LABORATORIJSKA VAJA 2

IDEJA VAJE

Namen vaje je seznaniti študente z računskimi postopki, ki omogočajo sintezo (umetno tvorjenje) zvočnih govornih signalov. Vaja vsebuje **tri naloge**, ki so podane v nadaljevanju.

Za izvedbo vaje pridejo v poštev python funkcije, ki so del knjižnic *numpy in scipy*, kot tudi funkcije v datoteki *lpc.py*, ki je priložena materialu za vaje.

TEORETIČNO OZADJE

Pri 1. laboratorijski vaji smo se spoznali s spektralno analizo preko diskretne fourierove transformacije in spektrogramov. Ugotovili smo, da *zveneči glasovi* slovenskega jezika (t.j., samoglasniki in zvočni soglasniki) vsebujejo manjše število *formantov* – harmoničnih elementov, ki jih v spektrogramih opazimo kot svetlejše črte na tenmem ozadju.

Če za posamezen glas dovolj natančno poznamo frekvence in amplitude njegovih formantov, ta glas lahko umetno tvorimo kot sinusno vrsto z ustreznimi amplitudami in frekvencami, tako, da definiramo signal

$$y(t) = amp_1 \sin(2\pi f_1) + amp_2 \sin(2\pi f_2) + \cdots + amp_n \sin(2\pi f_n)$$

Pri čemer amp_i ter f_i predstavljata amplitudo oz. frekvenco i-tega formanta za dani glas. Če poznamo formante za vse glasove, ki sestavljajo besedilo, ki ga želimo tvoriti, to lahko storimo tako, da tvorimo signal po krajših izsekih, na katerih nastopajo ustrezni formanti, in jih nato lepimo skupaj v ustreznem zaporedju.

V okviru te vaje bomo spoznali LPC analizo, ki nam omogoča samodejno določitev frekvenc in amplitud formantov zvenečega govora.

Linearno prediktivno kodiranje (angl. *Linear Predictive Coding*, LPC), je model, ki nam omogoča napovedovanje prihodnjega vzorca signala na podlagi prejšnjih. Naj bo $x(t)$ signal, ki ga obravnavamo. Predikcijo trenutnega vzorca predstavimo kot problem

$$x(t) = a_1 x(t-1) + a_2 x(t-2) + \cdots + a_n x(t-n),$$

Pri čemer n predstavlja red LPC sistema in členi a_n predstavljajo njegove koeficiente, ki jih moramo določiti. Koeficiente sistema LPC je možno določiti preko regresije najmanjših kvadratov. Naj bo okno signala $x(t)$, ki ga obravnavamo, dolžine N in mnogo daljše od reda n sistema LPC $(N \gg n)$. Na podlagi okna signala pripravimo regresijsko matriko oblike $(N - n - 1) \times n$,

$$\boldsymbol{X} = \begin{bmatrix} x(1) & \cdots & x(n) \\ x(2) & \cdots & x(n+1) \\ \vdots & \ddots & \vdots \\ x(N-n-1) & \dots & x(N-1) \end{bmatrix},$$

Ter vektor regresorjev

$$\boldsymbol{y} = [x(n+1), x(n+2), \dots, x(N)]^T,$$

Koeficiente sistema LPC zdaj določa tisti vektor $\boldsymbol{a}$, ki matrično enačno $\boldsymbol{Xa} = \boldsymbol{y}$ reši v smislu najmanjših kvadratov, t.j.,

$$\hat{\boldsymbol{a}} = \min_{\boldsymbol{a}} \|\boldsymbol{X}\boldsymbol{a} - \boldsymbol{y}\|$$

Dobljeni koeficienti LPC sistema predstavljajo prenosno funkcijo filtra, ki tvori zveneče glasove pri govoru – t.j., glasilk in ustne votline. Kompleksne ničle prenosne funkcije so določene z enačbo

$$z^n a_n + z^{n-1} a_{n-1} + \cdots + z a_1 = 0$$

In jih je možno določiti s standardnimi iterativnimi numeričnimi metodami za iskanje približkov ničel polinomskih funkcij (gl. funkcijo *np.roots*). Naj bodo ničle sistema LPC predstavljene v vektorju kompleksnih števil $\boldsymbol{k}$ (v izogib dvoumnosti s spremenljivko z; za kompleksne ničle polinomskih funkcij se namreč uporablja tudi izraz *koreni*). Frekvence formantov zvenečih glasov, ki se nahajajo v oknu signala x so tedaj določene kot kompleksni argument posameznih ničel, ustrezno preskalirani glede na znano frekvenco vzorčenja signala x, f_s:

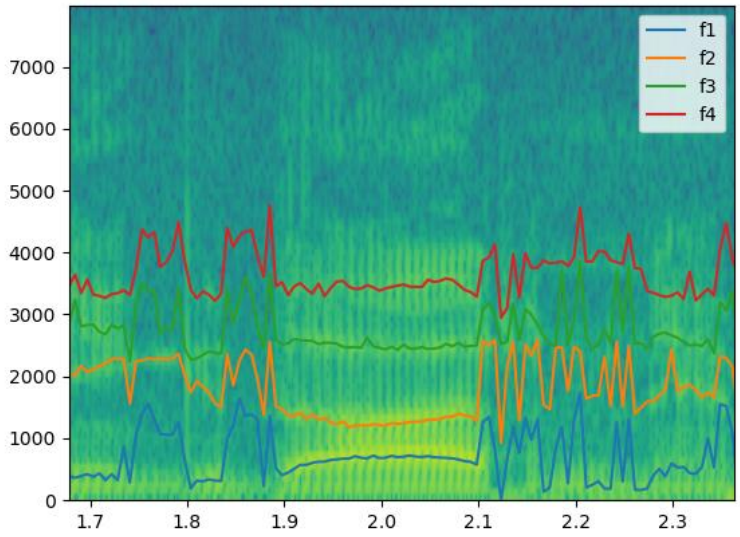
$$f_i = \frac{f_s}{2\pi} Arg(k_i) = \frac{f_s}{2\pi} atg\left(\frac{Im(k_i)}{Re(k_i)}\right)$$

Iz ničel LPC sistema lahko določimo tudi pasovne širine formantov, na podlagi njihovih absolutnih vrednosti,

$$w_i = |k_i|$$

Iz pasovne širine pa amplitudo formantov določimo kot $amp_i = e^{\frac{w_i}{60}}$.

Na spodnji sliki je prikazan potek frekvenc formantov $f_1 - f_4$, določenih preko LPC analize, v ozadju širokopasovnega spektrograma govornega posnetka. Vidno je, da se pri zvenečem govoru na tak način določene frekvence formantov ujemajo s frekvenčnim potekom na spektrogramu.



Naloga 1 (1 točka)

Spišite funkcijo ‚sinsum', ki tvori signal kot uteženo vsoto sinusnih signalov izbranih frekvenc in amplitud po spodnjem izrazu,

$$s(n) = \sum_{k=1}^{M} A_k \sin(2\pi n f_k), \qquad n = 0, \dots, N-1,$$

pri čemer f_s označuje izbrano frekvenco vzorčenja v Hz, A_k izbrano amplitudo posamezne sinusne komponente in f_k njeno izbrano frekvenco v Hz. Delovanje funkcije preverite s podano skripto:

```
import sys, os
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import wavfile

def sinsum(f, a, T, fs):
    """Funkcija za sintezo sinusne vrste. Vhodni argumenti:
        f: seznam frekvenc frekvenčnih komponent sinusne vrste
        a: seznam ojačanj frekvenčnih komponent, v linearni skali
        T: dolžina sintetiziranega signala, v sekundah
        fs: željena vzorčna frekvenca sintetiziranega signala."""

    # definicija časovne osi, preko katere sintetiziramo signal
    t = np.arange(0, T, 1/fs).astype("float32")
    # inicializacija praznega arraya za sintetiziran signal
    signal = np.zeros_like(t)

    # TODO: implementiraj sintezo s sinusno vrsto

    # normalizacija signala na zalogo vrednosti [-1, 1]
    signal -= signal.mean()
    signal /= np.abs(signal).max()
    return signal

if __name__ == "__main__":
    x = sinsum([220, 440, 880, 1760],
               [1, 0.5, 0.25, 0.125],
               1.00002,
               44100)
    x_prav = np.load("sinsum_test.npy")
    # oblika dejanskega in pravilnega signala
    print(x.shape, x_prav.shape)
    # maksimalno odstopanje
    print(np.abs(x - x_prav).max())
```

Naloga 2 (1 točka)

S pomočjo funkcije ‘sinsum’ iz prejšnje naloge tvorite signal, sestavljen iz dveh frekvenčnih komponent, katerih amplitude in frekvence se spremenijo vsake pol sekunde. Amplitude in frekvence sinusnih komponent nastavite na vrednosti formantov petih samoglasnikov (‘a’, ‘E’, ‘I’, ‘O’, ‘u’) slovenskega govora. Frekvence in amplitude izmerite s pomočjo fourierove analize tako, kot ste to storili **pri prvi laboratorijski vaji**, oziroma uporabite meritve, ki ste jih pridobili pri omenjeni vaji.

Glede na to, da so pri spektralni analizi amplitude frekvenčnih komponent podane v decibelih, je potrebno funkciji sinsum podati amplitude, preslikane v linearno merilo, tako, da obrnete preslikavo

$$P[dB] = 20 \log_{10} P$$

Sintetizirane pol sekunde dolge signale, ki predstavljajo posamezne samoglasnike zlepite skupaj v en array in ga z uporabo funkcije scipy.io.wavfile.write zapišite v zvočno datoteko v zapisu WAV. Primerjajte rezultate svojih meritev s signalom, sintetiziranim z uporabo referenčne tabele formantov samoglasnikov:

Vowel (IPA)	Formant F_1 (Hz)	Formant F_2 (Hz)	Difference $F_1 - F_2$ (Hz)
i	240	2400	2160
y	235	2100	1865
e	390	2300	1910
ø	370	1900	1530
ɛ	610	1900	1290
œ	585	1710	1125
a	850	1610	760
œ	820	1530	710
ɑ	750	940	190
ɒ	700	760	60
ʌ	600	1170	570
ɔ	500	700	200
ɤ	460	1310	850
o	360	640	280
ɯ	300	1390	1090
u	250	595	345

Naloga 3 (3 točke)

Posnemite primer svojega govora, dolg nekaj sekund. Govor naj vsebuje čim več zvenečih glasov, kot npr. stavek za zgled s samoglasniki iz 1. vaje. Z uporabo pomožnih funkcij v datoteki *lpc.py* izvedite LPC analizo svojega posnetka ter določite časovni potek frekvenc in amplitud formantov. Posnetek shranite z vzorčno frekvenco 16kHz, LPC analizo pa izvedite na oknih dolžine 200 vzorcev, s prekrivanjem 100 vzorcev.

Na podlagi določenih formantov nato z uporabo funkcije *sinsum* iz 1. naloge tvorite umetni govorni posnetek. Izseki tvorjenega signala naj bodo iste dolžine, kot okna pri LPC analizi Sintetizirane izseke oknite s Hannovim oknom (gl. funkcijo np.hanning) in lepite z enakim prekrivanjem, kot je bilo uporabljeno pri LPC analizi. Tvorjen signal shranite v novo zvočno datoteko ter ga skupaj z izvornim zvočnim posnetkom priložite poročilu o vaji. Komentirajte razlike med izvornim in tvorjenim posnetkom.

Dodatni nasveti in navodila za izvedbo

- LPC analizo posameznih oken signala izvede funkcija *lpc_okno*, ki neposredno vrne frekvence in amplitude formantov v oknu.
- Pri LPC analiza in tvorjenju signala preko sinusnih vrst predpostavljamo, da je izvorni signal možno predstaviti v taki obliki, kar pa pri človeškem govoru drži zgolj za zveneče glasove. Zato na teh delih govora pričakujemo bolj kvalitetno sintezo od nezvenečih glasov. Pričakovan rezultat sinteze je prikazan na izvornem posnetku *govor.wav* in njegovi sintetizirani različici *govor_synth.wav*.
- Red LPC sistema nastavimo glede na pričakovano pasovno širino signala, ki ga analiziramo. Za človeški govor se predpostavlja 1 formant na kHz frekvenc, temu pa dodamo 2 reda, da zavzamemo še enosmerno komponento in vrh frekvenčnega spektra. Pri frekvenci vzorčenja 16kHz torej uporabljajte LPC sistem z 18 koeficienti.