



Signály a systémy

Protokol

Tomason Viktoryia
xtomas34

30.12.2020

1 Úloha

Vzorkovací frekvence signálu je 16000 [Hz].

Bitová šířka 16 bitů

Tabulka testovacích nahrávek (tony)

Název souboru	Délka ve vzorcích	Délka ve sekundách
maskon_tone.wav	32356	00:00:02.02
maskoff__tone.wav	27650	00:00:01.73

2 Úloha

Tabulka testovacích nahrávek (věty)

Název souboru	Délka ve vzorcích	Délka ve sekundách
maskon_sentence.wav	90505	00:00:05.66
maskoff__sentence.wav	89003	00:00:05.56

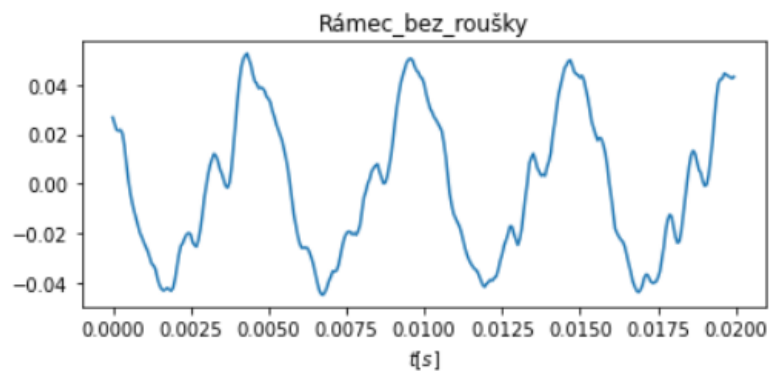
3 Úloha

Vzorec pro výpočet velikosti rámce ve vzorcích bez roušky :

```
maskoff_frame = []
```

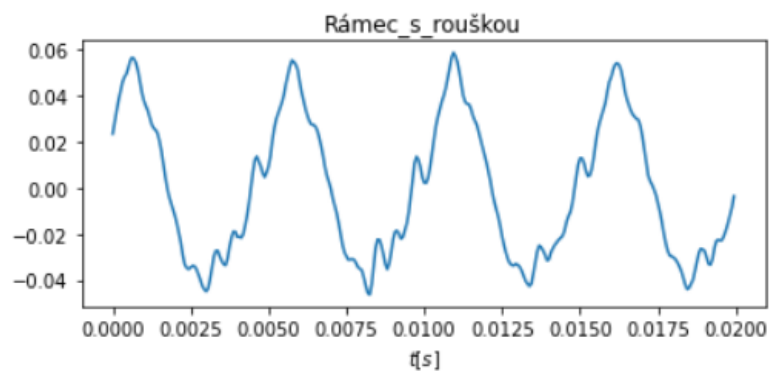
```
for v i range(99):
```

```
    maskoff_frame.append(np.array(tone_off[int(v*0.01*16000):int((v+2)*0.01*16000)]))
```



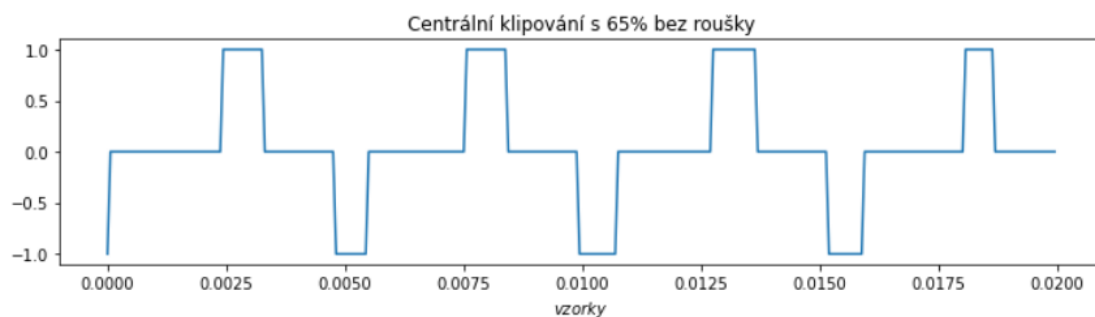
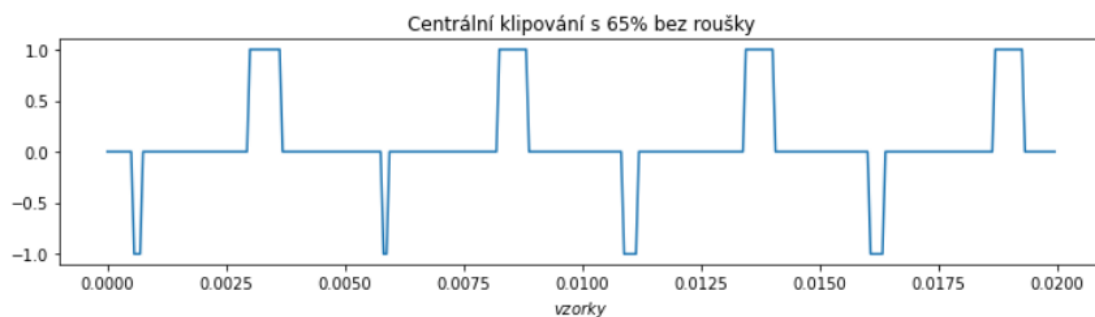
Vzorec pro výpočet velikosti rámce ve vzorcích s rouškou :

```
maskon_frame = []  
for v in range(99):  
    maskon_frame.append(np.array(tone_on[int(v*0.01*16000):int((v+2)*0.01*16000)]))
```

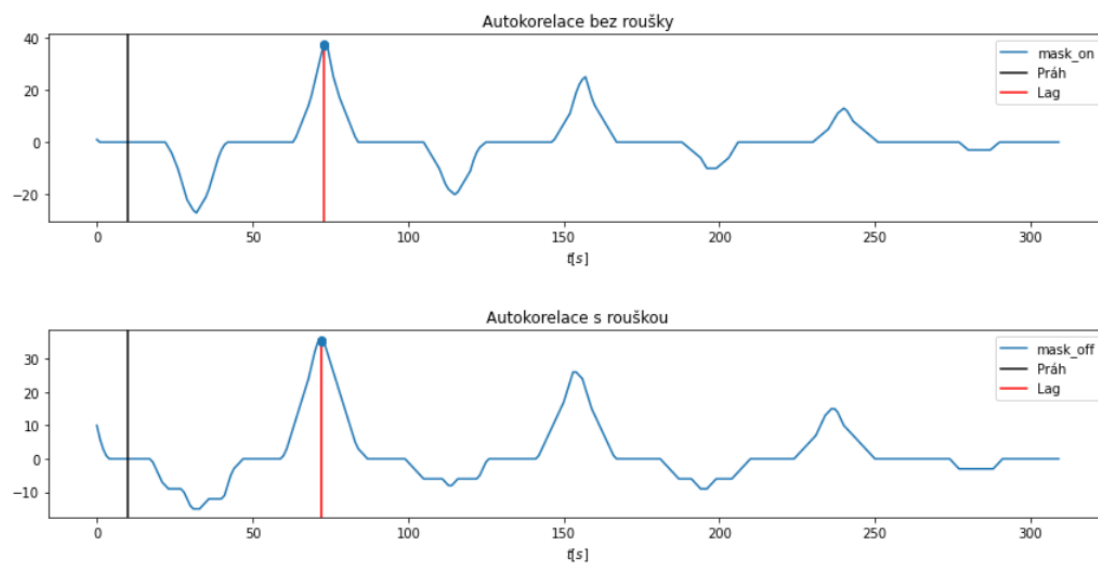


4 Úloha

Na rámce bylo aplikováno centrální klipování s 65%



Autokorelace rámce



Bez roušky:

Střední hodnota = 220.91723996669654

Rozptyl = 2.004500165125178

S rouškou:

Střední hodnota = 219.2478904807672

Rozptyl = 6.128244195362446

Odpověď na otázku:

Při vysokém klipování se mění počet vzorků, při změnšování počet tak zvaných "kopců" se zvětší. A ještě při malé délce rámce bude počet "kopců" velmi malý, což nebude stačit

5 Úloha

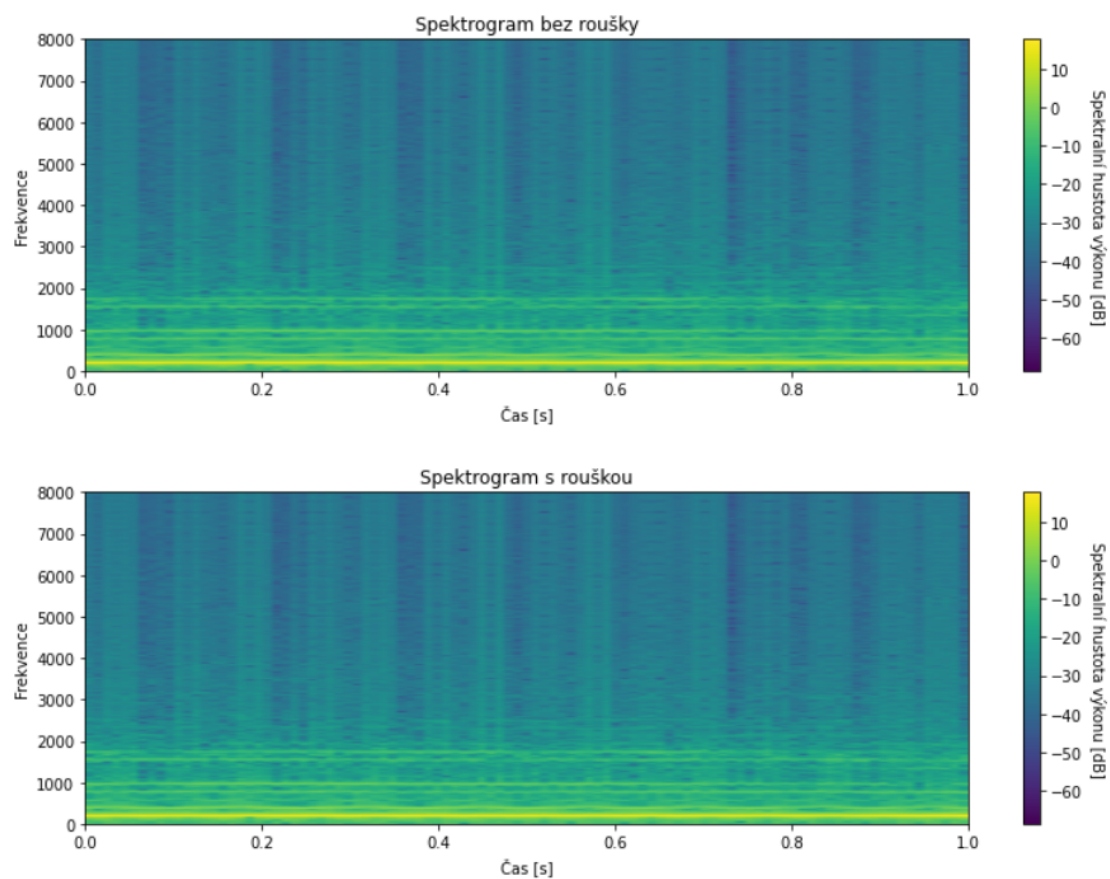
Vzorec pro výpočet funkce DFT :

$$F(k) \equiv \sum_{n=0}^{N-1} f(n) e^{-\frac{2\pi i}{N} kn}$$

Vlastní naimplementovaná funkce DFT, kde $N=1024$

```
def dft_func(mask_frame):  
    tone_dft = []  
    N=1024  
  
    for value in mask_frame:  
        tone_mask_dft = []  
        for k in range(1024):  
            dft = 0  
            step = 0  
            for suma in range(320):  
                dft = value[suma] * cmath.exp(-(cmath.pi*2j*k*suma/N))  
                step = step + dft  
            tone_mask_dft.append(step)  
        tone_dft.append(np.array(tone_mask_dft))  
    return tone_dft
```

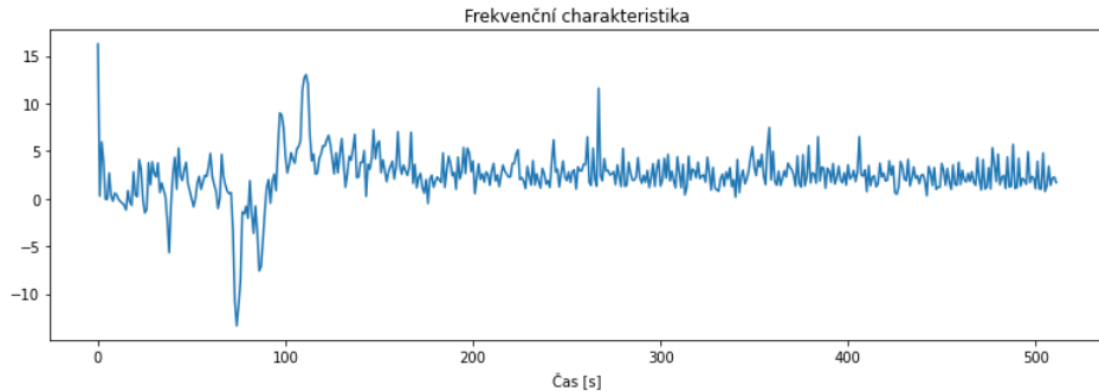
Vykreslené spektrogramy



6 Úloha

Vztah pro výpočet $H(e^{j\omega}) = \frac{DFT_maskon}{DFT_maskoff}$

Frekvenční charakteristika



Komentář:

Filtr simuluje roušku, a používám jen 512 vzorků kvůli zrcadlení seznamu

7 Úloha

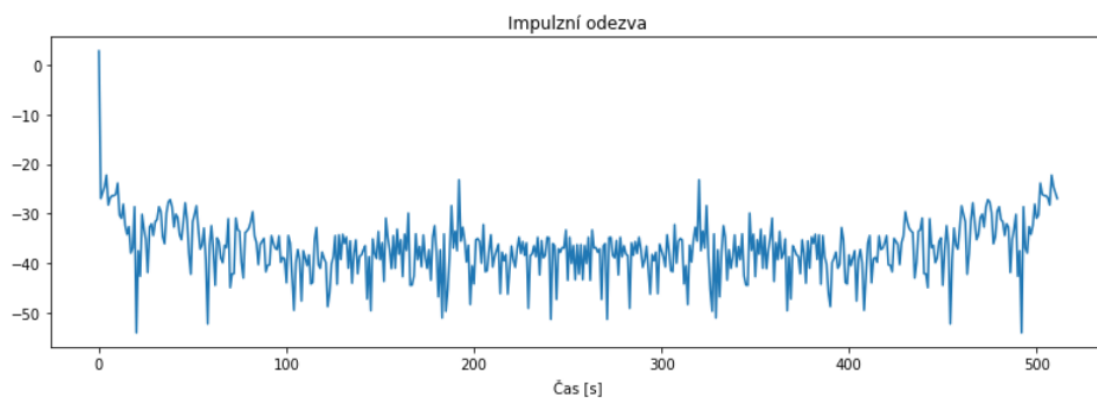
Vzorec pro výpočet funkce IDFT :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X_{DFT}[k] e^{j2\pi nk/N} \quad n = 0, 1, \dots, N-1$$

Vlastní naimplementovaná funkce IDFT

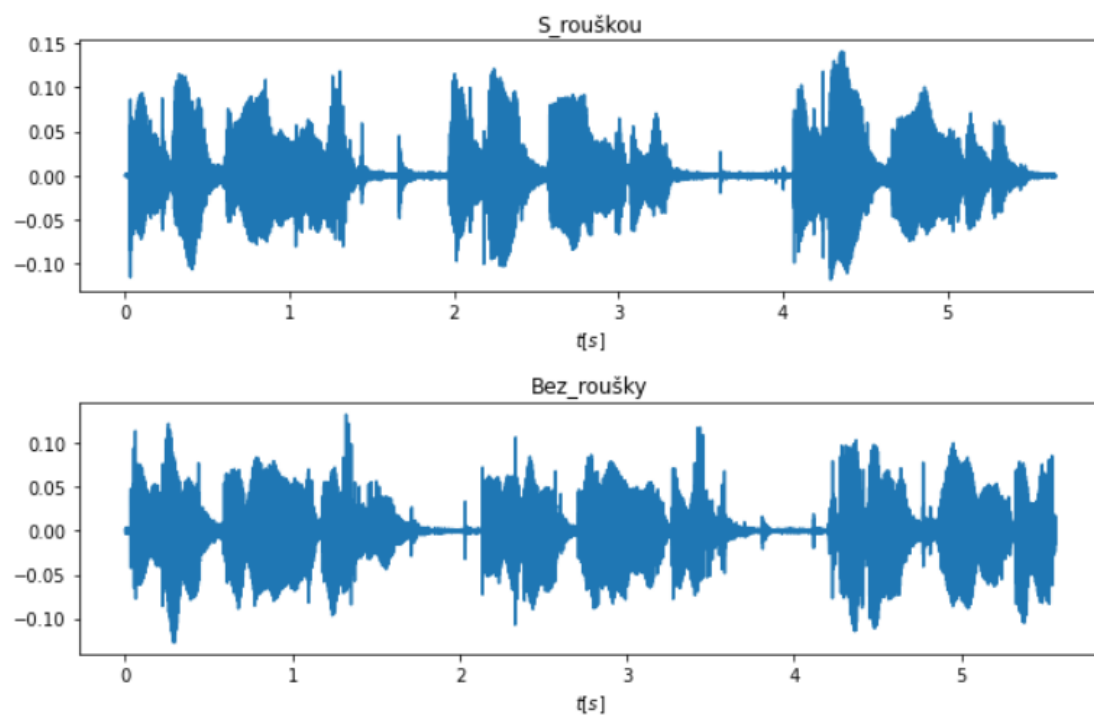
```
def dft_func(mask_frame):  
    tone_dft = []  
    N=1024  
  
    for value in mask_frame:  
        tone_mask_dft = []  
        for k in range(1024):  
            dft = 0  
            step = 0  
            for suma in range(320):  
                dft = value[suma] * cmath.exp(-(cmath.pi*2j*k*suma/N))  
                step = step + dft  
            tone_mask_dft.append(step)  
        tone_dft.append(np.array(tone_mask_dft))  
    return tone_dft
```

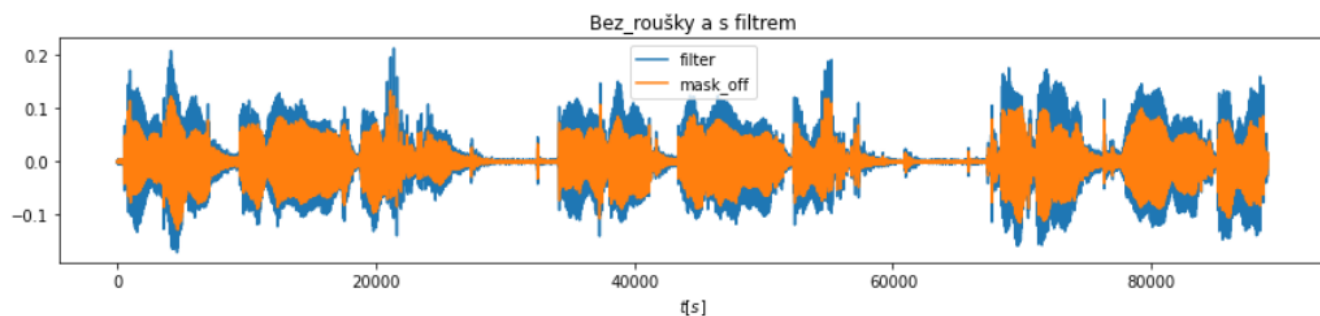
Impulzní odezva:



8 Úloha

Grafy nahrané věty s rouškou, bez roušky a se simulovanou rouškou





Ano, signál z rouškou a se simulovanou rouškou jsou podobné, ale rozdíl je vidět. Nejvíc se liší v těch místech kde je signál vyšší, jinak se dost podobají.

9 Úloha

Tento projekt byl zaměřen na práci se signály. První obtíže na kterou jsem narazila, byla hlasový záznam v jednotném tónu. Díky přehlednosti grafů bylo možné vidět změny za různých podmínek, ale ne vždy se grafy kreslily tak, jak měly být, kvůli nezkušenosti práce z signály. Studijní materiály umožnily práce s projektem a rozšířily moje znalosti