

Signali in vzorčenje: razpoznavnik akordov

Ob koncu semestra se gotovo prileže malo glasbe. Osnovni gradnik glasbe je ton [1]. Vsak ton ima svoje ime in svojo frekvenco, npr. $A_1 = 440$ Hz. Ko poslušamo glasbo, največkrat poslušamo sozvočje večih tonov - če so ti toni trije ali jih je več, temu sozvočju pravimo akord [2]. V tokratni nalogi se bomo poglobili v spektralno analizo zvočnega signala in skušali razpoznati akorde, ki so morebiti prisotni v danih posnetkih.

Zaradi enostavnosti se bomo pri razpoznavanju omejili na akorde iz treh tonov, pri čemer vzamemo samo dve klavirski oktavi tonov: od C_1 do B_2 (oktava je interval 12 poltonov) [3]. Ton, ki je za oktavo višji od drugega tona, ima dvokratnik njegove frekvence. Primer: ton A_2 ima frekvenco 880 Hz, kar je dvakrat več kot A_1 , torej $2 * 440$ Hz. V spodnji tabeli so navedeni vsi toni in njihove frekvence, ki lahko sestavljajo iskane akorde.

C_1	CIS_1	D_1	DIS_1	E_1	F_1	FIS_1	G_1	GIS_1	A_1	B_1	H_1
261,63	277,18	293,66	311,13	329,63	349,23	369,99	392	415,30	440	466,16	493,88
C_2	CIS_2	D_2	DIS_2	E_2	F_2	FIS_2	G_2	GIS_2	A_2	B_2	H_2
523,25	554,37	587,33	622,25	659,25	698,46	739,99	783,99	830,61	880	932,33	987,77

Naslednja tabela vsebuje imena in sestavne tone vseh akordov, ki jih želimo iskati v zvočnem signalu.

Cdur	C_1	E_1	G_1
Cmol	C_1	DIS_1	G_1
Ddur	D_1	FIS_1	A_1
Dmol	D_1	F_1	A_1
Edur	E_1	GIS_1	H_1
Emol	E_1	G_1	H_1
Fdur	F_1	A_1	C_2
Fmol	F_1	GIS_1	C_2
Gdur	G_1	H_1	D_2
Gmol	G_1	B_1	D_2
Adur	A_1	CIS_2	E_2
Amol	A_1	C_2	E_2
Hdur	H_1	DIS_2	FIS_2
Hmol	H_1	D_2	FIS_2

Opomba: pri molovskih akordih je navada, da se srednji ton zapiše kot znižan in ne kot zvišan ton. Primer za C-mol: ton DIS_1 je za pol tona zvišan ton D_1 in je v glasbeni teoriji enak znižanemu E_1 , ki se mu reče ES_1 . Ker gre za tone z istimi frekvencami se s tem ne bomo obremenjevali.

Naloga

Vaša naloga je, da v Pythonu napišete funkcijo `naloga4`, ki bo razpoznavala akord v zvočnem posnetku. Funkcija prejme zvočni signal `vhod` in frekvenco vzorčenja `fs`. Razpoznavo

izvedete tako, da signal transformirate s pomočjo hitre Fourierove transformacije – uporabite lahko funkcijo `fft`, ki je del paketa `numpy.fft`. Na ta način signal iz časovnega prostora predstavite v frekvenčni prostor. Iz transformiranega signala je namreč mogoče razbrati, kateri toni (sinusni signali) nastopajo v posnetku in iz tega ugotoviti, ali sestavljajo katerega izmed iskanih akordov. Izhod funkcije `naloga4` je niz z imenom akorda, npr. `'Cdur'`. Če v zvočnem signalu `vhod` ni prisotnega nobenega akorda iz zgornje tabele, potem naj funkcija vrne prazen niz `''`. V vsakem posnetku je prisoten največ en akord, ki traja od začetka do konca signala. Pri razpoznavi akordov vam bo prav prišel izračun močnostnega spektra zvočnega posnetka. Upoštevajte, da je vhodni signal realen in je zato Fourierov transform simetričen glede na Nyquistovo frekvenco.

Zvočni posnetki lahko vsebujejo tudi “motilne” frekvence, ki pa niso enake frekvencam podanih tonov v zgornji tabeli (tudi niso enake frekvencam tonov v nižjih in višjih oktavah). Te je potrebno pri razpoznavanju ignorirati. Prav tako se lahko pojavijo tudi posnetki, ki ne vsebujejo vseh potrebnih tonov za določen akord ali pa sploh ne vsebujejo nobenega tona - v teh primerih je izhod vašega programa prazen niz.

Poseben primer predstavljajo posnetki, kjer so prisotni višjeharmonski ali alikvotni toni [4, 5] – to so celoštevilski večkratniki osnovnih tonov. Primer: akord C-dur sestavljajo toni C_1 , E_1 in G_1 . Če so poleg njih prisotni tudi kateri od njihovih večkratnikov, na primer C_2 , E_2 ali G_2 , C_3 , ... jih razumemo kot višje harmonike in jih ignoriramo – torej razpoznamo C-dur.

Na spletni učilnici oddajte datoteko `naloga4.py`, v kateri je implementirana zahtevana funkcija.

Prototip funkcije:

```
def naloga4(vhod: list, fs: int) -> str:
    """
    Poisce akord v zvocnem zapisu.

    Parameters
    -----
    vhod : list
        vhodni zvocni zapis
    fs : int
        frekvenca vzorcenja

    Returns
    -----
    izhod : str
        ime akorda, ki se skriva v zvocnem zapisu;
        ce frekvence v zvocnem zapisu ne ustrezajo nobenemu od navedenih
        akordov,
        vrnemo prazen niz ''
    """

    izhod = ''
    return izhod
```

Testni primeri

Na učilnici se nahaja arhiv `TIS-naloga4.zip`, ki vsebuje tri testne primere. Primeri so podani v obliki datotek `.json`. Priloženo imate tudi funkcijo `test_naloga4`, ki jo lahko uporabite za preverjanje pravilnosti rezultatov, ki jih vrača vaša funkcija.

Pri testiranju vaše funkcije upoštevajte naslednje:

- zvočni signal bo dolg največ 480000 vzorcev,
- pri posameznem testnem primeru dobite točko, če se `izhod` popolnoma ujema z rešitvijo, ki je bodisi ime akorda bodisi prazen niz,
- rešitve, kjer funkcija `naloga4` vedno vrača vnaprej določen rezultat, ki ni dobljen algoritmično (t.i. “hard-coded” rešitve), bodo ocenjene z 0 točkami,
- izvajanje funkcije je časovno omejeno na 10 sekund,
- dovoljena je uporaba standardne knjižnice Python 3.9 (<https://docs.python.org/3.9/library/>) in paketov `numpy` (priporočamo) ter `scipy`.

Literatura

- [1] Wikipedia: <https://sl.wikipedia.org/wiki/Ton>.
- [2] Wikipedia: <https://sl.wikipedia.org/wiki/Akord>.
- [3] Wikipedia: https://en.wikipedia.org/wiki/Piano_key_frequencies.
- [4] Wikipedia: https://sl.wikipedia.org/wiki/Alikvotni_toni.
- [5] Wikipedia: <https://en.wikipedia.org/wiki/Harmonic>.