAB pokazuju da se sustavom dobivaju zaštićeni signali koji su izobličeni i u vremenskoj i frekvencijskoj domeni što daje jaču kriptografsku zaštitu od klasičnih metoda koje koriste izobličenja ili u vremenskoj ili frekvencijskoj domeni da bi signale učinili neprepoznatljivima.

S druge strane, rad prezentira teoretsku osnovu i neke od elemenata potrebnih za praktičnu izvedbu. Prvi problem koji treba riješiti da bi se omogućila izvedba sustava za rad u stvarnom vremenu je problem sinkronizacije između kodera i dekoder, tj. prijamnika i odašiljača. Rješenje ponuđeno ovim radom koristi Goertzelov algoritam da bi izračunalo fazu kodiranog signala iz faza dvaju frekvencija kojima je kodirani signal moduliran. Rješenje treba implementirati na sustav nakon što se sustav izvede.

Daljnji rad je potreban da bi se koder i dekoder izveli na TMS320VC5505 eZdsp modulu, te da bi se prikazao pravilan rad sustava ne samo u simulacijama, nego i u praktičnoj primjeni.

Literatura

[1] Vaidyanathan, P.“ Quadrature Mirror Filter Banks, M-Band Extensions and Perfect-Reconstruction Techniques“, IEEE ASSP Magazine, July 1987., strane 4-19

[2] Creuser C.D., Mitra S.K.,“A Simple Method for designing High-Quality Prototype Filters for M-Band Pseudo QMF Banks“, IEEE Transactions on Signal Processing, Vol. 43, NO.4, strane 1005-1007, travanj 1995.

[3] Bregović R., “Projektiranje digitalnih filtara za zrcalne filtarske slogove“, magistarski rad, FER, 1998

[4] Cox R.V., Tribolet J.M., "Analog voice privacy systems using TFSP Scrambling: Full Duplex and Half Duplex", The Bell System Tehnical Journal, Vol. 62, No.1, siječanj 1983, strane 47-61

[5] Creusere C.D., "Multirate filter banks and their use in communications systems", RTO Lecture Series, "Application of Mathematical Signal Processing Techniques to Mission Systems", Koln, Njemačka, 1999

[6] Apolinario J.A., Petraglia M.R., Alves R.G., "On Perfect Reconstruction in Critically Sampled Frequency Domain Scrambler",Instituto Militar de Engenharia Depto Eng Eletrica , Brazil

[7] Gunter Schmer, "DTMF Tone Generation and Detection: An Implementation Using the TMS320C54x", Application Report, SPRA096A - May 2000

[8] Brandau, M., "Implementation of a real-time voice encryption system", magistarski rad, Universitat Politècnica de Catalunya, 2008

Naslov, sažetak, ključne riječi

Naslov : Sustav zaštite govorne informacije

Sažetak : Radom je prikazana simulacija sustava za zaštitu govorne informacije. Sustav je baziran na izokretanju odziva pojedinih filtara u analizirajućem slogu pseudoQMF filtarskom slogu. Signal promijenjen i u vremenskoj i frekvencijskoj domeni dobije se rekonstrukcijom signala kroz sintetizirajući slog pseudoQMF-a. Dekodiranje je izvedeno inverznim postupkom rastava signala, te ponovnim izokretanjem signala po poznatom ključu. Koeficijenti filtara su dobiveni Creusere-Mitra algoritmom za projektiranje pseudoQMF filtarskih slogova. Simulacija je provedena u MATLAB programskom okružju, te su radom prikazani njeni rezultati u obliku razlika frekvencijskih spektara testnih signala i dekodiranih signala. Nadalje, rad postavlja određene uvjete za implementaciju sustava na TMS320VC5505 eZdsp modulu.

Ključne riječi : kodiranje govora, zaštita govora, filterski slogovi, pseudoQMF, TMS320VC5505, izokretanje, FIR filtri

Title, abstract, keywords

Title : Voice scrambler

Abstract : The theses shows a simulation of a voice encryption system. The system is based on scrambling the filter responses of an analysis bank of a pseudoQMF bank. A time and frequency scrambled signal is gained by reconstruction of the signal through the synthesis bank. A decoded signal is obtained by the same procedure. The coded speech is decomposed by an analasys filter bank and the filter responses are descrambled by a given permutation key. Filter coefficients are designed with the Creusere-Mitra algorithm for designing optimal pseudoQMF banks. The simulation is conducted using the MATLAB programing language, and the thesis shows its results as frequency spectrum differences of the original test signals and the decoded signals . Further, the thesis sets certain basis for the implementation of the system on the TMS320VC5505 eZdsp evaluation module.

Keywords : Speech coding, speech encryption, filter banks, pseudoQMF, TMS320VC5505, scrambling, FIR filters

Privitak

Svi dokumenti navedeni u privitku mogu se naći na CD-u priloženom uz rad :

- Privitak A : Kodovi MATLAB simulacijskog sustava
- Privitak B : Testni signali i primjeri njihova kodiranja, te dekodiranja
- Privitak C : ANSI C izvedba kodera
- Privitak D : Sve projektne datoteke potrebne za daljnji rad na sustavu na TMS320VC5505 eZdsp evaluacijskom modulu