ADOZ projekat [kt1]

Petar Marković RA 112-2021

28. November 2023

1 Uvod - okruženje

Ovaj dokument svedoči o mentalnim mukama, upsonima, padovima i ponovnim usponima kao i o fizičkoj iscrpljenosti prouzrokovane samom izradom projekta iz ADOZ-a.

Najpre sam pokušao odraditi projekat u Code Composer Studio, te sam zbog toga morao instalirati CCS. Nažalost izgleda da ne postoji način da se taj program pokrene na Linuxu (Fedora), što znači da čak i ozloglašeni program za emulaciju "Wine"nije bio od pomoći.

Rešenje sam našao tako što sam u VirtualBoxu napravio virtuelnu mašinu Windowsa i u njoj skinuo sve potrebne alate. No ipak na kraju, odlučio sam se da koristim Python, no više o tome u nastavku.

Na ulaznom signalu se čuje zemljotres, sinusni signal velike amplitude, i pištaljka (najverovatnije sastavljena od još jednog takvog sinusa više frekvencije i par slabiji harmonika)

2 Prvi koraci - Audacity

Analizom ulaznog signala pomoću programa Audacity, metodom slušanja su određene sve frekvencije od interesa. Na spektogramu i analizi frekvencije se vide jasna dva peak-a koji predstavljaju ometačke signale (šum).

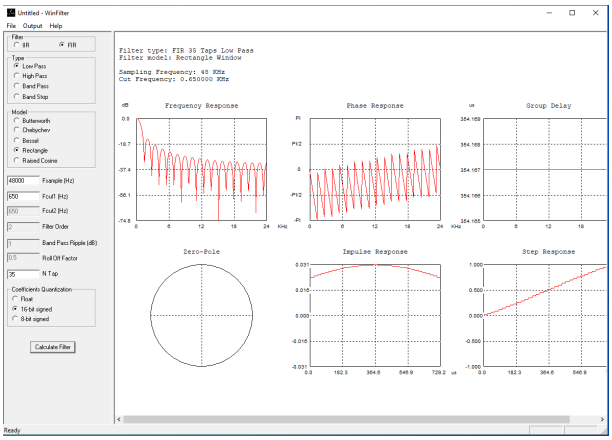
Oba ova signala (sinusa) se mogu otkloniti sa band stop filtrom, međutim zbog ličnih preferenci izabrana su dva filtra. Prvi je low pass filtar koji propušta frekvencije ispod 650hz, a drugi je notch (band stop) filtar koji iz ima parametre stop_freq=300 Q=30.

Ispostavilo se da ovo ne obuhvata celokupni korisni signal

3 Code Composer Studio 6 + WinFilter

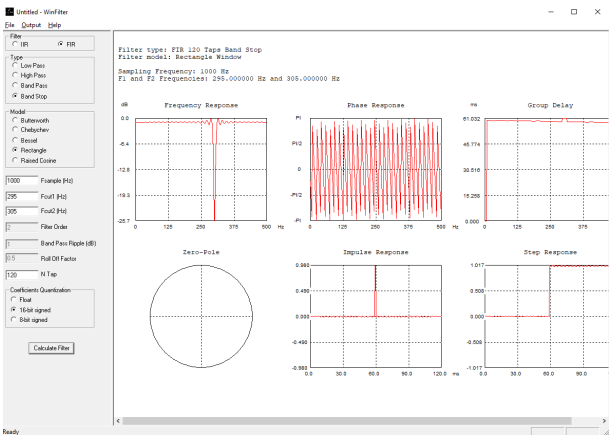
Iskorišćena je postavka iz 7. vežbi. Po uputstvu koje je došlo uz postavku problema generisana su dva filtra na ranije spomenutim frekvencijama:

Niskopropusni (low pass) filtar koji propušta frekvencije ispod 650hz:



Slika 1: WinFilter: Low pass filtar

I filtar koji zaustavlja opseg (band stop - notch) koji blokira harmonik od 300hz:



Slika 2: WinFilter: Band stop filtar

Nakon toga, oba filtra su eksportovana i ubačena u projekat na sličan način rađen na prethodnim vežbama. Međutim (nažalost), uprkos svim naporima autora kod nije mogao da se builduje (sistem prijavljuje grešku da ne postoji odgovarajući kompajler. Ovo, kao i još par neprijatnosti čiji je izvor samo radno okruženje odveli su projekat u novo, bolje, jednostavnije radno okruženje opisano u narednoj sekciji.

4 Python

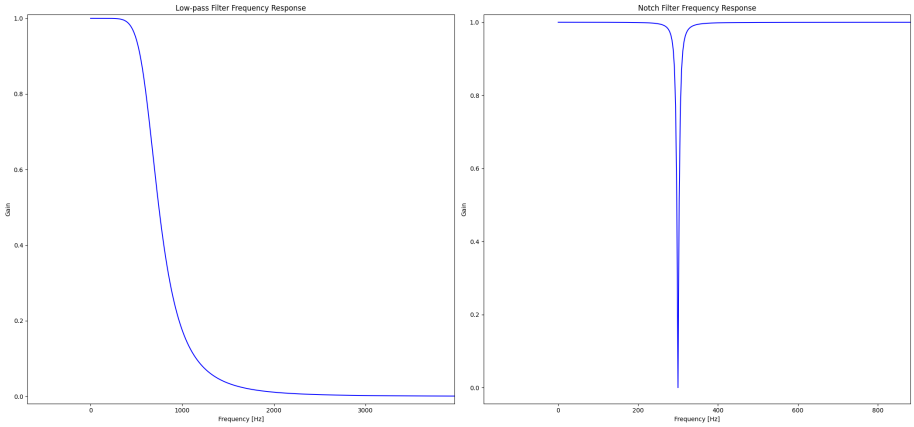
Python poseduje veliki broj biblioteka optimizovanih obostrano za dobre rezultate ali i lako korišćenje u gotovo svim inženjerskim oblastima interesovanja. Ni DSP nije zaboravljen, tako da postoji mnoštvo korisnih biblioteka od kojih su neke korišćene u ovom kodu:

4.1 Opis korišćenih biblioteka

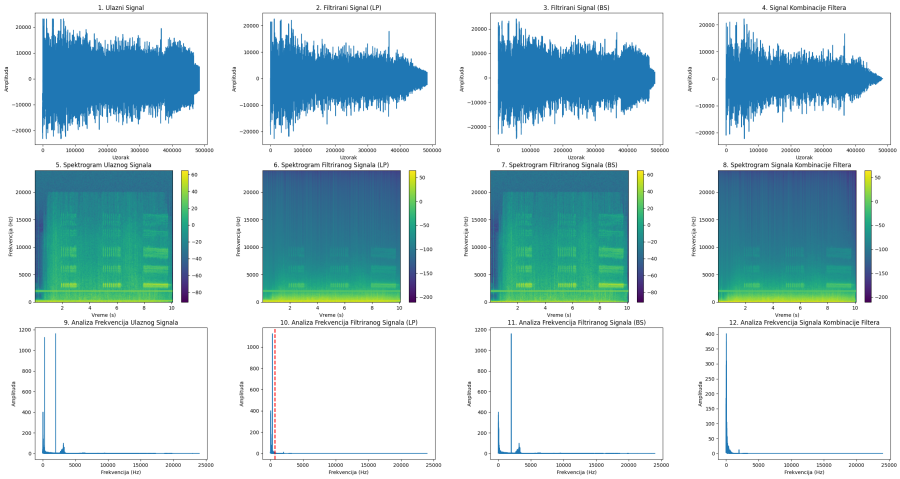
- **Matplotlib:**
  - Svestrana biblioteka za crtanje grafika u Python-u.
  - Omogućava kreiranje statičkih, animiranih i interaktivnih vizualizacija.
  - Pruža interfejs sličan MATLAB-u za generisanje visokokvalitetnih grafika.
  - Široko se koristi u oblastima poput data science, mašinskog učenja, inženjeringa i naučnih istraživanja.
- **NumPy:**
  - Moćna numerička biblioteka za Python.
  - Pruža podršku za velike, višedimenzionalne nizove i matrice.
  - Obezbeđuje kolekciju matematičkih funkcija za manipulisanje numeričkim podacima.
  - Osnovna biblioteka za numeričko i naučno računanje u Python-u.
- **SciPy:**
  - Open-source biblioteka zasnovana na NumPy-u.
  - Pruža dodatnu funkcionalnost za naučno i tehničko računanje.
  - Uključuje module za optimizaciju, integraciju, interpolaciju, obradu signala i slika, linearnu algebru, statističke operacije, itd.
  - Dopunjuje NumPy, pružajući sveobuhvatan skup alatki za naučna istraživanja i inženjerske primene.

4.2 Kod

Baza koda, svih figura u ovom dokumentu kao i rezultujućih signala mogu se naći na repozitorijumu `https://github.com/petarpetarpetar/DSP1_projekat`



Slika 3: Matplotlib: Frekvencijski odzivi filtara



Slika 4: WinFilter: Svi spektogrami i frekvencijske analize