

Elektrotehnički fakultet

Univerzitet u Banjoj Luci

**IZVJEŠTAJ PROJEKTNOG ZADATKA**

iz predmeta

**SISTEMI ZA DIGITALNU OBRADU SIGNALA**

Student: Mentori:

*Emanuela Buganik 1209/18* prof. dr Mladen Knežić

prof. dr Mitar Simić

ma Vedran Jovanović

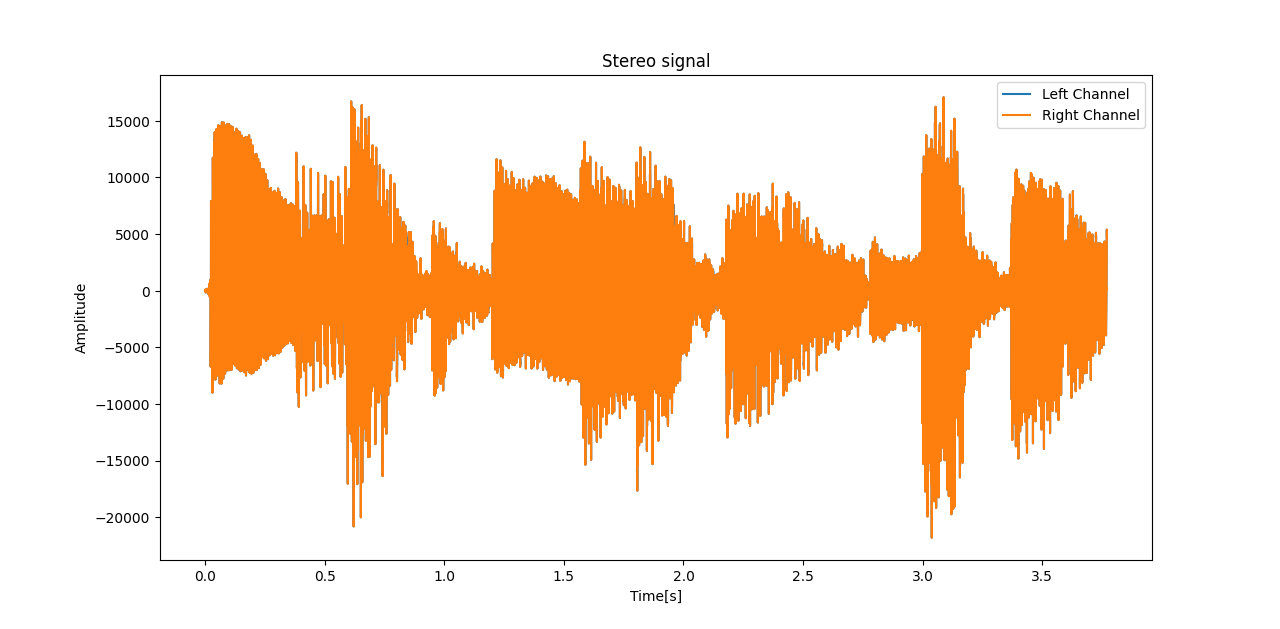
dipl. inž. Damjan Prerad

Februar 2024. godine

# Opis projektnog zadatka

Cilj izrade projektnog zadatka je implementacija muzičkih efekata koji se mogu primijeniti na audio signal, korištenjem razvojnog okruženja ADSP-21489. Prvi dio izrade, ujedno i referenca za dalji rad odnosi se na generisanje referentnog signala i implementaciju gitarskih efekata (iz grupa definisanih projektnim zadatkom ili proizvoljno), a u Python programskom jeziku. Drugi dio izrade odnosi se na implementaciju istih efekata, ali u C programskom jeziku i unutar *CrossCore Embedded Studio* programskog paketa za ciljanu platformu. Odmjerci signala koji je korišten za analizu u prvom dijelu izrade upisuju se u file koji se zatim učitava u CCES kao *header* file i omogućava obradu istih. Moguće je reprodukovati rezultate obrade signala i u jednom i u drugom slučaju, eksportovati kao *.wav* file, a vršeno je i profilisanje koda s ciljem da se uoče potencijalni *bottleneck*-ovi i ukoliko je moguće, izvrši optimizacija koda na DSP-u. Efekti o kojima će biti riječ u narednom poglavlju su efekti iz prve i druge grupe - manje zahtijevni filtri *Delay* i *Echo* te umjereno zahtijevni *Tremolo*, *Flanger* i *Bit Crusher*.

# Izrada projektnog zadatka

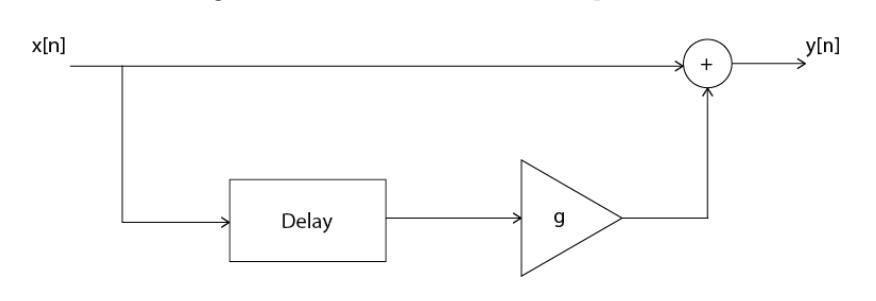
S obzirom na primjenu efekata koje je potrebno implementirati, kao audio signal koji će se koristiti za obradu u daljem radu koristi se stereo zvuk gitare u trajanju od 3,77 s, a čija je frekvencija odmjeravanja prvobitno 44 100 Hz. Jedna od opcija da se analizira stereo signal jeste izdvajanje kanala, primjena efekata, a zatim ponovno objedinjavanje u 2D niz. Međutim, kako je prikazano i na narednoj *Slici 1.1*

*Slika* *2.1* Stereo signal

da se zapaziti kako nema razlike između lijevog i desnog kanala, te se postupak daljeg rada pojednostavljuje i tako radi sa mono signalom koji je nastao usrednjavanjem kanala stereo signala. Pored ovog razmatranja, u dalji rad se polazi idući koji korak unaprijed, pa se razmatra i kako uvesti odmjerke ovog sada mono signala u CCES, imajući na umu da se radi o signalu sa pomenutom frekvencijom odmjeravanja, gdje je broj odmjeraka 166 267. Postavlja se pitanje ima li smisla raditi sa velikim brojem odmjeraka kada radimo sa ograničenim signalom, uzimajući u obzir i da je potrebno omogućiti i učitavanje odmjeraka (pitanje skalabilnosti unutar *header*-a?), smještanje u memoriju, ispis obrađenih odmjeraka u *file* i slično. Sljedeći metod koji se može iskoristiti da pojednostavimo rad jeste *decimacija* (eng. downsampling), kojom za neki cjelobrojni faktor D možemo da smanjimo kako frekvenciju odmjeravanja nekog već odmjerenog signala, tako i njegov broj odmjeraka. U ovom slučaju izvršena je decimacija sa faktorom četiri, čime je nova frekvencija odmjeravanja 11 025 Hz, a broj odmjeraka 41 567. Korištena je python funkcija decimate, unutar koje je spesifikovan i tip niskopropusnog filtra u vidu Hamming prozora. Naredna slika prikazuje promjenu, tačnije mono signal prije i nakon decimacije.

DODATI SLIKU MONO I DECIMATED MONO

*Slika 2.2* Mono signal po usrednjavanju i nakon decimacije

*Delay* efekat, kao prvi efekat koji je obrađen realizuje se korištenjem FIR filtra, kako se efekat zakašnjelog signala implementira sabiranjem originalnog signala sa svojom pomjerenom, zakašnjelom verzijom, koja je pomnožena faktorom g, a što je prikazano i na blok dijagramu na *Slici 2.3*. Naivna implementacija u Pythonu prikazuje kako se pomjerena verzija signala može ostvariti pomjeranjem originalnog signala za broj odmjeraka koji određuju kašnjenje.

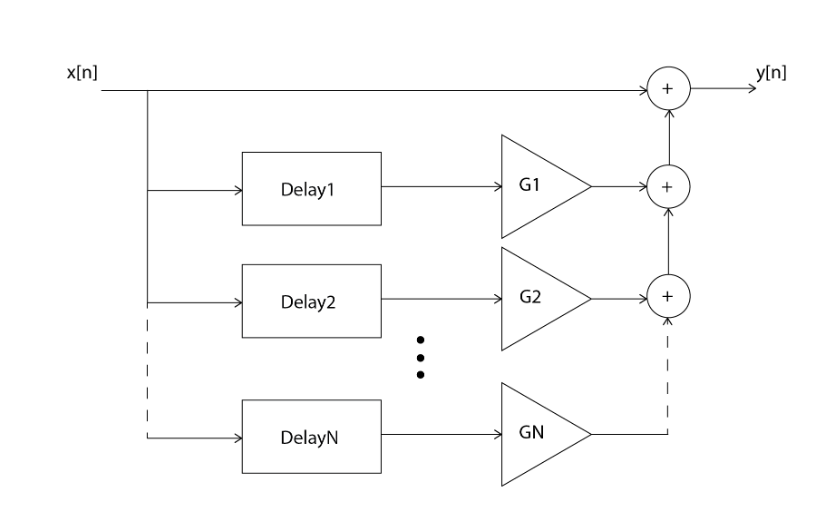
*Slika 2.3* Blok dijagram *Delay* efekta

U ovom pa i narednim primjerima realizacije efekata, vrijednost faktora g je između -1 i 1, broj odmjeraka za pomjeraj (određuje se kao umnožak frekvencije odmjeravanja i vremena kašnjenja u sekundama.

(2.1)

Ovdje dodati:

* Sliku pomjeraja u odnosu na original u pythonu
* Sliku pomjeraja python vs CCES
* Slika signala greške

*Echo* efekat, sa druge strane, predstavlja višestruki efekat kašnjenja koji se može realizovati kroz kašnjenje na više linija ili *stage*-ova, kako je prikazano i na narednoj slici:

*Slika 2.4* Blok dijagram *Echo* efekta

Kako se da i primijetiti, *Echo* efekat se sastoji od dodavanja originalnog audio signala N broju zakašnjelih verzija istog, gdje amplituda (, gdje je i = 1, 2, 3, …) opada sa svakom linijom kašnjenja. Očito je da je za slučaj N=1, u pitanju gorepomenuti slučaj *Delay* efekta.

(2.2)

Ovdje dodati:

* Sliku pomjeraja u odnosu na original u pythonu
* Sliku pomjeraja python vs CCES
* Slika signala greške

*Tremolo* efekat iz druge grupe efekata realizuje se kao vid amplitudske modulacije, što znači da se amplituda ulaznog signala (eng. carrier) mijenja sa promjenama LFO (oscilatora na niskim frekvencijama). Narednom jednačinom to se može prikazati na sljedeći način:

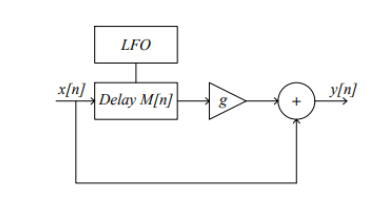
(2.3)

Gdje je n)

*Depth* određuje amplitudu signala modulatora, uopšte – dubinu amplitudske modulacije, a frekvenciju modulatora (obično u opsegu od 1 Hz do 25 Hz).

Ovdje dodati:

* Sliku tremola u odnosu na original u pythonu
* Sliku tremola python vs CCES
* Slika signala greške

Osnovna ideja iza *Flanger* efekta je stvaranje konstruktivne i destruktivne interferencije, dodavanjem originalnog zvuka zakašnjeloj verziji zvuka, čije je kašnjenje određeno funkcijom vremena, a što prikazuje i sljedeći blok dijagram:

*Slika 2.5* Blok dijagram *Flanger* efekta

Pomenuta funkcija vremena je u ovom slučaju LFO, kao u primjeru realizacije efekta *Tremolo*, a čija frekvencija modulatora iznosi između 0.1 Hz do 10 Hz.

Ovdje dodati:

* Sliku tremola u odnosu na original u pythonu
* Sliku tremola python vs CCES
* Slika signala greške

Posljednji implementiran efekat je *Bit Crusher*, poznatiji i kao *lo-fi* (eng. low-fidelity) efekat. Sa opcijama kao što su smanjenje frekvencije odmjeravanja (poznato i kao eng. „downsampling“ i „rate crush“) te smanjenje rezolucije (poznato i kao eng. „bit depth“ i „bit crush“), ovim efektom se postiže ciljana distorzija audio zapisa, što je u ovom radu i implementirano.

# Zaključak

# Literatura