LABORATORIJSKA VAJA 1

Namen vaje je seznaniti študente s postopki predstavitve zvočnih signalov v obliki spektrogramov z uporabo kratkočasovne Fourierove analize in okenskih funkcij, ter s pomenom in analizo signalov v tej predstavitvi.

Za izvedbo vaje pridejo v poštev funkcije programskih knjižnic numpy, scipy in matplotlib.pyplot, ki so dokumentirane na:

https://docs.scipy.org/doc/numpy/reference/

https://docs.scipy.org/doc/scipy/reference/

https://matplotlib.org/api/pyplot_api.html

Dokumentacija vseh funkcij je dostopna tudi iz interaktivega interpreterja Python, preko ukaza help(ime_funkcije). Za namen snemanja lastnih govornih posnetkov in grafično visualizacijo posnetkov uporabljamo še prosto dostopen program Audacity, ki je na voljo iz spletne strani https://www.audacityteam.org/, kjer se nahaja tudi njegova dokumentacija.

Naloga 1 (1 točka)

S pomočjo programa Audacity posnemite zvočni posnetek glasno in razločno izgovorjenega stavka:

„Nikóli in nikdár ne pústi pêti níz pôsla nekjé v vŕsti"

Posnetek naj vsebuje približno 500ms začetnega in končnega premora. Posnetek shranite v zvočno datoteko „govor.wav" v WAV formatu, pri čemer uporabite frekvnenco vzorčenja 16kHz in 32-biten zapis vzorcev.

V primeru, da ne uspete pridobiti primernega govornega posnetka, za nadaljnje naloge uporabite posnetek govor.wav, ki se nahaja v gradivu za vajo.

V zapisu 16kHz mono wav. S pomočjo grafičnih predstavitev Določite časovne odseke, na katerih se v vašem posnetku pojavi vsak izmed samoglasnikov slovenskega jezika.

Izgovorjeni stavek vsebuje vseh osem naglašenih samoglasnikov slovenskega govorjenega jezika, ki jih v računalniški različici mednarodne fonetične abecede IPA zapišemo (po vrsti pojavljanja v stavku na podčrtanih mestih) s simboli o, a, u, E, i, O, e, in @.

V pomočjo grafičnih predstavitev s programom audacity ugotovite in si zabeležite časovne intervale v govornem signalu, kjer je izvedena akustična uresničitev omenjenih osmih samoglasnikov.

Naloga 2 (2 točki)

Z uporabo funkcije scipy.io.wavfile.read naložite svoj posnetek v python. Iz naloženih podatkovnih struktur določite

- Vzorčno frekvenco
- Dolžino posnetka v sekundah
- Podatkovni tip in zalogo vrednosti vzorcev

S funkcijo matplotlib.pyplot.specgram z ustrezno izbiro parametrov izrišite dva različna spektrograma vašega posnetka, in sicer

- Širokopasovni spektrogram, na katerem je viden časovni potek resonance glasilk, in
- Ozkopasovni spektrogram, na katerem je viden potek formantnih frekvenc in njihovih harmoničnih mnogokratnikov.

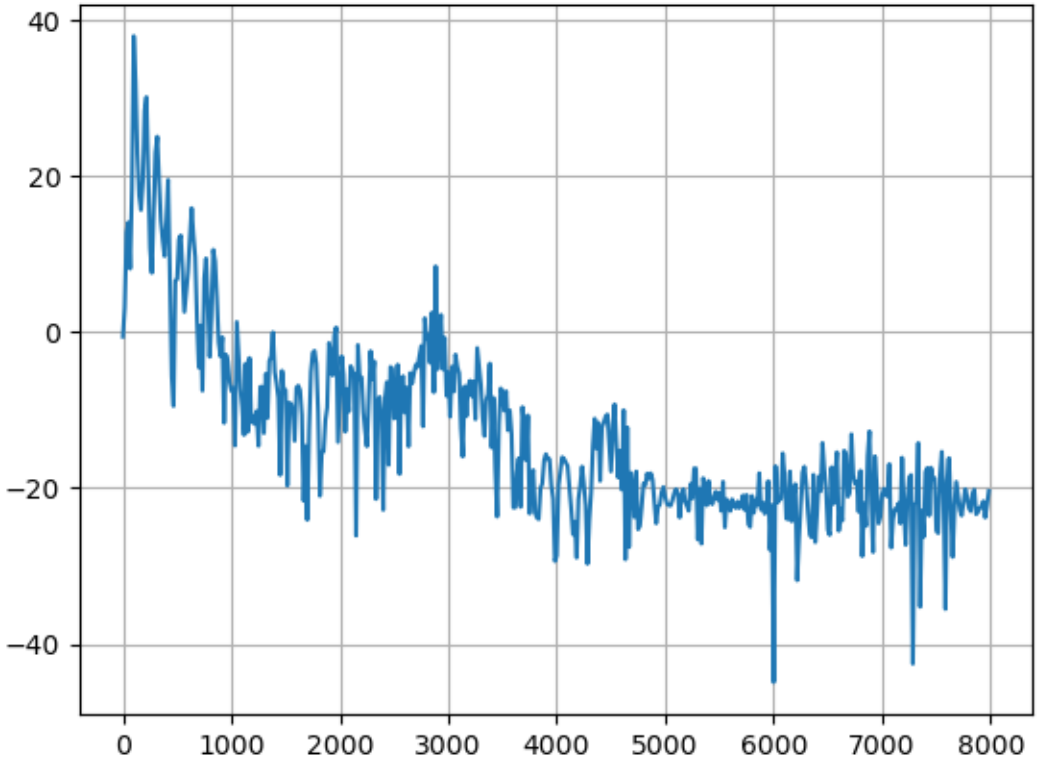
Naloga 3 (2 točki)

Zabeležene časovne intervale izgovorjave posameznih samoglasnikov iz prve naloge pretvorite v začetne in končne indekse vzorcev govornega signala za posamezen samoglasnik. Iz močnostnega spektra izseka govornega signala, kjer se pojavljajo samoglasniki razberite prvi dve formantni frekvenci (F1 in F2) vsakega izmed samoglasnikov.

Za izračun spektra se zgledujte po sledeči python skripti:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io.wavfile import read

fs, signal = read("govor.wav")


t0 = 0.7
t1 = 0.76

# iz časovnega intervala izračunaj začetni in končni indeks izseka
#i0 = ...
#i1 = ...

i = signal[i0 : i1]
NFFT = i.shape[0]
I = np.fft.fft(i)[:len(i)//2]
f_os = np.arange(0, fs/2, fs/NFFT)
P = 20 * np.log10(np.abs(I))
plt.plot(f_os, P)
plt.grid(True)
plt.show()
```

Primer močnostnega spektra enega izmed samoglasnikov je podan na spodnji sliki:



Pri čemer frekvenčna vrhova pri 150Hz in 2800Hz predstavljata formantni frekvenci F1 in F2.

Ugotovljene vrednosti formantnih frekvenc F1 in F2 za vseh osem samoglasnikov nato vrišite v graf vrednosti formantnih frekvenc F1/F2, kjer se vrednosti za F1 raztezajo od zgoraj navzdol od 0 do 100 Hz in vrednosti za F2 od leve proti desni od 0 do 3000 Hz.

Grafični prikaz takšnega diagrama lahko dobimo s sledečo python kodo:

```
import matplotlib.pyplot as plt
import numpy as np

F1 = np.array([100, 200, 300, 400, 500, 600, 700, 800])
F2 = 3 * F1

for i in range(8):
    plt.plot(F2[i], F1[i], "o")
plt.xlim((0, 3000))
plt.ylim((1000, 0))
plt.xlabel("F2 [Hz]")
plt.ylabel("F1 [Hz]")

plt.legend("o a u E i O e @".split(" "))

plt.show()
```