

Elektrotehnički fakultet

Univerzitet u Banjoj Luci

**IZVJEŠTAJ PROJEKTNOG ZADATKA**

iz predmeta

**Sistemi ZA DIGITALNU OBRADU SIGNALA**

Student: Mentori:

*Tripić Nemanja, 11124/18* prof. dr Mladen Knežić

prof. dr Mitar Simić

dipl. inž. Damjan Prerad

ma Vedran Jovanović

Januar, Februar 2024. godine

# Opis projektnog zadatka

U sklopu projektnog zadatka potrebno je realizovati sistem za dodavanje muzičkih gitarskih efekata u audio signal. Za realizaciju ovog sistema potrebno je koristiti razvojno okruženje *ADSP*-21489 kroz programski paket *CrossCore Embedded Studio* pomoću kojeg se kreira projekat, zatim piše i kompajlira kod, a zatim i spušta na razvojnu ploču. Pisanje koda za ovo razvojno okruženje podrazumijeva korištenje programskog jezika *C*. Osim toga potrebno je sve efekte koji se izaberu, kao i same audio signale, realizovati u programskom jeziku *Python,* a to u svrhu poređenja rezultata dobijenih na dva načina realizacije radi profilisanja koda, te mjerenja performansi.

U nastavku će biti pobrojani neki od najviše korištenih audio efekata u današnjoj muzičkoj industriji, podijeljeni u nekoliko kategorija, prema načinu obrade:

* filtriranje: nisko-visokopropusni filtri, ekvilajzer,
* vremenski promjenljivi filtri: *wah-wah*, *phaser*,
* kašnjenje: vibrato, *flanger*, *chorus*, *echo*, *delay*,
* modulatori: ring modulacija, *tremolo*,
* nelinearna obrada: kompresija, limiter, distorzija, *noise* *gate*,
* specijalni efekti: *panning, reverb*, *surround, pitch shifter, rotary speaker, simulation…*

Kako bi se projektni zadatak smatrao uspješno odrađenim, potrebno je izabrati minimalno tri audio efekta, te ih realizovati na gore pomenuti način. Efekti mogu biti izabrani proizvoljno, ili po preporuci neki od efekata navedenih u tekstu projektnog zadatka, podijeljeni u grupe po težini, pri čemu bar jedan efekat treba biti izvan Grupe 1 sa manje zahtjevnim filtrima.

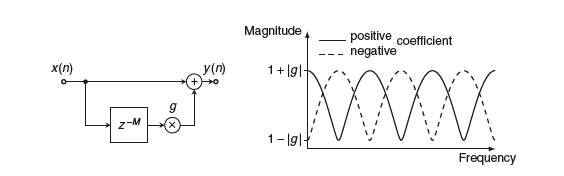
Osnovna ideja projektnog zadatka je da se za početak efekti **realizuju u *Python-u****,* zatim na ***ADSP*** **procesoru**. Nakon toga se **radi analiza performansi** algoritama implementiranih na ADSP procesoru, tj. brzina izvršavanja, zauzeće memorije i slično, te na osnovu toga izvrši neka optimizacija. Sljedeći korak je da se uradi **validacija rezultata** poređenjem sa rezultatima dobijenim u *Python-u.* Poslednji korak je **korekcija implementacije** algoritma na *DSP*.

# Izrada projektnog zadatka

Pri izradi projektnog zadatka izabrani su sljedeći efekti: *delay, distortion, wah-wah, phaser, reverb.* U narednim pasusima biće ukratko teorijski opisan svaki od efekata, uz to će biti dati detalji realizacije, objašnjenja korištenih parametara, osim toga biće priložen kod koji će biti propisno dokumentovan

# *Delay* (kašnjenje) efekat

Ovaj audio efekat spada u grupu efekata zasnovanih na kašnjenju, odatle i njegov naziv. Zapravo to je jedan od najjednostavnijih i u suštini osnovni efekat iz ove grupe, većina drugih efekata je zasnovana na njemu. Jako je sličan još jednom efektu iz ove grupe pod nazivom eho, pa se ova dva naziva često koriste i kao sinonimi. . Postoji nekoliko mogućih realizacija ovog efekta. Može se koristiti FIR filtar, kao i IIR filtar. U konkretnom slučaju ovog projektnog zadatka koristićemo FIR filtar. Realizacija pomoću IIR filtr može se naći u referenciranoj literaturi [1][2]. Blok šema kojom se može jednostavno opisati audio efekat kašnjenja data je na sljedećoj slici:



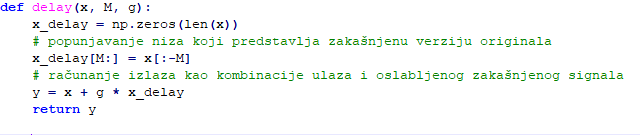
Slika 2.1.1 – *Blok šema flitra kojim se realizuje efekat kašnjenja i amplitudski spektar*

Sa blok šeme jasno vidi kako se izlazni signal dobija kao suma originalnog signala i originalnog signala zakašnjenog za *M* odmjeraka i skaliranog koeficijentom *g*. Na slici 2.1.1 sa desne strane nalazi se amplitudski spektar filtra korištenog za realizaciju efekta kašnjenja, i na njemu se primjećuju neki zanimljivi oblici. Naime amplitudski spektar filtra ima oblik češlja, pa se prema tome ovaj filtar u literaturi često naziva češljasti filtar [1][2]. Iz blok šeme dobija se jednačina diferencija na osnovu koje se onda programski može jednostavno realizovati efekat:

(2.1.1)

Za realizaciju efekta, kao što se vidi iz jednačine diferencija, bitna su dva parametra, a to su *M* i *g*. Parametar *M* označava broj odmjeraka za koji je signal zakašnjen iz relacije *M = t \* Fs* , gdje je *t* vrijeme kašnjenja signala u sekundama, a *Fs* frekvencija odmjeravanja, dok je *g* faktor pojačanja zakašnjelog signala, tj. često se kaže odnos amplituda reflektovanog i direktnog signala. Obično uzima vrijednosti između 0 i 1.

Implementacija je data na sljedećoj slici:



Slika 2.1.2 Implementacija efekta kašnjenja u pajtonu

# Distortion(distorzija) efekat

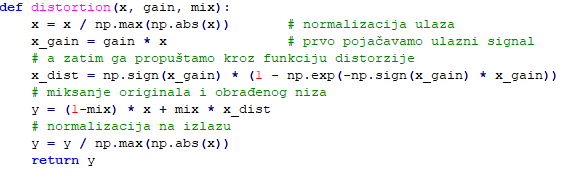
Efekat distorzije spada u grupu efekata sa nelinearnom obradom. Termin nelinearna obrada podrazumijeva sve algoritme obrade signala koji ne zadovoljavaju princip linearnosti. Linearnost je nešto što se često spominje u kontekstu obrade signala, i jedan je od najbitnijih koncepata u ovoj oblasti. Većina algoritama za klasičnu obradu signala su linearni. Međutim, kao što se može vidjeti i na primjeru ovog efekta, postoje neki algoritmi koji nisu linearni, a koji nam daju zanimljive rezultate. U tu grupu spada dosta muzičkih efekata kao što je ovaj. Ono što rade nelinearni algoritmi obrade, jeste da unose dodatne spektralne frekvencijske komponente kojih nema u originalnom signalu. A za slučaj audio signala, to znači da će se promijeniti boja zvuka, što slušalac može jasno da čuje.

Što se tiče konkretno efekta distorzije, ovaj efekat kada se pojavio postao je okosnica rok muzike i uopšte zvuka gitare od tada do danas. Ranije ovaj efekat je kreiran analogno, raznim pedalama i slično, ali pojavom digitalne obrade signala brzo se došlo do algoritama kojima se to može uraditi na računaru. U suštini ono što distorzija radi ulaznom signalu jeste da ga nelinearno transformiše tzv. nelinearnom krivom pojačanja. Konkretno kod efekta distorzije ta kriva je obično neka verzija eksponencijalne krive i postoji nekoliko funkcija kojim se može izvršiti distorzija. Jedna od najčešće korišćenih i ona koja je korištena u ovom zadatku je sljedeća:

(2.2.1)

Osnovni parametri distorzije su pojačanje i miks. Prije primjene funkcije iznad, na ulazni signal, potrebno ga je pojačati određenim pojačanjem. Određivanje tog pojačanja tako da se na izlazu dobije željeni zvuk je posao muzičkih inženjera i tonaca, i njihov posao je da nađu optimalno pojačanje tako da se dobije baš onakav zvuk na izlazu kakav je u konkretnom slučaju idealan.Što se tiče miksa to je obično drugi podesivi parametar distorzije i on određuje koliko će u rezultatnom signalu biti udjela originalnog signala, a koliko obrađenog, po principu *(1 – mix) \* original + mix \* obrađeni*. Dakle miks ima vrijednost između 0 i 1. Često prije obrade radi normalizacija signala, kao i normalizacija konačnog obrađenog signala, da bi se izbjegla dodatna izobličenja uzrokovana odsjecanjem audio signala, zbog toga što izlazi izvan opsega vrijednosti koje je moguće fizički reprodukovati na zvučniku.

Implementacija efekta data je sljedećom slikom:



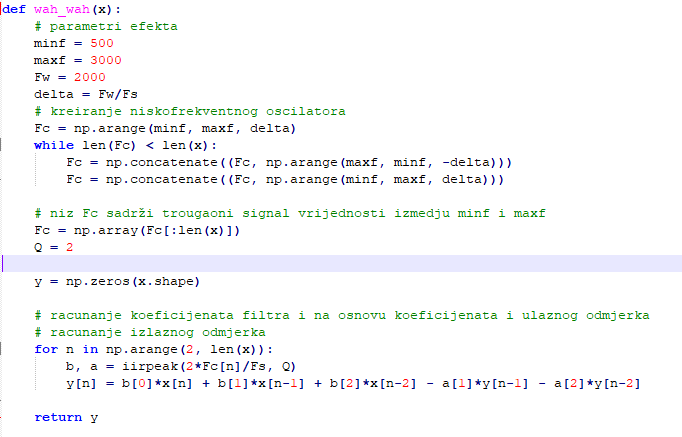
Slika 2.2.1 Implementacija efekta distorzije u pajtonu

# *Wah-Wah* (vah-vah) efekat

Vah-vah efekat spada u grupu efekata sa vremenski promjenljivim filtrima. Efekti iz ove grupe dobijaju se tako što se u toku obrade signala mijenjaju parametri filtra. Karakterističan primjer efekta iz ove grupe jeste vah-vah efekat. Slično kao i distorzija to je efekat jako često korišten za obradu zvuka električne gitare u rok muzici. U prošlosti efekat je obično kreiran pomoću nožne pedale sa analognim filtrima untar nje, a koja je spojena na gitaru, te pritiskom na nju u određenom trenutku i sa određenim intenzitetom stvara se efekat zvuka koji podsjeća na izgovaranje riječi vah-vah. Naravno kreiranje ovog efekta moguće je uraditi i digitalno na računaru.

Za realizaciju efekta koristi se filtar propusnik opsega sa veoma uskim propusnim opsegom, koji u idealnom slučaju propušta samo komponente jako bliske tzv. centralnoj frekvenciji ili čak samo koponente na toj centralnoj frekvenciji, ostale komponente idealno poništava ili barem dovoljno oslabi. Ovakav filtar često se zove i pik filtar (eng. *peak*). Riječ pik se prevodi kao vrh, špic, a naziv je otuda što amplitudska karakteristika ima izražen vrh u području oko centralne frekvencije. Zatim, da bi se na osnovu ovog filtra kreirao vah-vah efekat, potrebno je „šetati“ centralnu frekvenciju po frekvencijskoj osi i provlačiti ulazni signal kroz te filtre i slati ih na izlaz. To šetanje frekvencije, kao što je već spomenuto, može se kontrolisati pedalom, ali može se i kreirati niskofrekventni oscilator koji će kontrolisati centralnu frekvenciju.

Konkretno u našemu slučaju kao niskofrekventni oscilator kreira se povorka trougaonih signala, dužine jednake dužini ulaznog niza odmjeraka. Taj trougaoni signal ima vrijednosti između minimalne i maksimalne frekvencije koju može da poprimi centralna frekvencija, u kodu označene sa *minf* i *maxf* i one predstavljaju parametre efekta. Zatim kao parametar imamo frekvenciju označenu sa *Fw* i to je tzv. vah frekvencija koja određuje frekvenciju oscilatora. Nakon toga za svaki odmjerak na osnovu trenutne centralne frekvencije računaju se koeficijenti pik filtra, te se računa izlazni odmjerak na osnovu odgovarajućeg ulaznog odmjerka. Kod u Python-u dat je u nastavku:



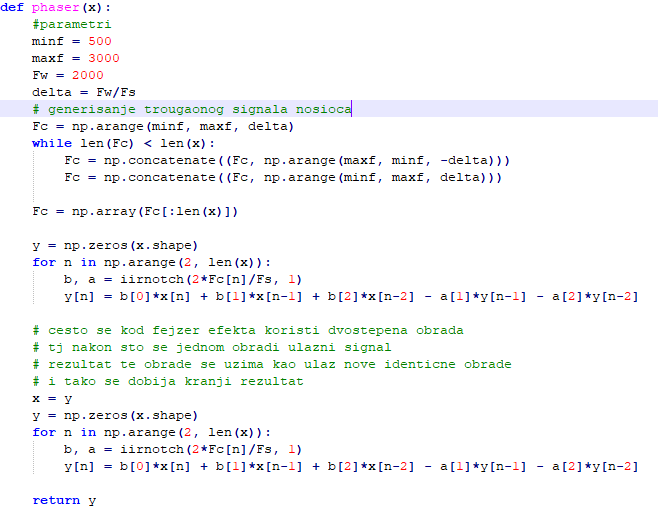
Slika 2.3.1 Implementacija vah-vah efekta u *python-*u

# 2.4 *Phaser* (fejzer) efekat

Pripada istoj grupi efekata kao i vah-vah efekat, i u suštini jako su slični, i na neki način inverzni jedan u odnosu na drugog. Princip funkcionisanja je isti, postoji centralna frekvencija koja je promjenljiva i na osnovu koje se kreiraju filtri, kroz koje se onda propušta ulazni signal. Razlika je u tome što se kod fejzer efekta, umjesto filtra propusnika opsega, koristi filtar nepropusnik opsega sa centralnom frekvencijom i jako uskim opsegom ili tzv. noč (eng. *notch)* filtar. To je filtar koji, idealno, propušta sve komponente frekvencija osim jedne, a to je komponenta centralne frekvencije, koju jako slabi.

U mojoj realizaciji sve ostaje isto kao kod vah-vah efekta, samo se umjesto pik filtra koristi noč filtar, koji također postoji u *scipy* biblioteci, kao *iirnotch*()

Na sljedećoj slici je realizacija fejzer efekta u pajtonu:



Slika 2.4.1 Implementacija fejzer efekta u pajtonu

# 2.5 *Reverberation* (reverberacije) efekat

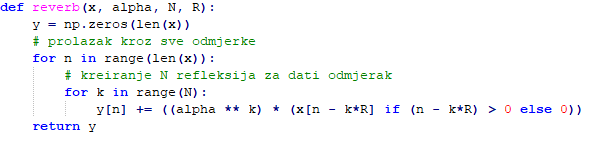
Efekat reverberacije u tekstu projektnog zadatka smješten je u grupu specijalnih efekata, a često se u literaturi smješta i u grupu prostornih efekata [1]. Prostorni efekti su efekti koji ulaznom audio signalu, nakon obrade dodaju elemente koje slušaoc doživljava na način da ima osjećaj da je zvuk bogatiji, puniji i najbitnije da dolazi iz nekoliko izvora zvuka sa različitih strana. U suštini ovi efekti simuliraju puštanje audio signala u zatvorenoj prostoriji sa jednog ili više zvučnik. Ljudski slušni sistem je koncipiran tako da način obrade pristiglog zvuka zavisi od smjera iz kojeg je zvuk pristigao i tako čovjek dobija predstavu o izvoru zvuka u prostoru. Jedan od najpoznatijih efekata iz ove grupe je efekat poznat pod nazivom reverberacije. Kada se zvuk prostire od izvora do slušaoca on do njega može stići putem velikog broja različitih putanja, naročito ako se slušanje odvija u zatvorenom prostoru. Postoji naravno direktni, najbliži put, a zatim i sve one putanje na kojima se zvuka odbija o površinu predmeta u prostoriji i tako stiže do slušaoca. Reflektovani talasi će biti oslabljeni i zakašnjeni u odnosu na signal koji je prvi stigao do slušaoca. Svi ovi talasi se međusobno sabiraju i tako daju zvuku taj prostorni efekat i nije moguće jasno razlikovati reflektovane talase od onih koji su stigli direkno.

Reverberacije se mogu realizovati na dva osnovna načina: pomoću filtara i linija za kašnjenje, te drugi način pomoću konvolucije sa impulsnim odzivom prostorije. Ja sam koristio prvi način sa filtrima i linijama za kašnjenje. Jednostavna realizacija je korištenjem FIR filtra koji se može opisati sljedećom jednačinom diferencija:

(2.5.1)

Na ovaj način simulira se prvo direktni put talasa od izvora ka slušaocu, a zatim i višestruke refleksije talasa koje pristižu sa određenim kašnjenjem i oslabljene do slušaoca. Svaka refleksija koja je više zakašnjena, tj. kasnije stiže do slušaoca, je više oslabljena zbog stepenovanja parametra .Parametar R određuje nakon koliko vremena se pojavljuje prva refleksija, tj. koliko je zakašnjena. A parametar N određuje koliko refleksija će se desiti za konkretni odmjerak. Ovi parametri su jasno fiksni za sve odmjerke, iako realno oni mogu biti promjenljivi, ali to i jeste glavni nedostatak ovog načina realizacije reverberacija.

Realizacija efekta u pajtonu je data na sljedećoj slici:



Slika 2.5.1 Implementacija reverberacija u pajtonu

Jednačine se u izvještaj dodaju uvijek u novom redu i numerišu se u formatu (*x.y*), pri čemu *x* označava poglavlje a *y* predstavlja redni broj jednačine u poglavlju. Nakon čega se jednačine u tekstu mogu referencirati referenciranjem dodijeljenog broja. Jednačine se u tekst ne smiju dodavati kao slike.

(2.1)

Primjer referenciranja jednačine u tekstu: “Jednačina 2.1 predstavlja formulu za izračunavanje diskretne Furijeove transformacije – DFT-a.”

# Zaključak

# Literatura

[1] Vladimir Risojević. *Multimedijalni sistemi*. Univerzitet u Banjoj Luci, Elektrotehnički fakultet, 2018.

[2] Udo Zolzer. DAFX: *Digital Audio Effects.* John Willey and Sons Ltd, 2011.