

Signali in vzorčenje: razpoznavnik DTMF

DTMF (ang. *dual-tone multi-frequency signaling*) se uporablja za komuniciranje med telefonskimi napravami in centralo po analognih linijah [1]. Standard ni bil mišljen za izmenjavo podatkov, ampak samo za pošiljanje kontrolnih signalov. V klasičnih telefonskih napravah ga uporabljamo za izbiranje klicane številke. Pritisk na vsako izmed tipk na telefonski številčnici generira signal, ki je sestavljen iz dveh sinusnih komponent. Frekvence sinusnih signalov so preračunane tako, da jih je možno prenašati po standardnih telefonskih linijah in je število napak pri razpoznavanju generiranih tonov čim manjše. Spodnja tabela prikazuje preslikavo med tipkami in generiranimi sinusnimi signali:

f	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#

Zgornja tabela se ujema s postavitvijo telefonske matrične tipkovnice. Vrstica in stolpec, v kateri se pritisnjena tipka nahaja, povesta, kateri dve frekvenčni komponenti morata v signalu nastopati. Primer: pritisk na tipko 1 generira signal (ton) sestavljen iz dveh seštetih sinusoid, ene s frekvenco 697 Hz in druge s frekvenco 1209 Hz. Frekvence se nahajajo v območju, ki ga človeško uho lahko zazna, tako da ob pritiskanju tipk na telefonu slišimo generirane tone.

Naloga

Vaša naloga je, da v Pythonu napišete funkcijo `naloga4`, ki bo razpoznavala pritiske tipk v zvočnem posnetku. Funkcija prejme zvočni signal `vhod`, frekvenco vzorčenja `fs` in trajanje posameznega tona v sekundah `t`. Razpoznavo izvedete tako, da signal razdelite na intervale dolžine trajanja posameznega tona in vsakega transformirate s pomočjo hitre Fourierove transformacije – uporabite lahko funkcijo `fft`, ki je del paketa `numpy.fft`. Na ta način signal iz časovnega prostora prestavite v frekvenčni prostor. Iz transformiranega signala je namreč mogoče razbrati, katere frekvenčne komponente (sinusni signali) nastopajo v posnetku in iz tega ugotoviti zaporedje pritisnjenih tipk. Za posameznim tonom (pritiskom tipke) je vedno premor, ki je enak dolžini trajanja tona. Izhod funkcije `naloga4` je niz znakov, ki predstavlja zaporedje pritisnjenih tipk npr. `'12*#3'`. Če v nekem delu zvočnega signala `vhod` ni prisotne ustrezne kombinacije frekvenc iz zgornje tabele, potem ta del signala ignorirate. Pri razpoznavi pritiskov tipk 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. Dekodiranje lahko opravite tudi s pomočjo Goertzelovega algoritma [2], ki se v praksi pogostokrat uporablja za dekodiranje tonov DTMF.

Zvočni signali lahko vsebujejo tudi “motilne” frekvence, ki pa niso enake frekvencam podanih tonov v zgornji tabeli. Te je potrebno pri razpoznavanju ignorirati. Prav tako se

lahko pojavijo tudi posnetki, ki ne vsebujejo vseh potrebnih frekvenc za določen tipko ali pa sploh ne vsebujejo nobenega tona - v teh primerih je izhod vašega programa prazen niz.

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

Prototip funkcije:

```
def naloga4(vhod: list, fs: int, t: float) -> str:
    """
    Poišče sekvenco pritiskov tipk v zvocnem zapisu.

    Parameters
    -----
    vhod : list
        vhodni zvocni zapis
    fs : int
        frekvenca vzorčenja
    t : float
        trajanje posameznega pritiska tipke v sekundah ter pavze za vsakim
        pritiskom tipke

    Returns
    -----
    izhod : str
        niz pritiskov tipk, ki se skriva v zvocnem zapisu
    """

    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:

- v zvočnem signalu bo največ 100 pritiskov tipk, najvišja frekvenca vzorčenja pa 44100,
- pri posameznem testnem primeru dobite točko, če se `izhod` popolnoma ujema z rešitvijo,
- izvajanje funkcije je časovno omejeno na 15 sekund,
- dovoljena je uporaba standardne knjižnice Python 3.11 (<https://docs.python.org/3.11/library/>) in paketov `numpy` (priporočamo) ter `scipy`.

Literatura

[1] Wikipedia: <http://en.wikipedia.org/wiki/DTMF>.

[2] Wikipedia: http://en.wikipedia.org/wiki/Goertzel_algorithm.