

FIT VUT

ISS Projekt

Protokol s řešením

Peter Růček, xrucek00
3.1.2021

1. Úloha

maskon_sentence.wav	2.76s 44211 samples
maskoff_sentecne.wav	2.65s 42353 samples

2. Úloha

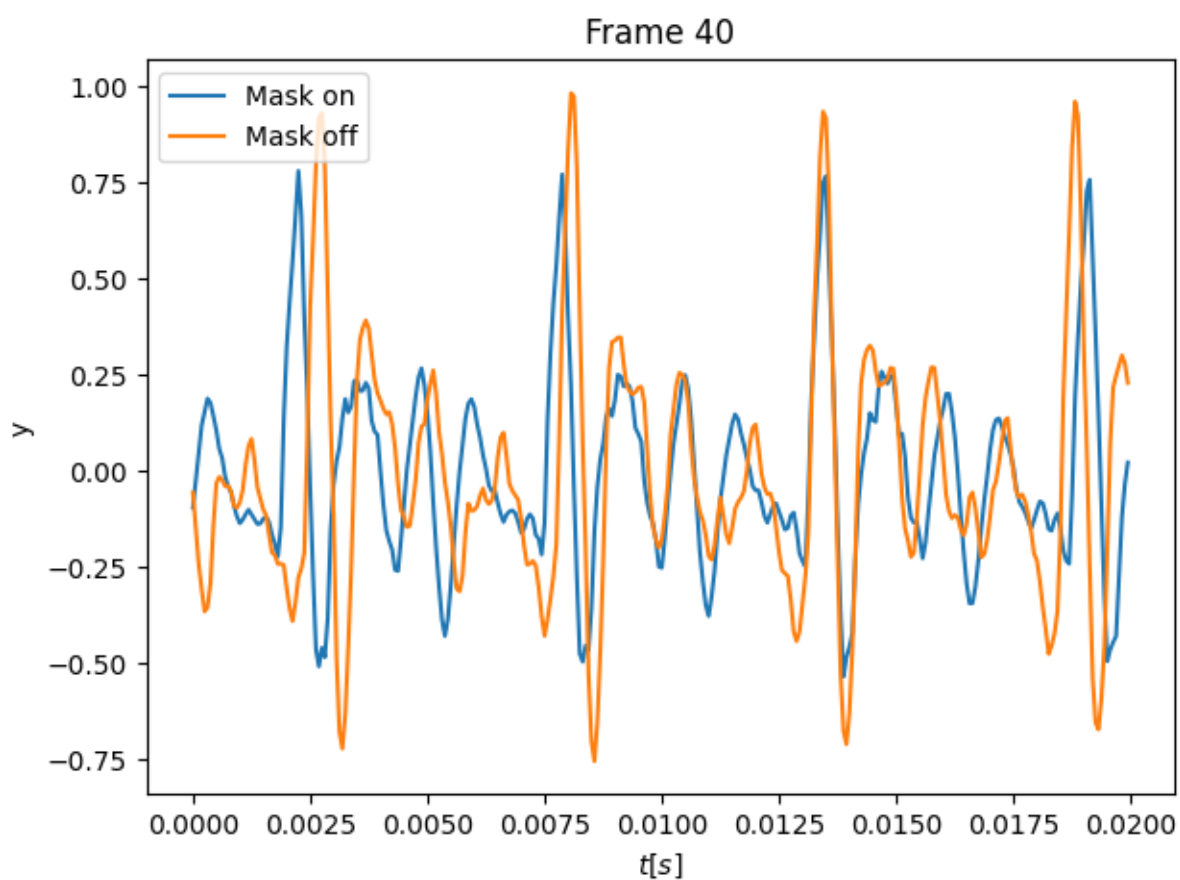
maskoff_tone.wav	2.41s 38638 samples
maskon_tone.wav	2.28s 36409 samples

3. Úloha

$\text{framesize} = \text{window_size} * \text{fs}$

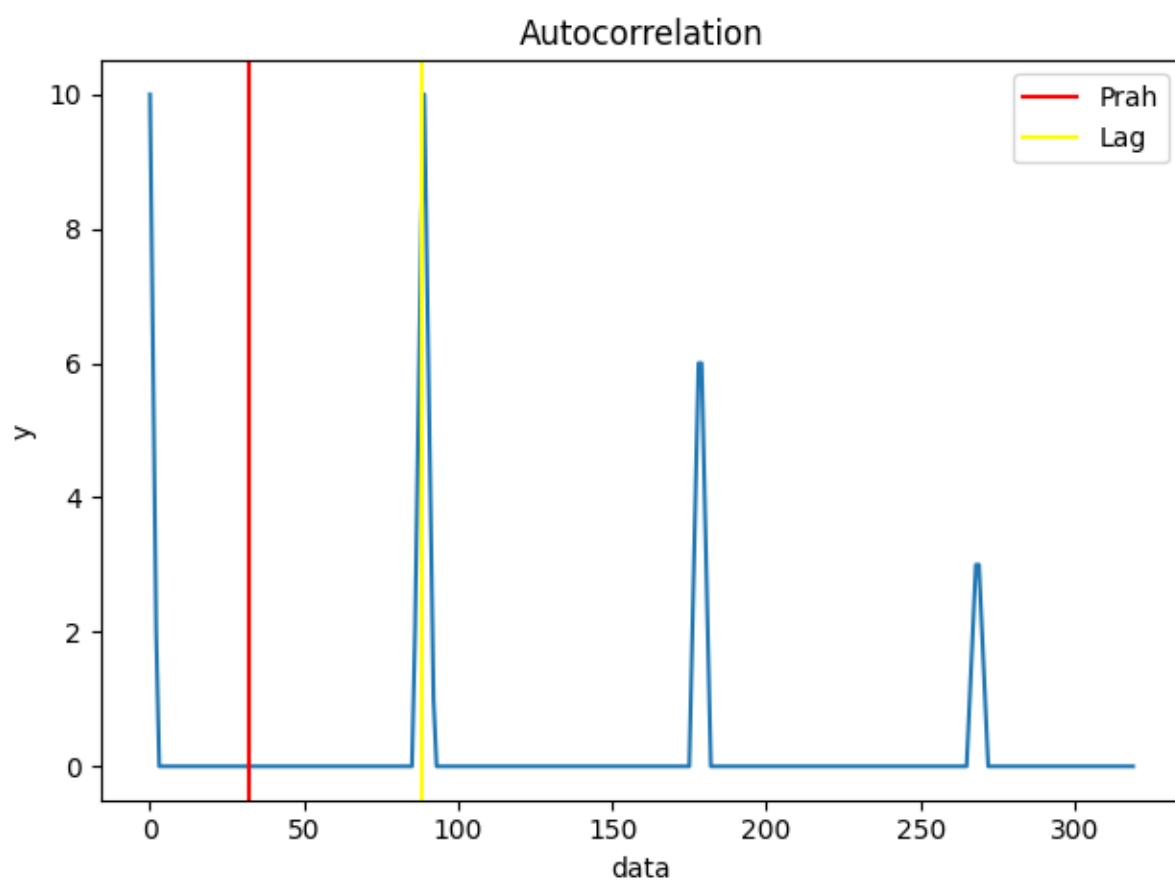
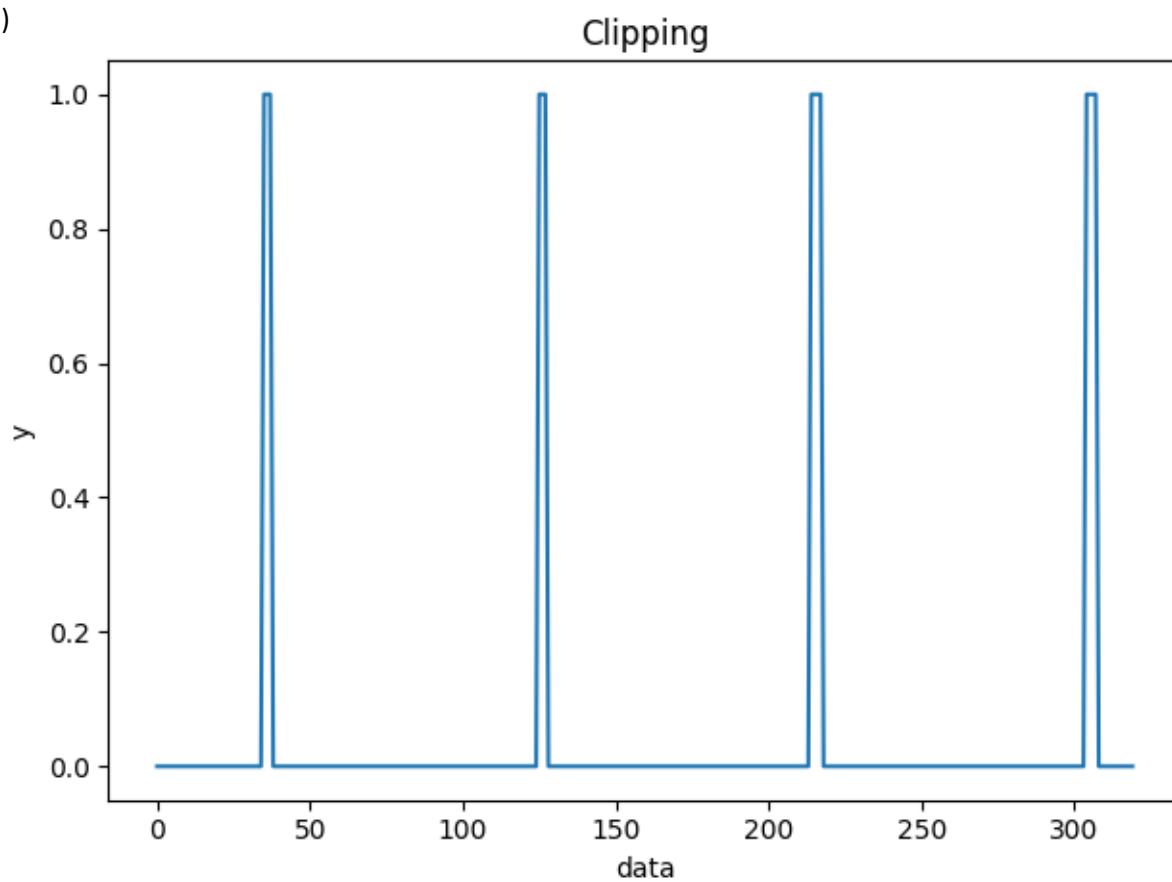
$\text{window_size} = 20 \text{ ms}$

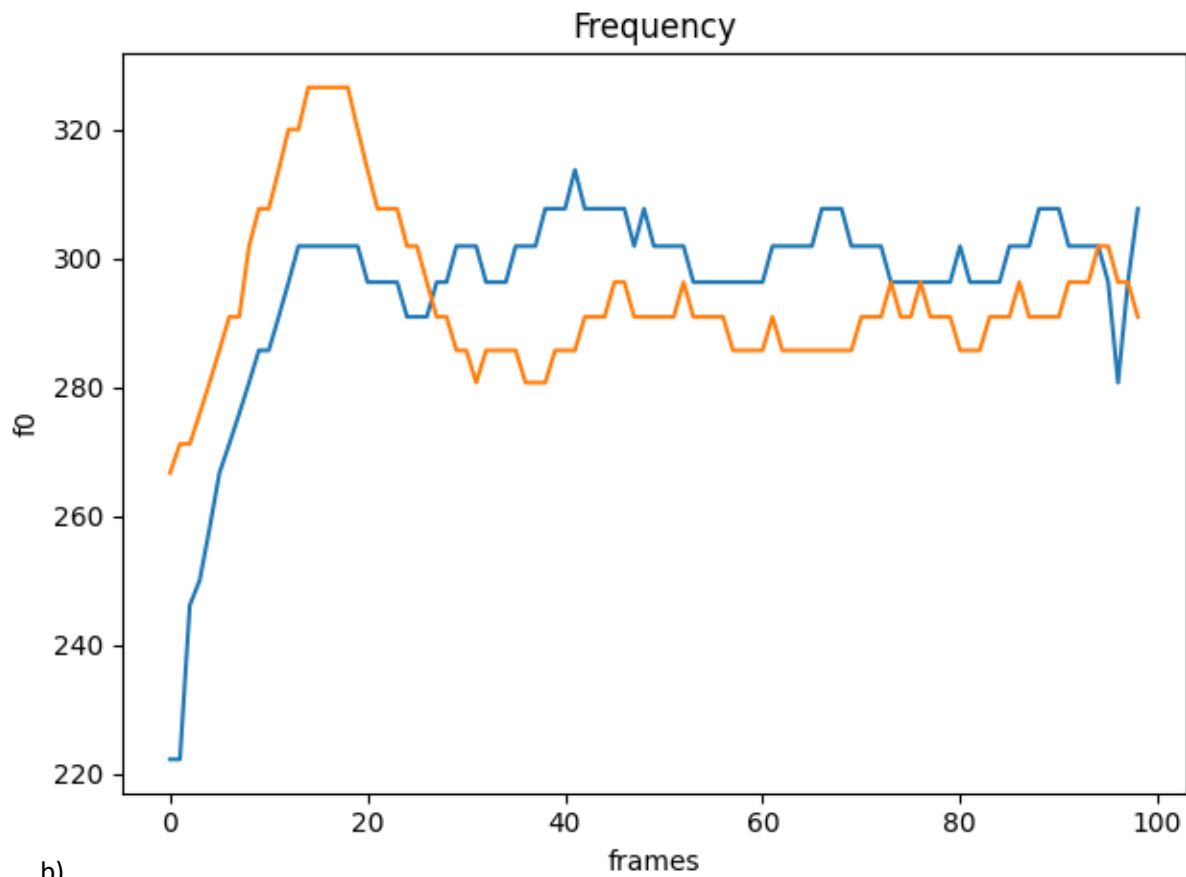
$\text{fs} = 16\,000 \text{ Hz}$



4. Úloha

a)





b)

Mask On - stredná hodnota = 290.90909090909093

Mask Off - stredná hodnota = 301.8867924528302

Mask On - rozptyl = 145.61056584681748

Mask Off - rozptyl = 235.43183771923574

c)

Mohli by sme skúsiť spriemerovať danú hodnotu s predošlou hodnotou.

5. Úloha

a)

```

N = 1024

# My dft implementation

def dft(data,k):
    result = 0
    for n in range(N):
        result += data[n]*cmath.exp(-1j*2*cmath.pi*k*n/N)
    return result

dft_result_on = np.zeros((len(iss3.on_frames),N),dtype=np.complex128)
dft_result_off = np.zeros((len(iss3.off_frames),N),dtype=np.complex128)

def padding(data):
    result = np.zeros((len(data),N))
    for i in range(len(data)):
        result[i] = np.pad(data[i], (0,N-len(data[i])), "constant", constant_values = (0,0))
    return result

padded_off = padding(iss3.off_frames)
padded_on = padding(iss3.on_frames)

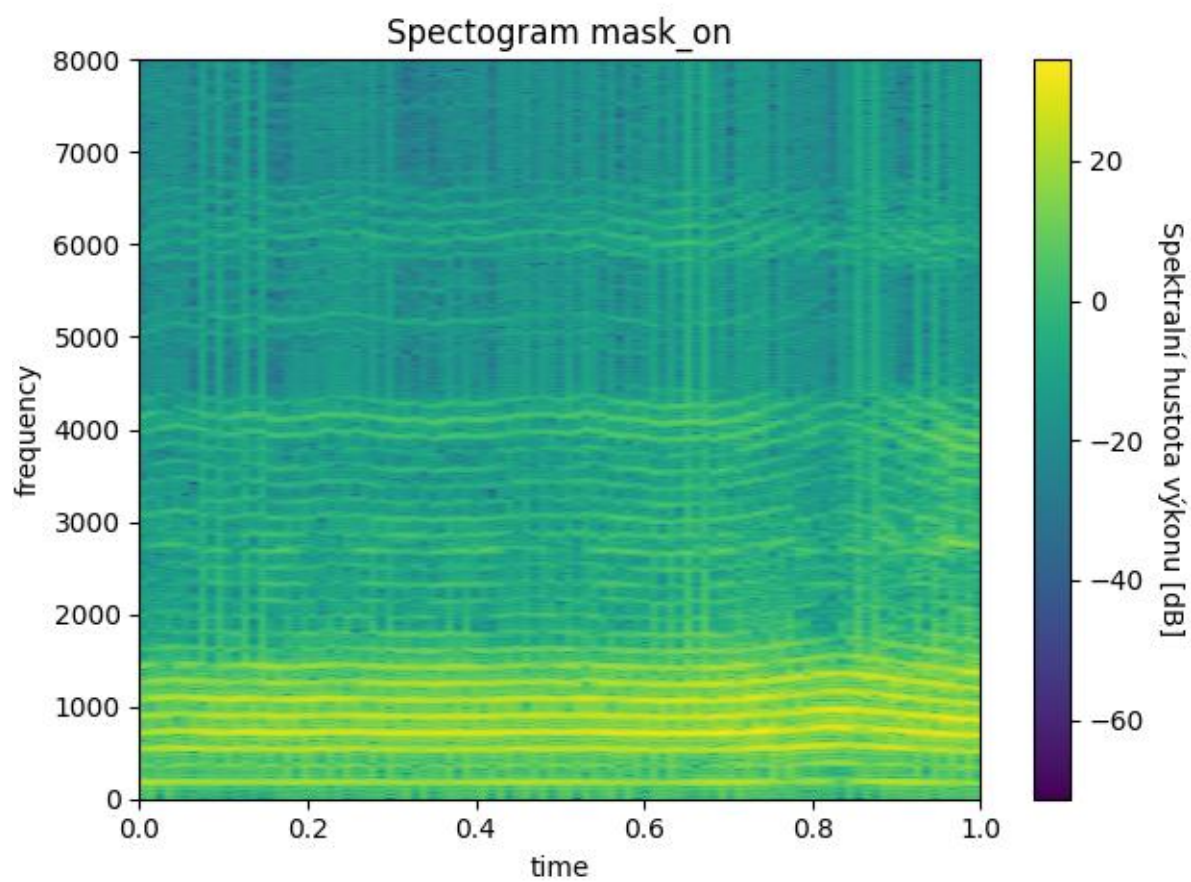
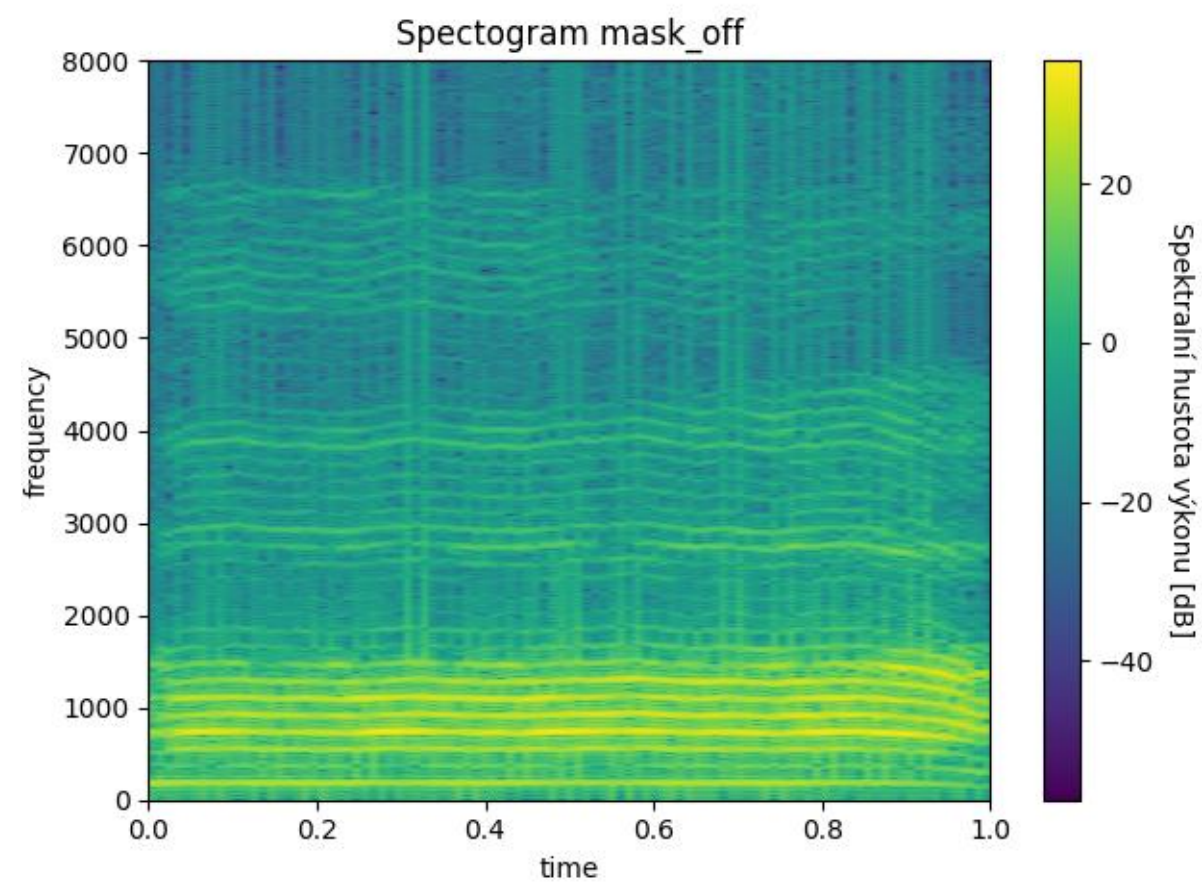
for i in range(len(padded_off)):
    for k in range(N):
        dft_result_off[i][k] = dft(padded_off[i],k)
        dft_result_on[i][k] = dft(padded_on[i],k)

to_print_off = 10* np.log10(np.abs(dft_result_off)**2)
to_print_on = 10* np.log10(np.abs(dft_result_on)**2)

half_to_print_off = np.array(to_print_off)[: , 0:512]
half_to_print_on= np.array(to_print_on)[: , 0:512]

```

b)

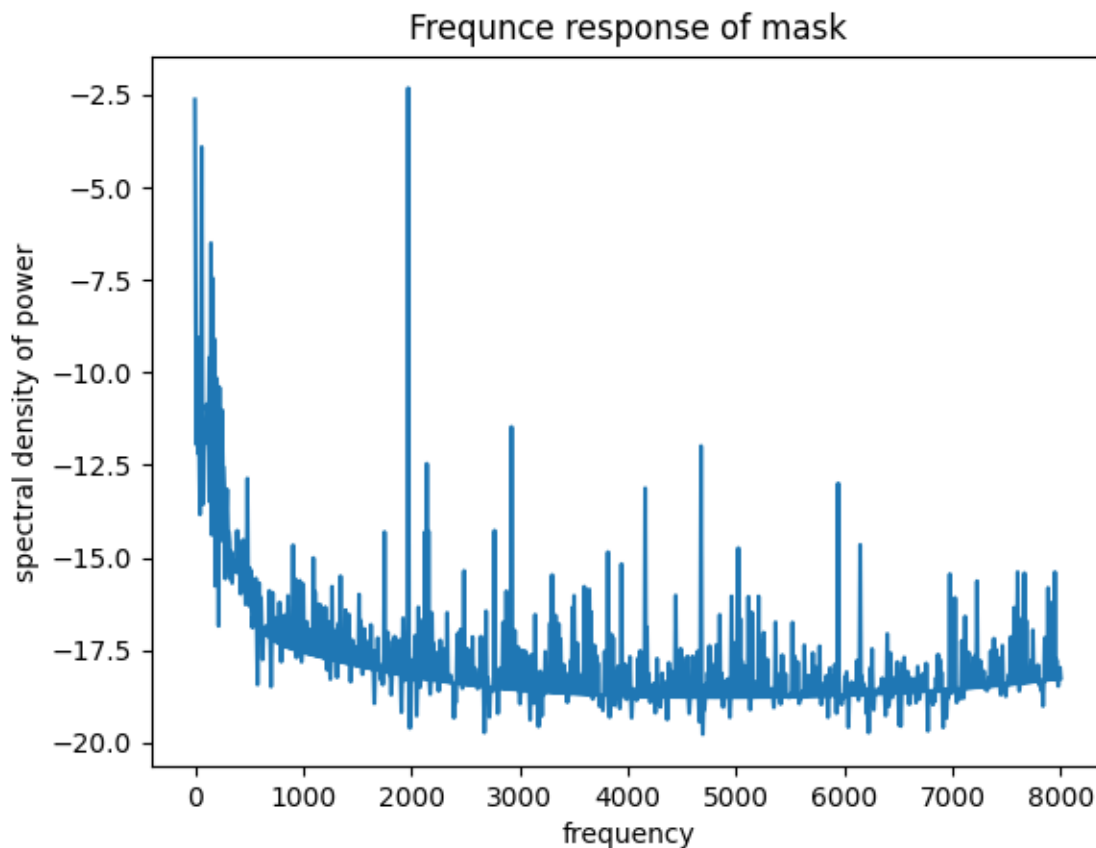


6. Úloha

a)

$$H(e^{j\omega}) = \frac{\sum_{r=0}^M b_r e^{-j\omega r}}{\sum_{k=0}^N a_k e^{-j\omega k}}$$

b)



c)

Filter vznikol z vzorca a) pričom v čitateli sa nachádzal výsledok DFT s maskou a v menovateli výsledok DFT bez masky. Asi tento filter nebude úplne presný, plynie to z rôznych chýb a nepresností, ktorých som sa mohol dopustiť, ktoré budem detailnejšie rozoberať v závere.

7. Úloha

a)

```
N = 1024

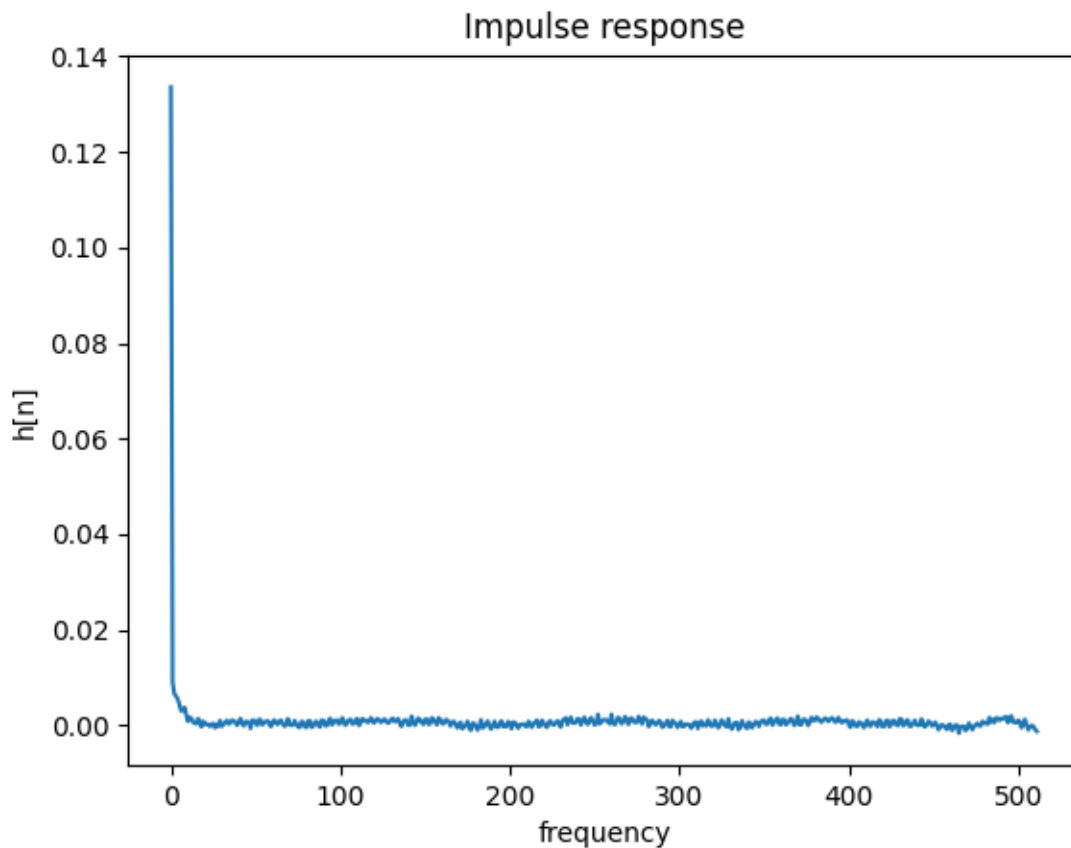
'''
My idft implementation
'''

def idft(data,n):
    result = 0
    for k in range(N):
        result += data[k]*cmath.exp(1j*2*cmath.pi*k*n/N)
    return result/N

my_idft = np.zeros((len(iss6.result)), dtype=np.complex128)

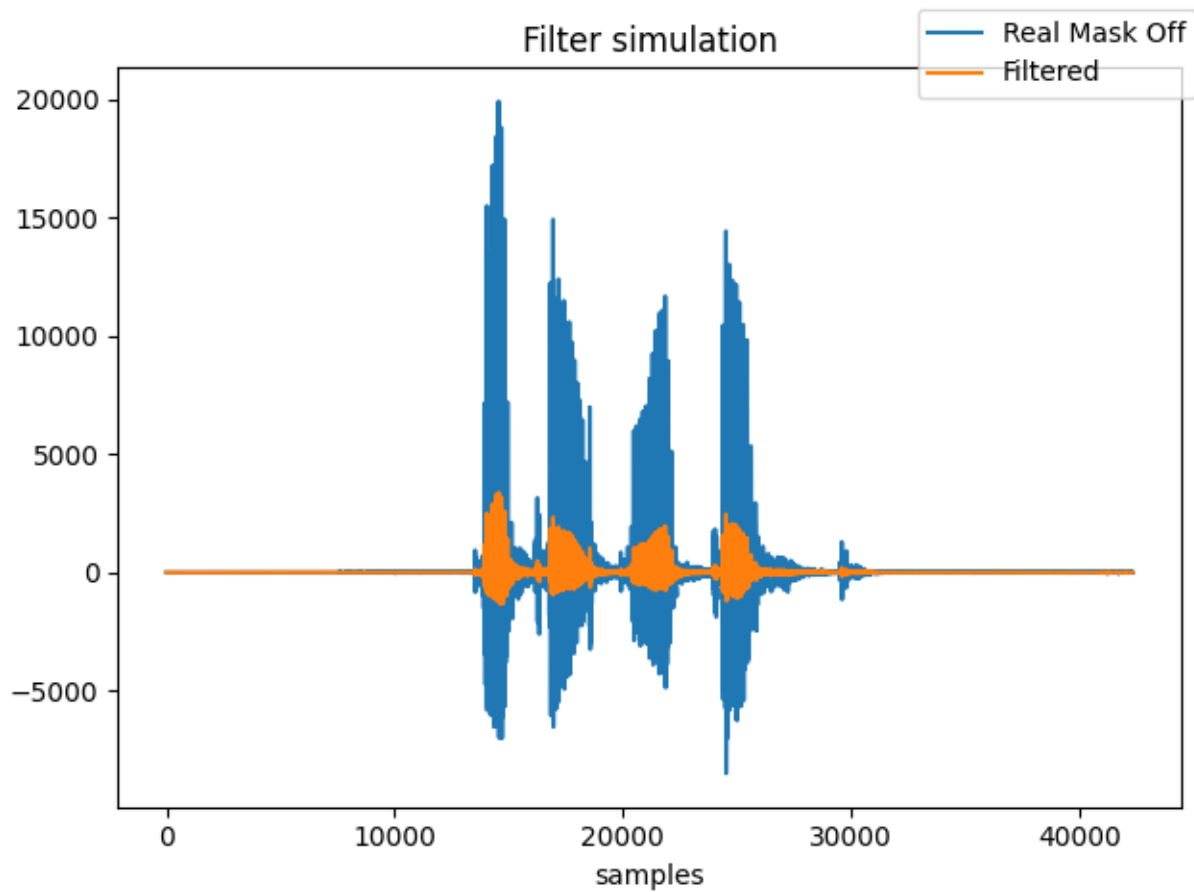
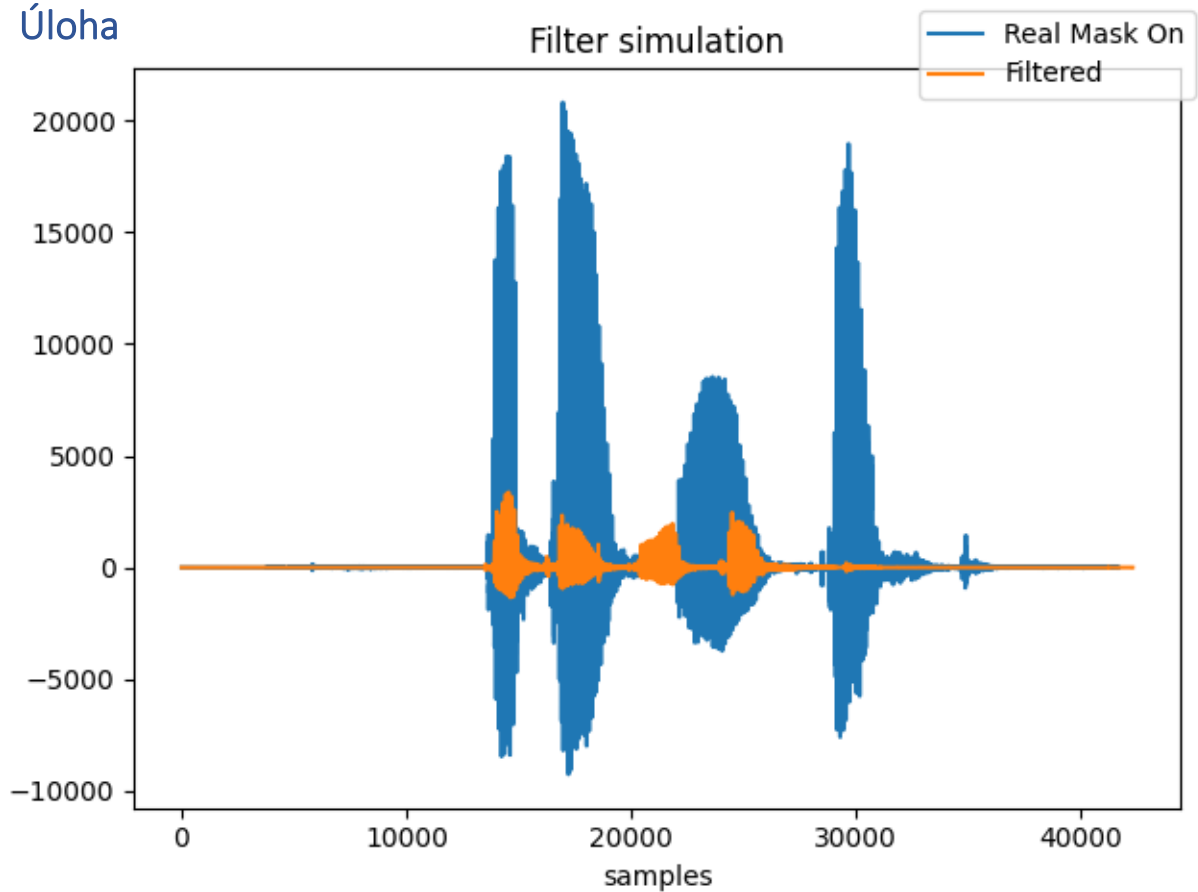
for i in range(N):
    my_idft[i] = idft(iss6.result,i)
```

b)



8. Úloha

a)



b)

Výsledky sa veľmi nepodobajú, je to spôsobené pravdepodobne zlým filtrom. Takisto veta asi nebola nahraná rovnako (s rovnakými pauzami, intonáciou a pod., napriek tomu, že som sa snažil). Nahrávku som nahrával viac krát a to čo vidíte je to najlepšie čo mi vyšlo.

9. Záver

Na začiatok treba povedať, že to bol veľmi náročný projekt, strávil som nad ním veľa času, no bol zaujímavý. Prvý krát som robil v pythone ako aj so signálmi. Stretlo ma ja veľké množstvo problémov, z ktorých niektoré ani neviem či som vyriešil. Napríklad, pri vykresľovaní základnej frekvencie rámcov sa mi viac krát stalo, že mi f_0 vyšlo nekonečno, bolo to kvôli tomu, že mám moc hrubý hlas a nenašiel v rámci žiaden lag ($\text{lag} = 0$). Nepomohlo mi ani zväčšenie rámca na 25 ms, tak som nahrávku prehrával viac krát, dokonca som poprosil aj brata, či by mi nahrávku nemohol nahráť on. To čo môžete vidieť je to najlepšie čo sa mi podarilo. Myslím si, že aj toto spôsobilo, že aj filter zlyhal a simulácia rúška sa nepodarila.

Projekt som začal riešiť v dostatočnom predstihu, no blížiac sa skúšky a fakt že ma postihol COVID-19 ma trochu zastavili. Nebyť aj týchto prekážok, by som sa mohol viacej zanoriť do problému a zlepšiť výsledky. Projekt bol veľmi zaujímavý, poučný, no veľmi náročný.