BRNO UNIVERSITY OF TECHNOLOGY FACULTY OF INFORMATION TECHNOLOGY

Signály a systémy

2020/2021

ISS Projekt - Protokol

Šesták Pavel(xsesta07)

Brno, January 2, 2021

Contents

1 Tabulka vstupních signálů

..

Název nahrávky	Délka nahrávky[s]	Počet vzorků
maskoff_tone	01.00	16000
maskon_tone	01.00	16000
maskoff_sentence	05.56	89036
maskon_sentence	05.84	93413

2 Úloha 3 - Rámce

Vstupní signály byly ustředněny a normaliyzovány do dynamického rozsahu [-1;1]
Frekvence vstupního signálu je 16Khz, podělením dostaneme počet vzorků (samples) na 1ms

$$f = 16KHz => 1ms = 16$$

1 milisekunda odpovídá 16 vzorkům vstupního signálu

1 rámec (frame) je dlouhý 20ms, vynásobením předchozí hodnoty získáme počet vzorků v rámci

$$20ms = 20 * 16 = 320$$

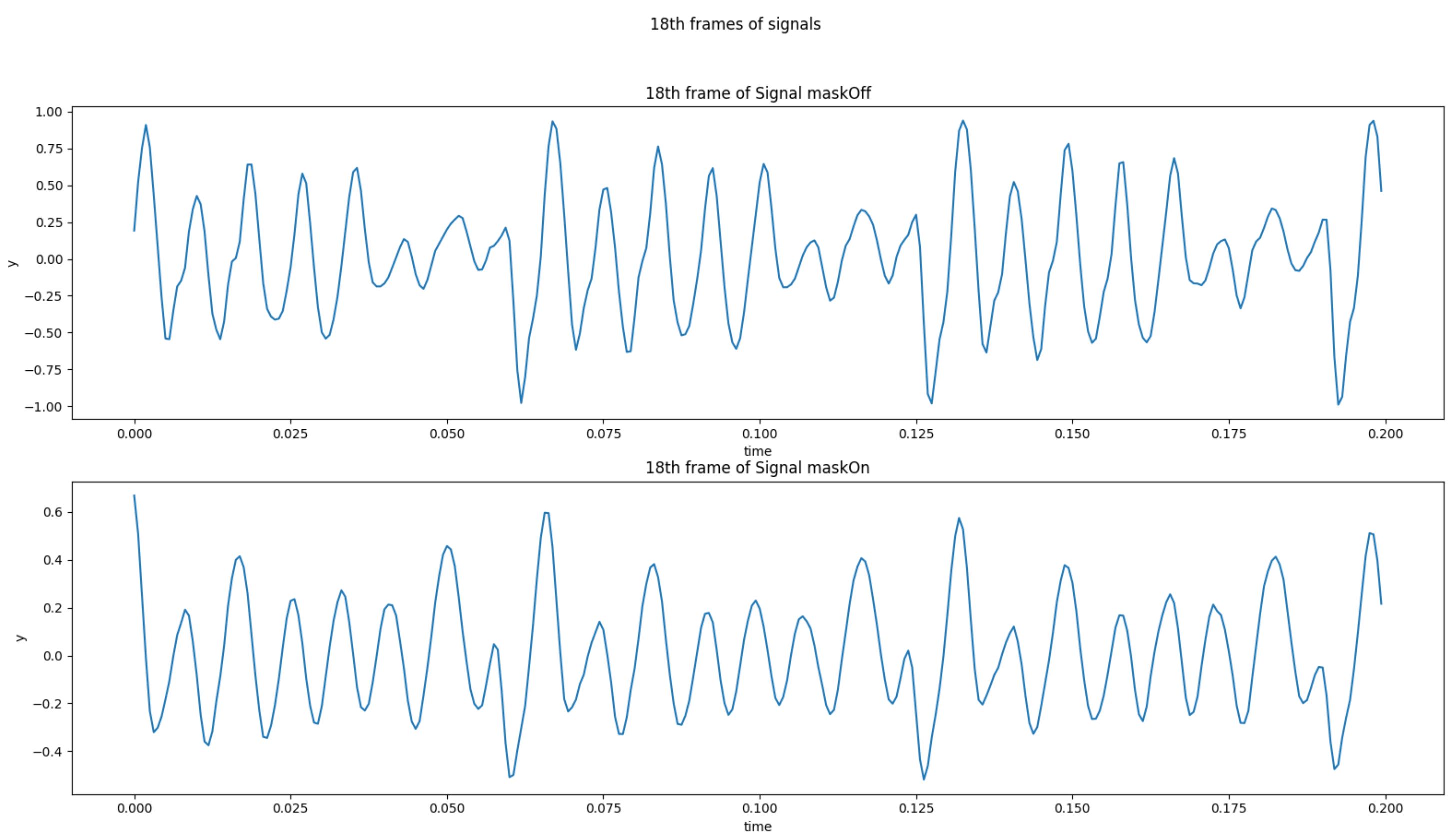
Výsledná délka rámce je tedy 320 vzorků

Rámce se vždy polovinou své délky překrývají, nyní vypočítáme počet rámců na 16 000 vzorků

$$samples/(frame/2) = 16000/160 = 100$$

Poslední rámec se již nevleze do 1s celý, pouze jeho polovina, z tohoto důvodu dále uvažujeme pouze 99 rámců

18. rámec signálu bez roušky a s rouškou



3 Úloha 4 - základní frekvence rámců

Z předchozí úlohy jsme zjistili že máme 99 rámců a každý rámec obsahuje 320 hodnot. Nyní bude demonstrován výpočet základní frekvence 18 tého rámce. Pro získání grafu základních frekvencí je nutné výpočet provést pro každý rámec.

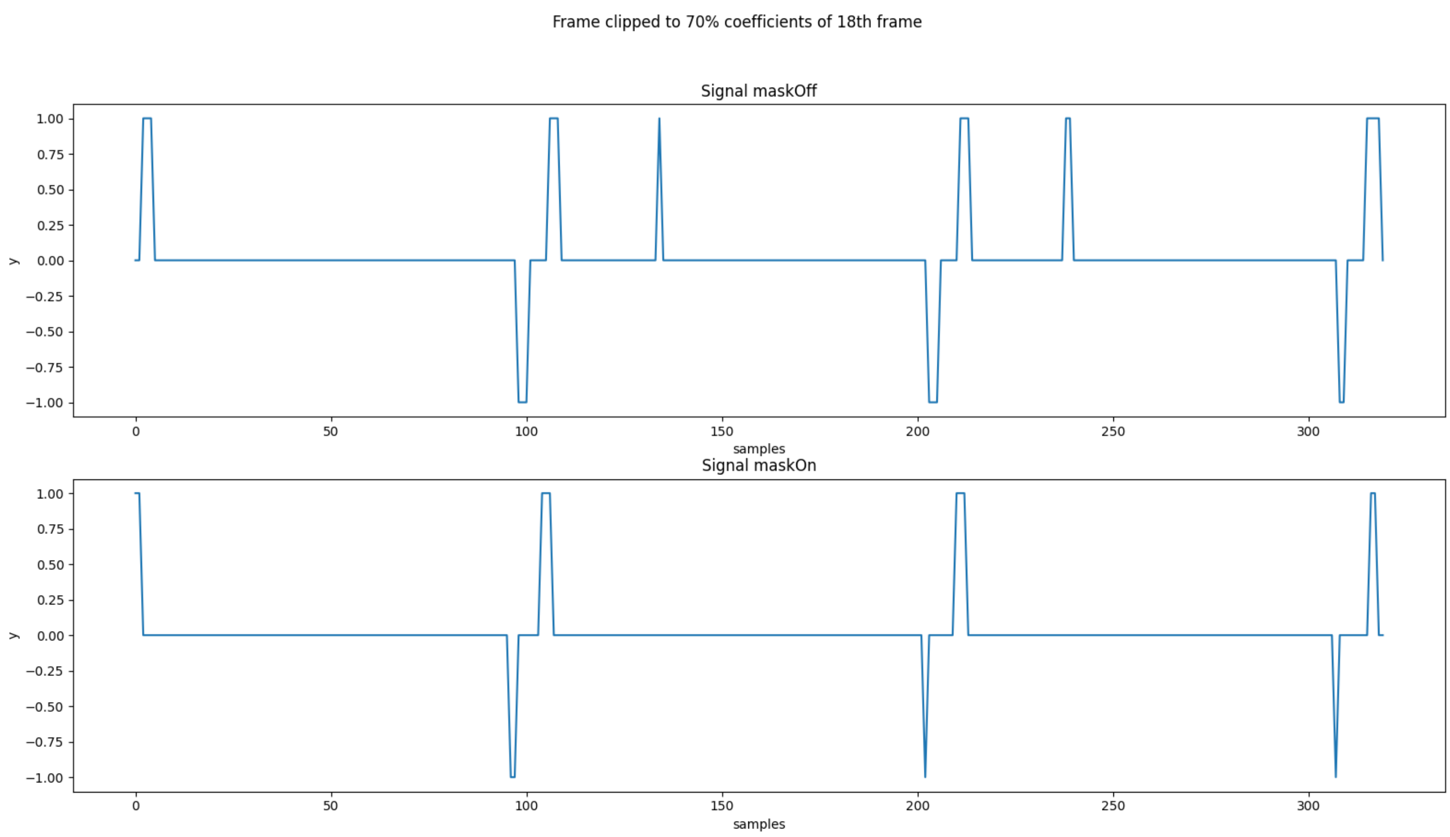
3.1 Centrální klipování s 70% práhem

Pro výpočet centrálního klipování musíme najít hodnotu práhu P s kterou budeme následně vzorky porovnávat

$$P = max(abs(frame)) * 0.7$$

Následně provedeme porovnání pro každý vzorek v rámci a v případě že hodnota je větší než P pak hodnotu nastavíme na 1, v případě že hodnota je menší než -1*P pak nastavíme na -1 v ostatních případech bude hodnota rovna 0.

18. rámec po centrálním klipování



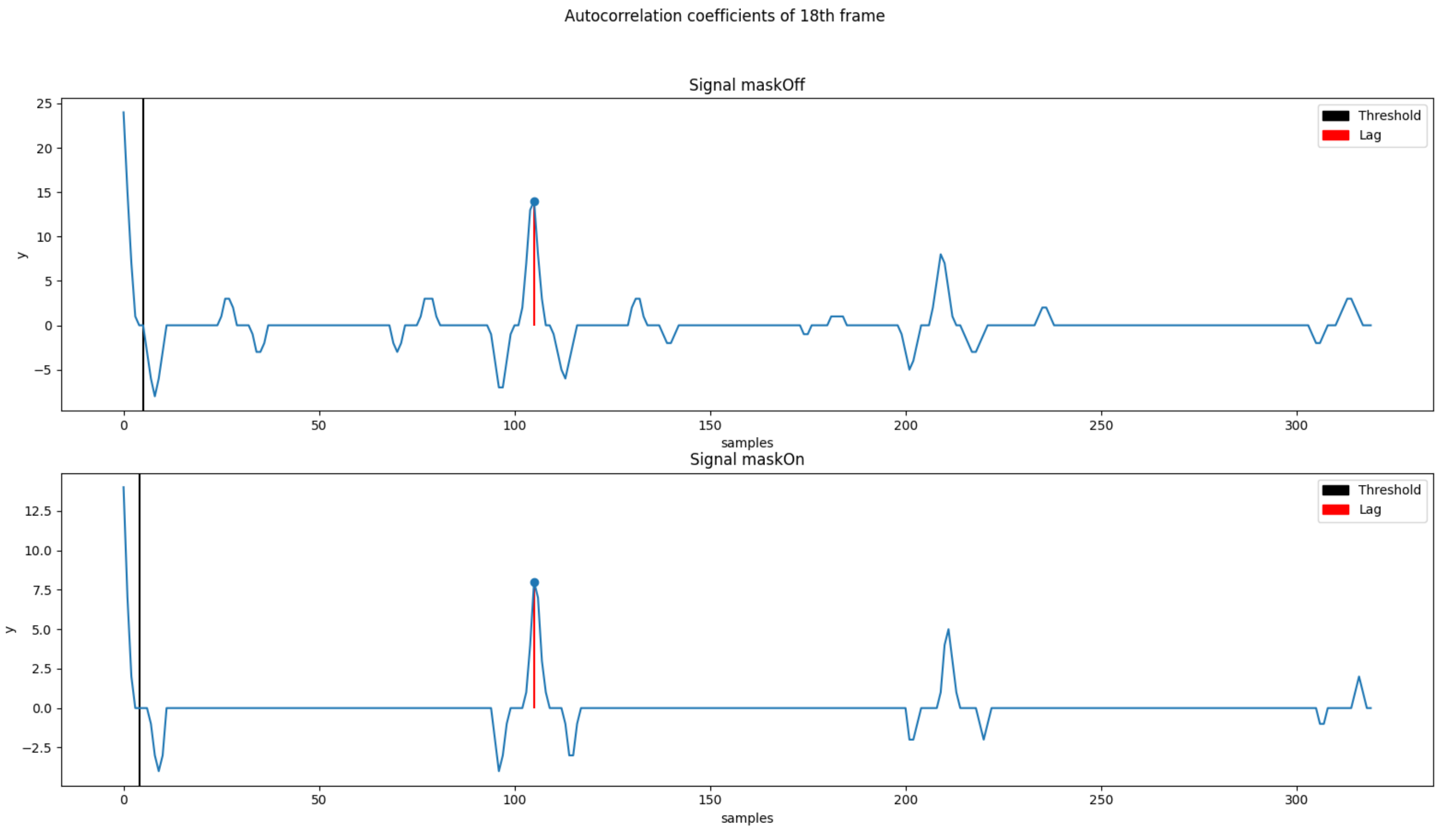
3.2 Autokorelace rámců

Nyní provedeme autokorelaci na signál, který jsme získali centrálním klipováním

$$R(k) = \sum_{0}^{N-k} s[k]s[k+n]$$

Autokorelace nám vyjde vysoká u nuly, protože každý signál je podobný sám na sebe. Proto si určíme práh (trashold), od kterého budeme hledat maximální hodnotu neboli lag. Index práhu v projektu je určován prvním výskytem hodnoty 0 v signálu a následně inkrementován o 1.

18. rámec po autokorelaci



3.3 Základní frekvence rámců

Z předchozího výpočtu jsme získali koeficient lagu. Nyní vyjdeme z těchto vztahů

Perioda základního tónu je dána vztahem:

$$T_0 = \frac{1}{F_0}$$

Lag je dán vztahem:

$$L = T_0 F_S$$

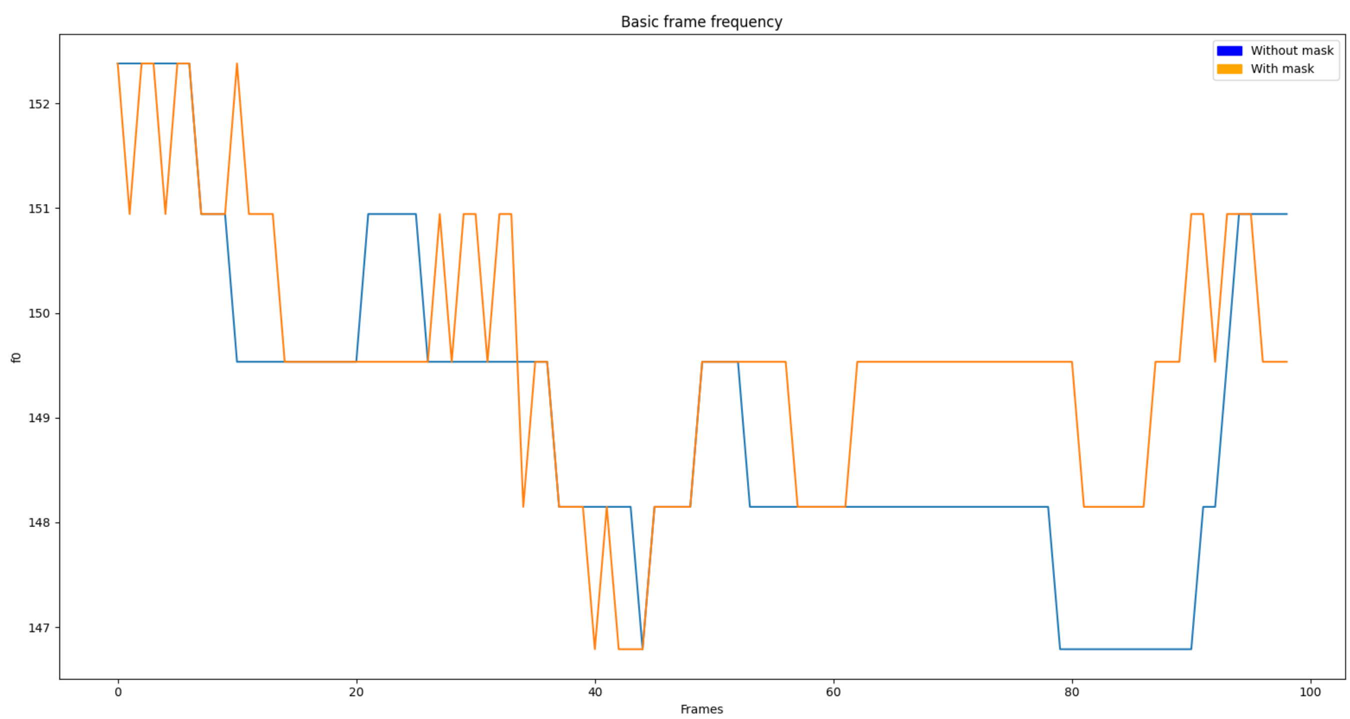
Po výpočtu autokorelace jsme získali hodnotu L, hodnota $\mathrm{F_S}$ je vzorkovací frekvence dána signálem Nyní vyjádříme $\mathrm{F_0}$

$$F_0 = \frac{F_S}{L}$$

Z důvodu že $\mathrm{F_S}$ je dost vysoké číslo (řádově nižší desítky tisíc) a L je hodnota okolo 100 tak velký rozdíl mezi těmito hodnotami při změně L o +- 1 způsobí velký rozdíl ve změně frekvence. Snížení změn by se docílilo snížením vzorkovací frekvence.

Střední hodnota pro signál bez roušky: 149.0136
Střední hodnota pro signál s rouškou: 149.5713
Rozptyl pro signál bez roušky: 1.5196
Rozptyl pro signál s rouškou: 1.5196

Frekvence rámců



4 Úloha 5 - Spektrogramy

4.1 Diskrétní Fourierova transformace

Pro výpočet spektrogramu budeme potřebovat vypočítat DFT.
Vzorec pro výpočet DFT:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j2\pi \frac{k}{N} n}$$

Implementace DFT:

```
for i in range(SIGNAL_COUNT):
      for frame_id in range(FRAMES_COUNT):
         print("Frame ",frame_id)#checking status (very long operation)
         for k in range(N):
            for n in range(SAMPLES_IN_FRAME):
               X[i][frame_id][k] += frames[i][frame_id][n]*(np.exp(-1j*2*np.pi*(k/N)*n))
```

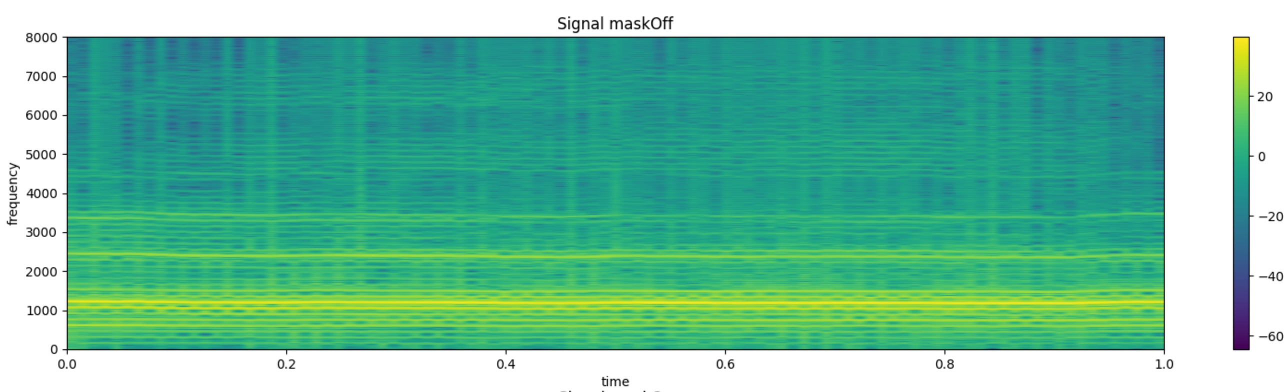
V rámci matematických knihoven, v našem případě numpy, je možno volat numpy.fft.fft(signal). FFT neboli rychlá Fourierova transformace patří k nejvýznamnějším algoritmům na světě a významně urychluje výpočet DFT. Implementace základní DFT je velmi časově náročná a výpočet trvá v řádu minut.

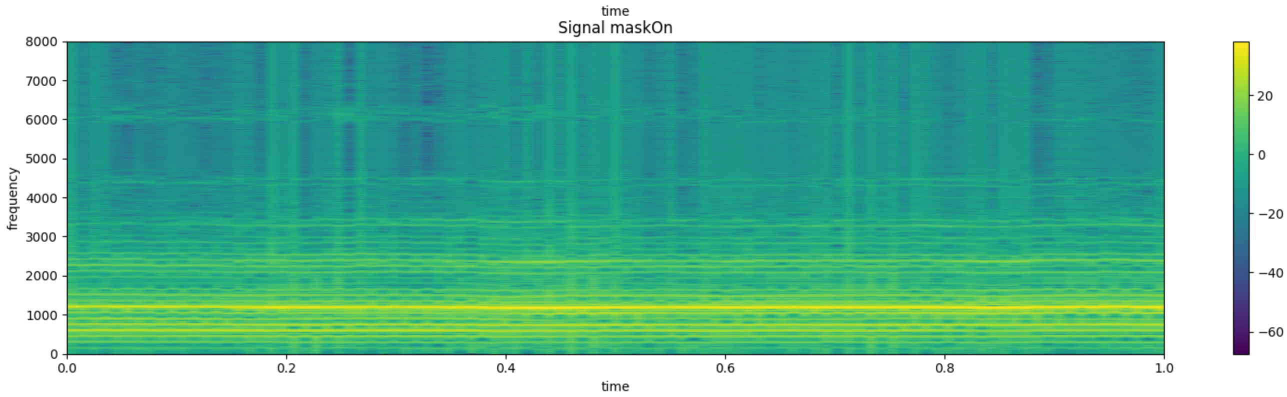
4.2 Logaritmický výkonový spektrogram

Koeficienty získané pomocí DFT je nyní nutné upravit na výkon.

$$P[k] = 20log_{10}|X[k]|$$

Následně necháme vykreslit spektrogram



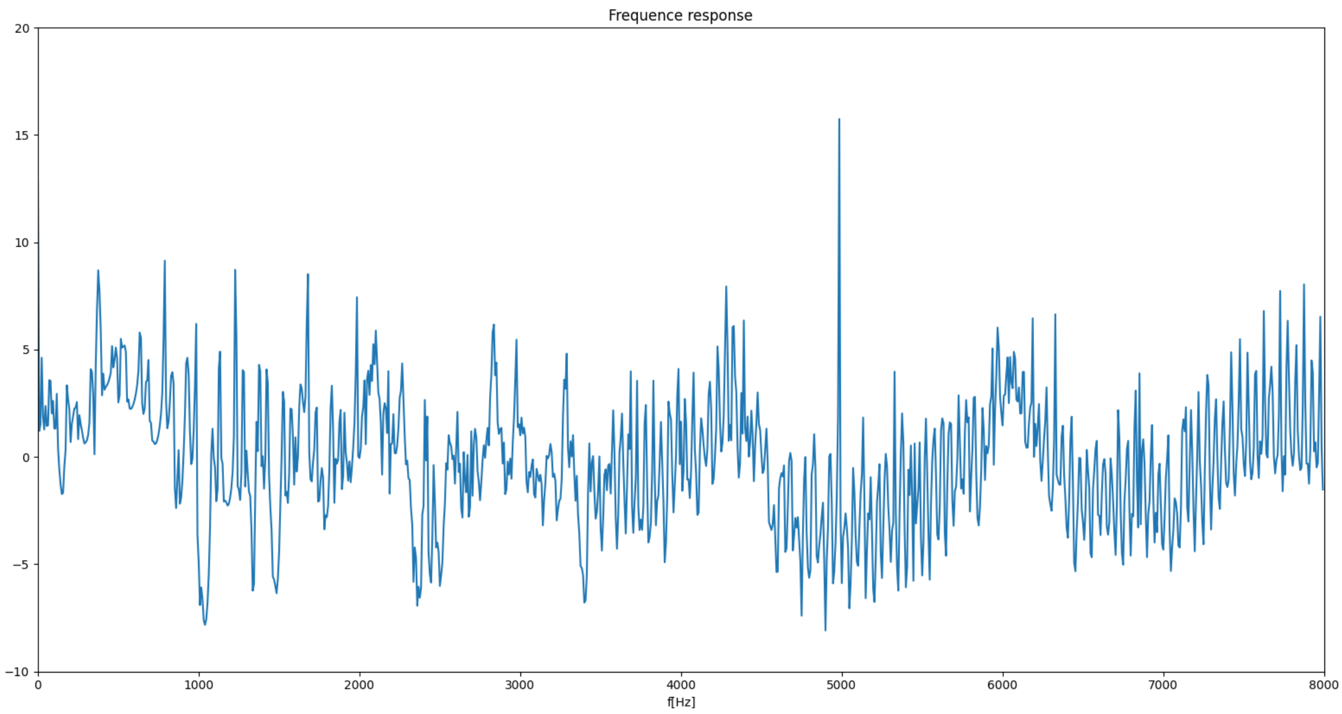


5 Úloha 6 - Frekvenční charakteristika roušky

Frekvenční charakteristika je dána následujícím vztahem:

$$H(e^{j\omega}) = \frac{\sum_{k=0}^{M} Y[k]e^{-j\omega k}}{\sum_{l=0}^{N} X[k]e^{-j\omega k}}$$

Výsledná frekvenční charakteristika:



Z frekvenční charakteristiky filtru nelze jednoznačně popsat o jaký typ filtru se jedná. Vidíme například potlačení frekvence okolo 1KHz.

6 Úloha 7 - Impulzní odezva

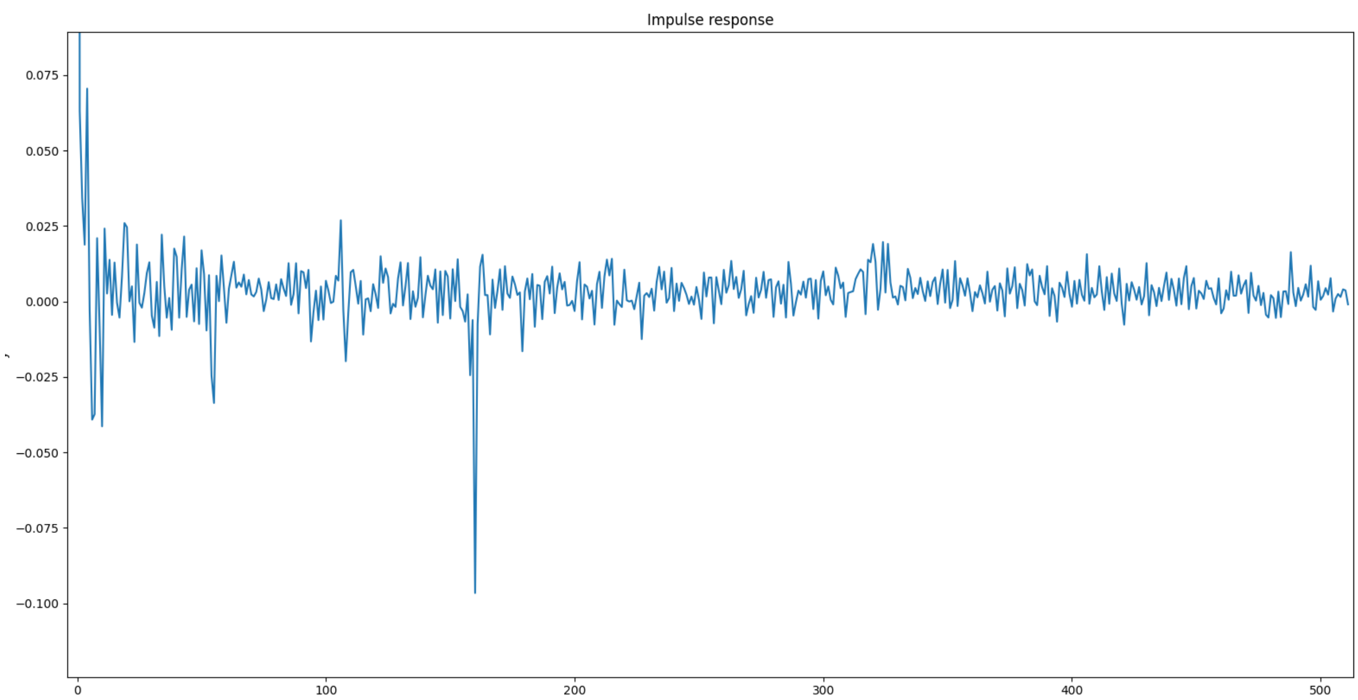
Aplikací inverzní diskrétní Fourierovy transformace(IDFT) na frekvenční charakteristiku dostaneme impulzní odezvu.
Vzorec pro výpočet IDFT:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi\frac{k}{N}n}$$

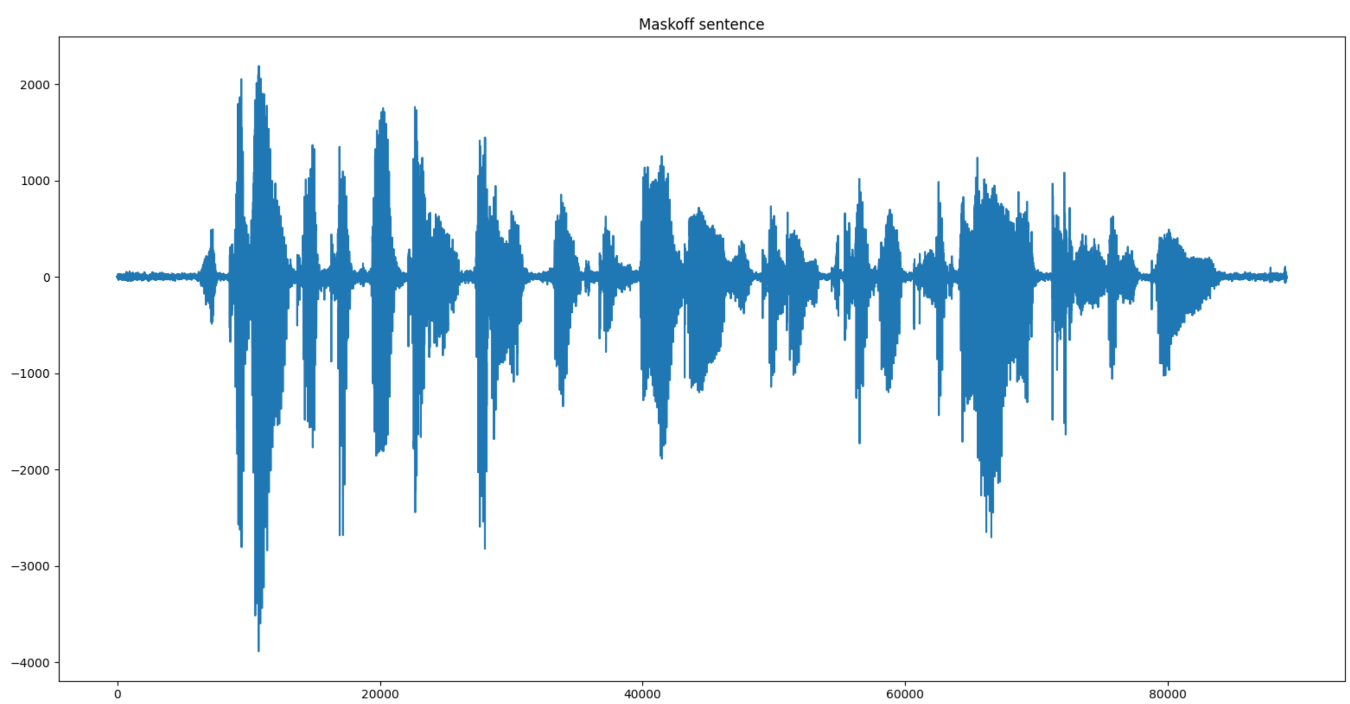
Implementace IDFT:

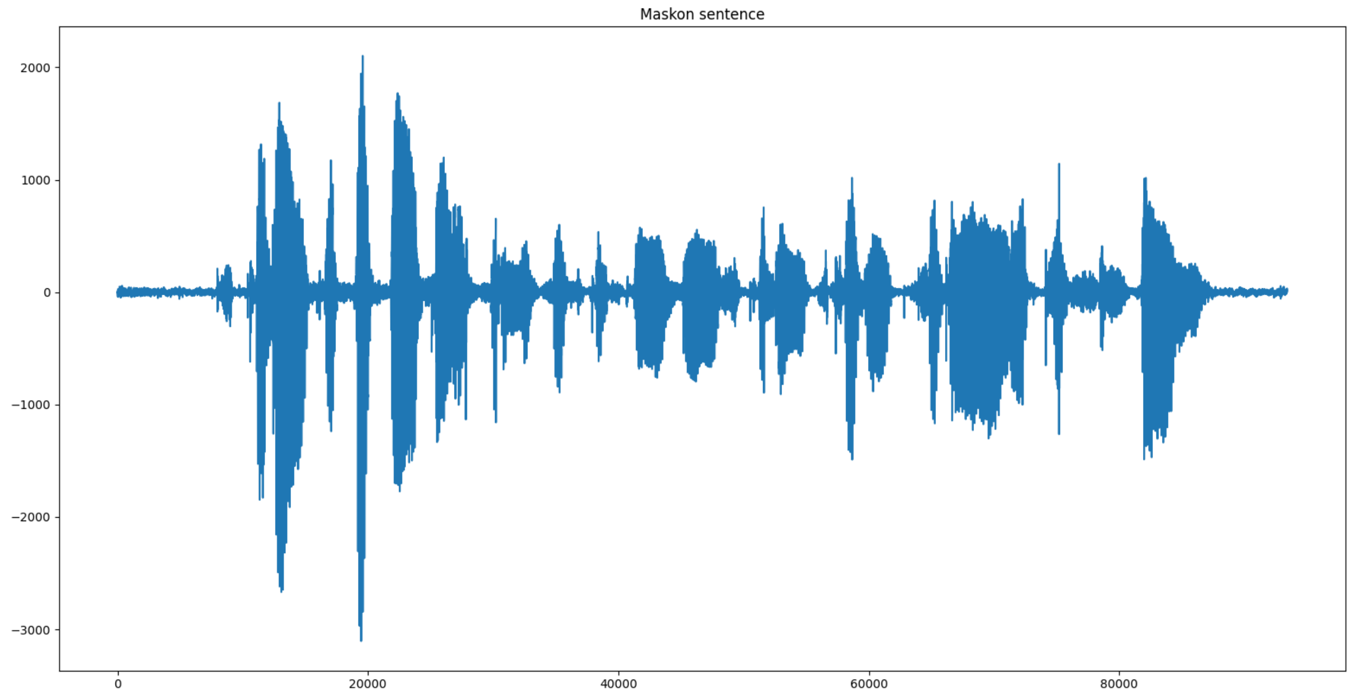
```
for n in range(N):
       for k in range(N):
           h[n] += HjwAvg[k]*(np.exp(1j*2*np.pi*(k/N)*n))
       h[n] /= N
```

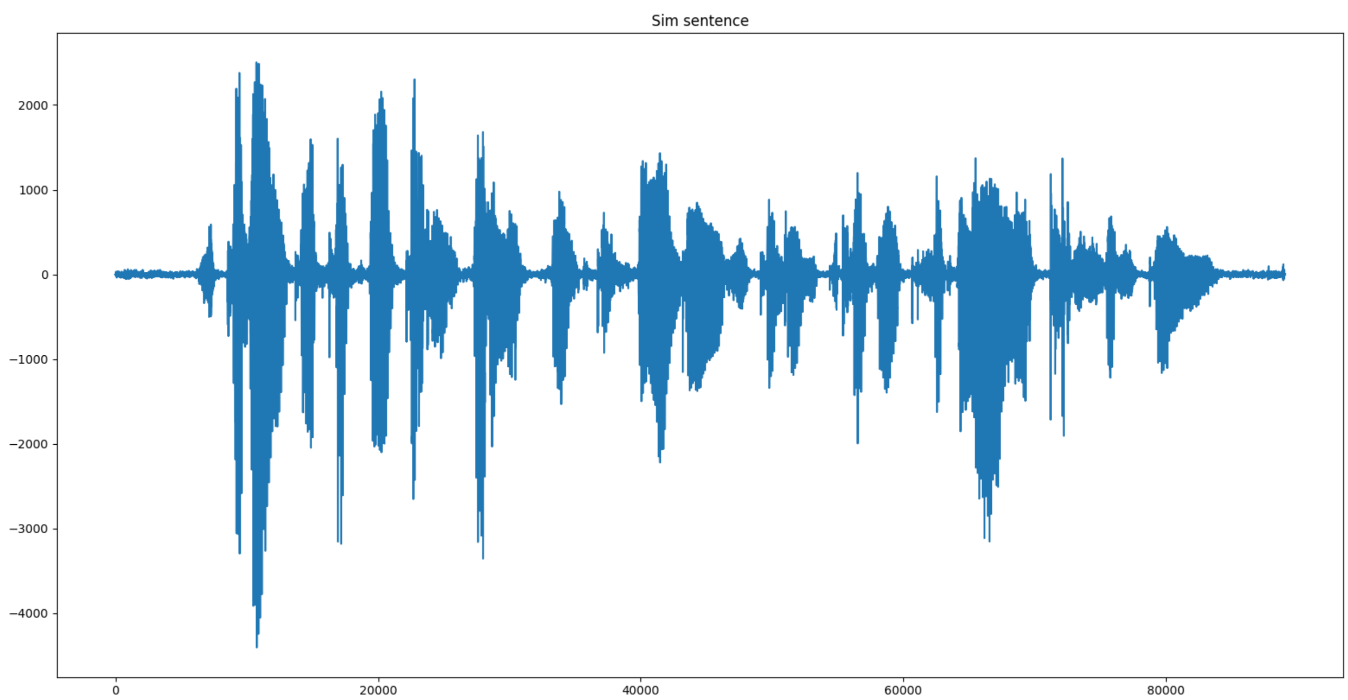
Graf impulzní odezvy:



7 Simulace roušky







Simulovaná rouška se více podobá původnímu signálu bez roušky.

8 Závěr

Na začátku práce s projektem jsem odhadoval roušku na dolní propusť. Výsledná nahrávka simulované roušky je srozumitelná, nicméně není identická s originální rouškou, hlas je zašumělý. V průběhu výpočtu se počítalo s komplexními čísly a čísly s plovoucí řádovou čárkou, kde mohlo dojít ke zkreslení. Problém mohl vzniknout i při tvorbě nahrávek, které nebyly nahrány kvalitním mikrofonem a problémem držet jeden konkrétní tón. V průběhu zpracovávání protokolu musela být nahrávka několikrát vyměněna.