

Vysoké učení technické v Brně
Fakulta informačních technologií



Signály a systémy

Projekt

Jaroslav Kvasnička (xkvasn14)

Řešení

Řešeno v Jupyter notebook, v jazyce Python. Ve složce src můžete najít tři soubory, xkvasn14.ipynb je Jupyter notebook, requirements.txt má knihovny které je potřeba použít. Přiložil jsem i xkvasn14.py, který obsahuje stejný kód, pokud by Jupyter notebook nefungoval, lze použít i tuto variantu.

Úkol 1,2

Name	Samples	Seconds
maskoff_ton.wav	58329	3,646
maskoff_sentence.wav	25263	1,579
maskon_ton.wav	43072	3,692
maskon_sentence.wav	24706	1,544
sim_maskon_ton.wav	58329	3,646
sim_maskon_sentence.wav	25263	1,579
sim_maskon_ton_only_match.wav	58329	3,646
sim_maskon_sentence_only_match.wav	25263	1,579

Vzorkovací frekvence
signálů je 16000 Hz.
Velikost jednoho rámce je
20 ms. Všechny soubory
jsem přeměřil a ujistil jsem
se, že nedošlo ke ztrátě
délky.

Úkol 3

Ruční metodou jsou vybral co nejpodobnější rámce o délce 1 sekundy.

Ustřednil pomocí: `maskoff_tone = maskoff_tone - np.mean(maskoff_tone)`

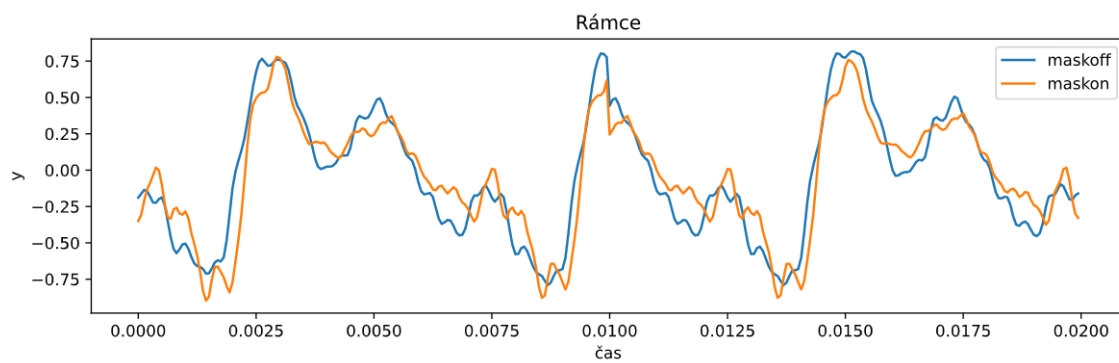
Normalizoval: `maskoff_tone = maskoff_tone/np.abs(maskoff_tone).max()`

Výpočet velikosti rámce: $R = f * t$

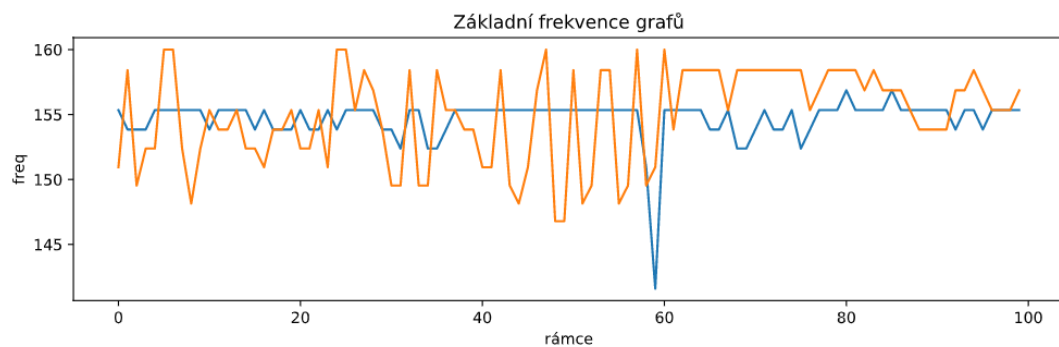
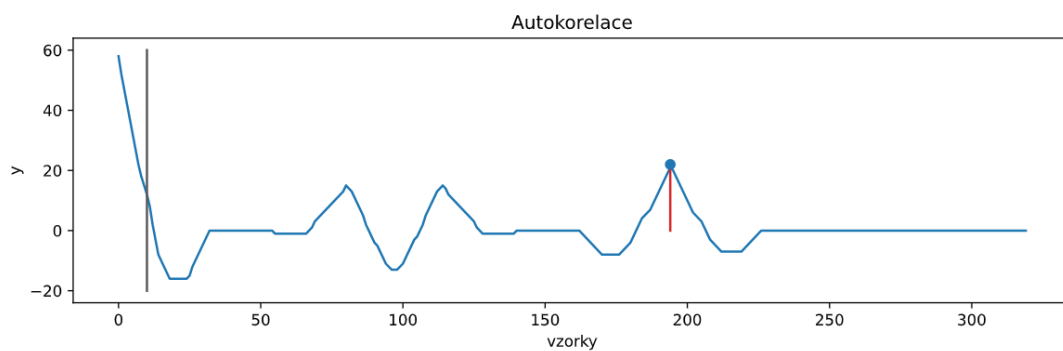
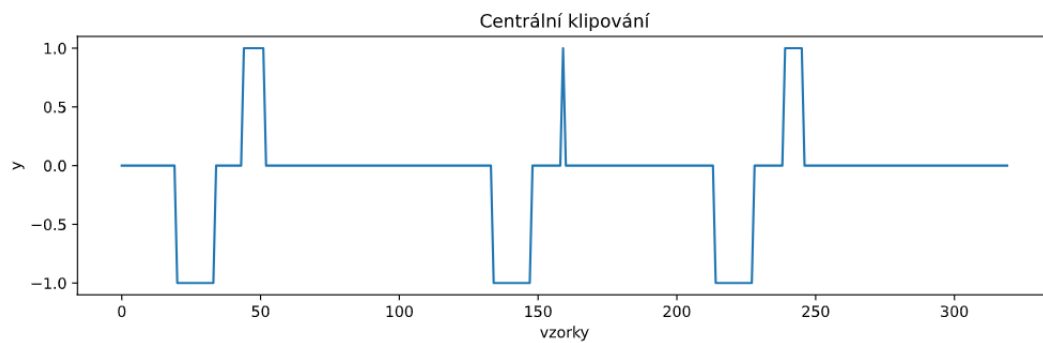
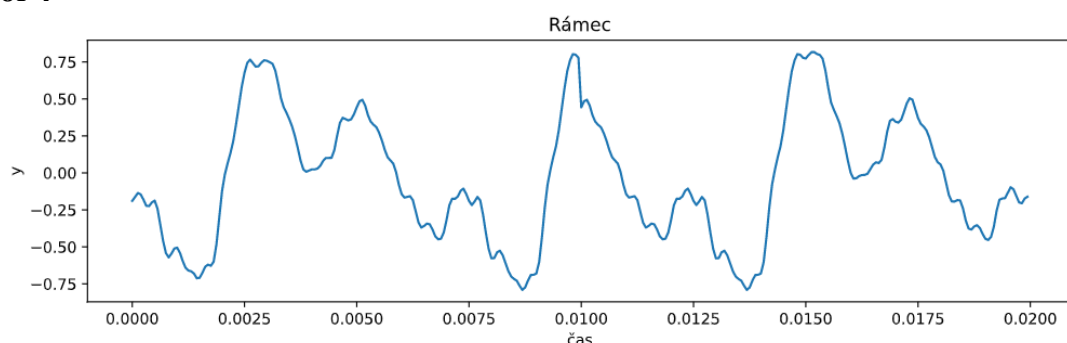
R ... velikost rámce ve vzorcích

f ... frekvence v Hz

t ... čas v ms



Úkol 4



Rozptyl maskoff_tone: 2.7137556650521537
Střední hodnota maskoff_tone: 154.6826970179408
Rozptyl maskon_tone: 13.072347503071355
Střední hodnota maskon_tone: 154.95905600132565

”Lag” se v grafu Autokorelace nachází na prvku 194.

Velikost změny f_0 by se dala zabránit např. úpravou fázového posunu signálů, vyhnutí se rezonancí nebo obdržení lepších nahrávek.

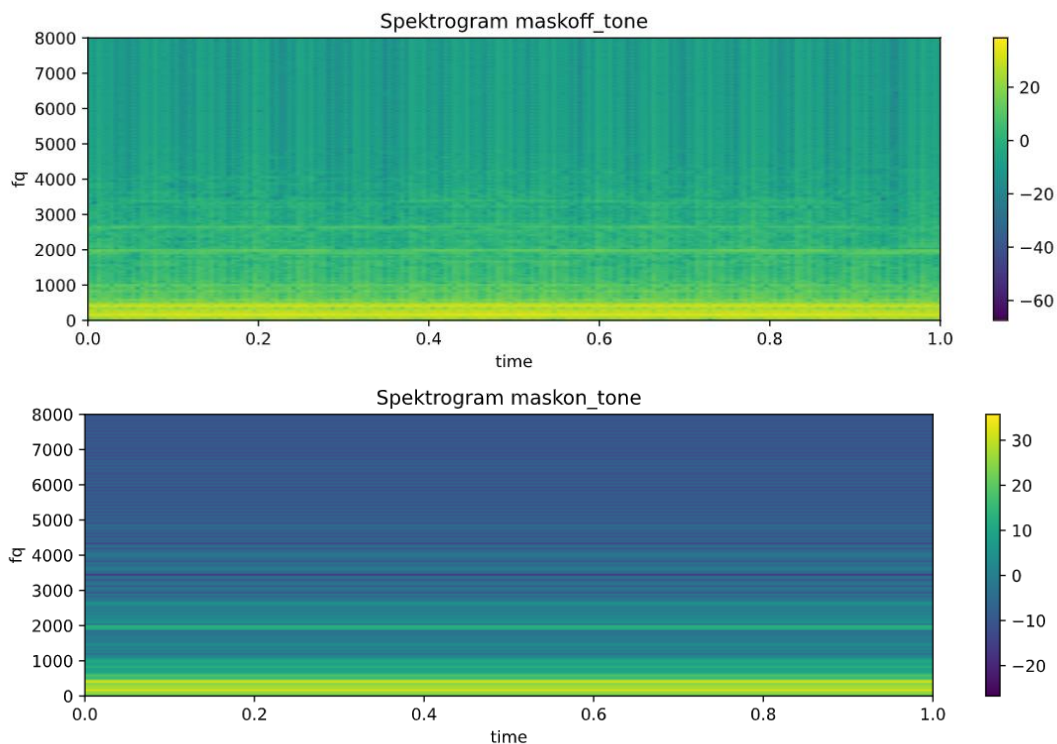
Úkol 5

Funkce Fourierovy diskrétní transformace:

```
def DFT(x):  
    x = np.asarray(x, dtype=complex)  
    M = x.shape[0]  
    f = x.copy()  
    for i in range(M):  
        sum = 0  
        for j in range(M):  
            tmp = x[j]*np.exp(-2j*np.pi*j*i*np.divide(1, M, dtype=complex))  
            sum += tmp  
        f[i] = sum  
    return f
```

Pro urychlení načítání používám vestavěné funkce `np.fft.fft()`

Výsledky těchto dvou funkcí jsou v porovnání stejné.



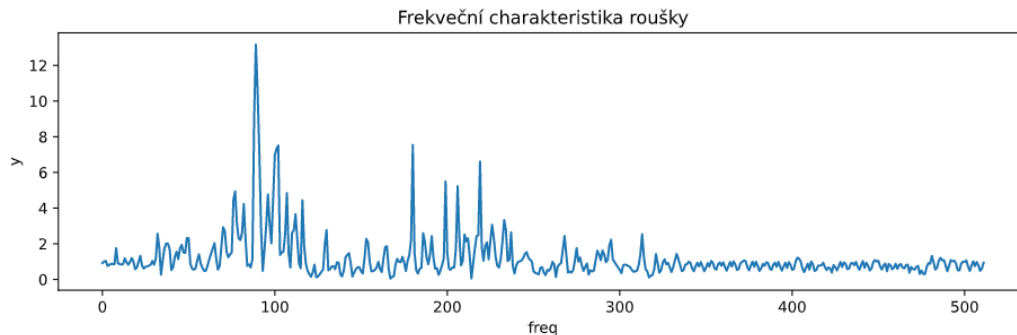
Úkol 6

Vzorec pro výpočet $H(e^{j\omega})$.

$$H(e^{j\omega}) = \frac{B(e^{j\omega})}{A(e^{j\omega})} = \frac{b[0]e^{-j\omega*0} + b[1]e^{-j\omega*1} + \dots + b[M]e^{-j\omega M}}{a[1]e^{-j\omega*0} + a[1]e^{-j\omega*1} + \dots + a[N]e^{-j\omega N}}$$

Průměroval jsem hodnoty pouze s absolutními hodnotami:

```
for i in range(len(Fourier1)):
    H12.append(abs(Fourier2[i])/abs(Fourier1[i]))
```



Rouška bude nejvíce ovlivňovat charakteristiky signálů od 50 do 250 Hz

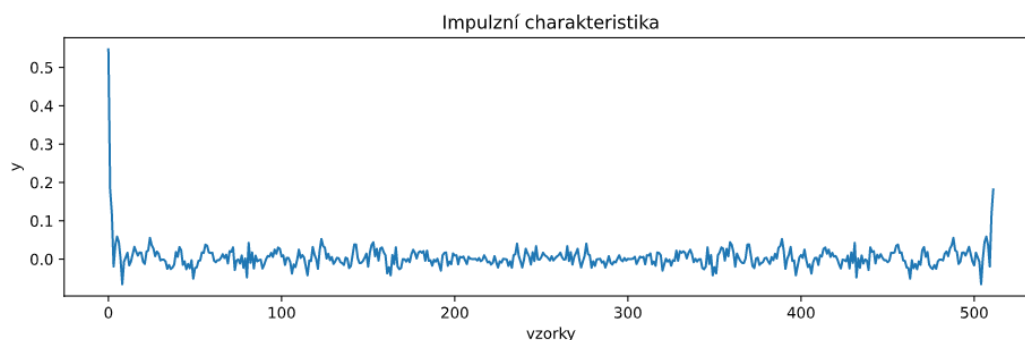
Úkol 7

Funkce zpětné Fourierovy transformace:

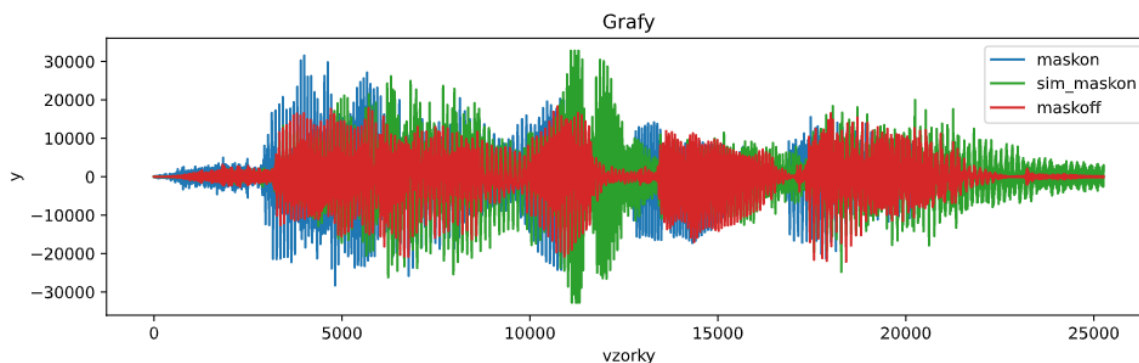
```
def idFT(x):
    x = np.asarray(x, dtype=complex)
    M = x.shape[0]
    f = np.zeros(M, dtype=complex)
    for i in range(M):
        sum = 0
        for j in range(M):
            tmp = x[j]*np.exp(2j*np.pi*i*j*np.divide(1, M, dtype=complex))
            sum += tmp
        f[i] = np.divide(sum, M, dtype=complex)
    return f
```

Pro urychlení načítání používám vestavěné funkce `np.fft.ifft()`

Výsledky těchto dvou funkcí jsem porovnával grafy, a byly totožné.



Úkol 8

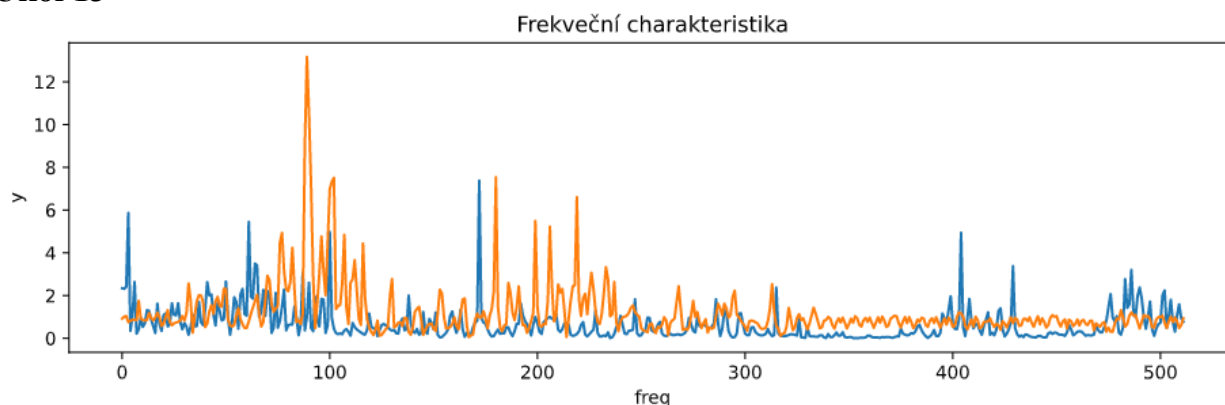


Signály nahrávané věty se liší především u vyslovování začátku slov, kdy člověk na ně přirozeně dává akcent. Uprostřed věty došlo nejspíše k rezonanci, protože tam simulovaný signál dostává vyšší hodnoty než oba dva původní signály zvlášť. Naopak nejpodobnější jsou na začátku a konci nahrávek.

Úkol 9

Viz závěr

Úkol 13



Graf zobrazuje frekvenční charakteristiky s a bez použití rámců se stejnou základní frekvencí.

Oranžový graf ... bez použití metody stejné frekvence

Modrý graf ... s použitím metody stejné frekvence

Závěr

Největší potíže při vypracovávání jsem měl kvůli špatnému nahrávacímu zařízení. Od toho se odvozuje většina nedostatků jako nelineární základní frekvence, nebo příliš zvětšená charakteristika v cílovém signálu.

Mohli jsme si ve 13 úloze osvědčit, že pokud by rouška měla pouze takové malé spektrum, tak by interagovala se signály daleko méně a tudíž, by řeš tolik netlumila.

Zdroje

Dokumentace Numpy, Scipy, Matplotlib, Discord diskuze, Stackoverflow fórum a Likegeeks fórum, ISS opora, Zmolikova_repositories Jupyter