ISS Projekt 2020 / 21

Leden 04/2020

Vypracoval: Vladyslav Tverdokhlib
xtverd01@stud.fit.vutbr.cz

1. Nahrajeme tóny na diktafonu smartphonu a upravíme format.

maskoff_tone.wav	05:10 *sec.* / 81579 *samples*
maskon_tone.wav	04:95 *sec.* / 79189 *samples*

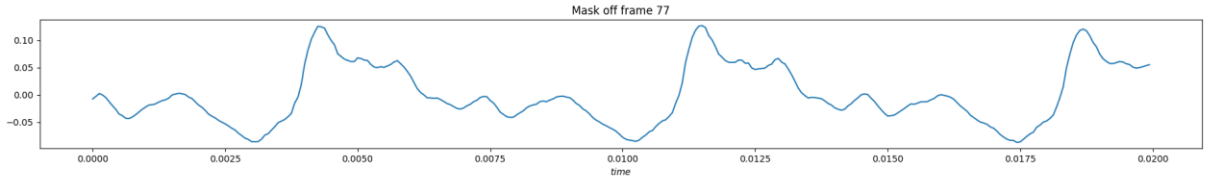
2. Nahrajeme věty na diktafonu smartphonu a upravíme format.

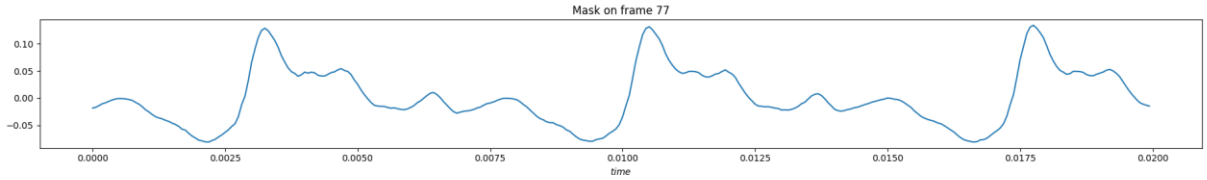
maskoff_sentence.wav	03:09 *sec.* / 49493 *samples*
maskon_sentence.wav	03:56 *sec.* / 57003 *samples*

3. Pro načítání použijme knihovnu soundfile, která automaticky normalizuje hodnoty signalu.

   Velikost rámce = *ms* * *Fs* = 0.02 * 16000 = 320 samples.

   Kreslení rámců budeme vykonávat pomocí knihovny *matplotlib.pyplot*.





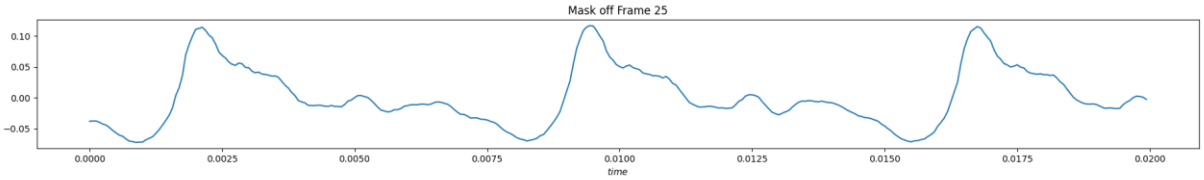
4. Aplikujeme metodu center clipping podle zadání.

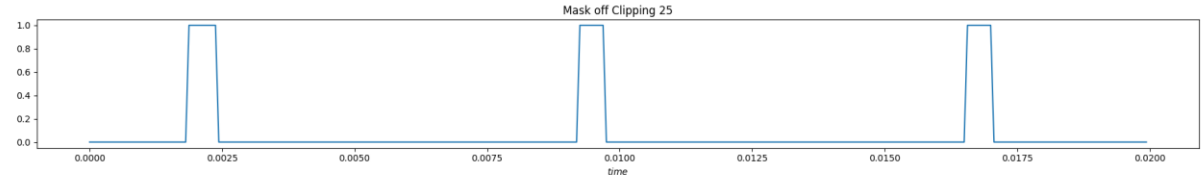
   Spočítáme autocorrelation pomocí vzorku[1]: $Y_{ff}[k] = \sum_{n=-\infty}^{\infty} f[n]f[n-k]$

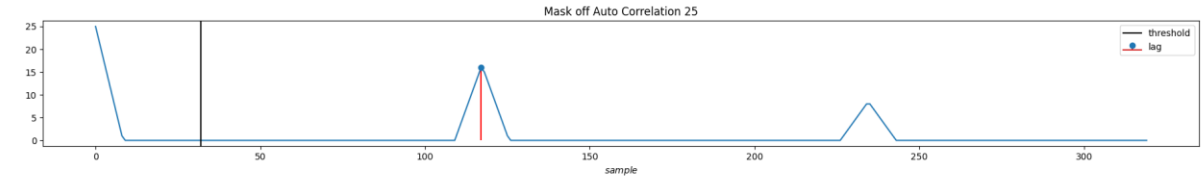
   Aplikujeme práh *500hz* (Fs=16000 => 32 samples), pro každý autokorelační rámec najdeme lag a převedeme na frekvenci *(fs/lag*).

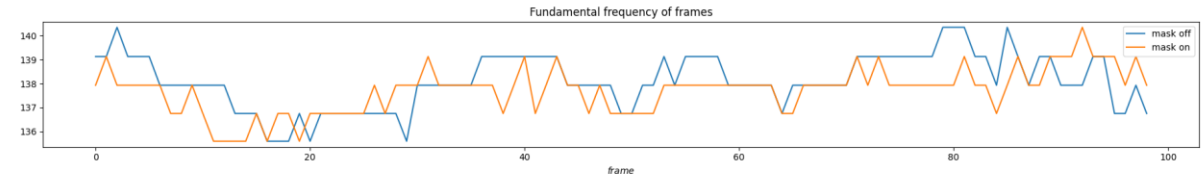
   Pomocí frekvencí nakreslíme srovnávající graf a zároveň vypočítáme průměr a rozptyl frekvencí.

(a)









(b)

	Mask off	*Mask on*
Průměr	138,1206	137,7097
Rozptyl	1,2680	0,8963

(c) Velikost změny by se dala zmenšit do 1. Protože hodnoty v našem rozsahu lag(114, 117) jsou skoro odmocniny Fs=16000.

5. Implementujeme vlastní funkci počítající DFT.

   Aplikujeme DFT pro každý rámec s N = 1024.

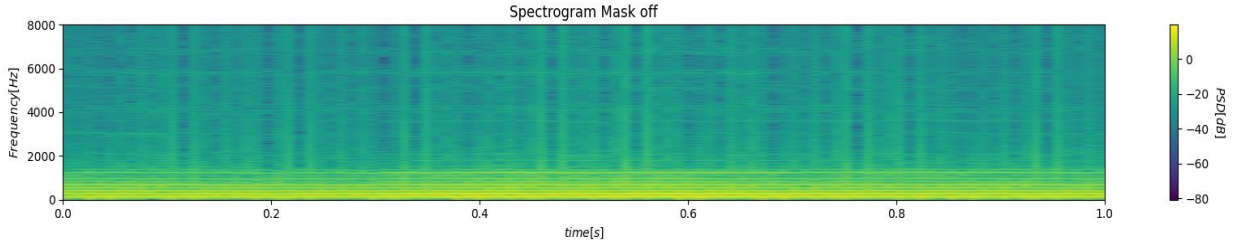
   Všechny hodnoty upravíme pomocí daného logaritmického vzorce.

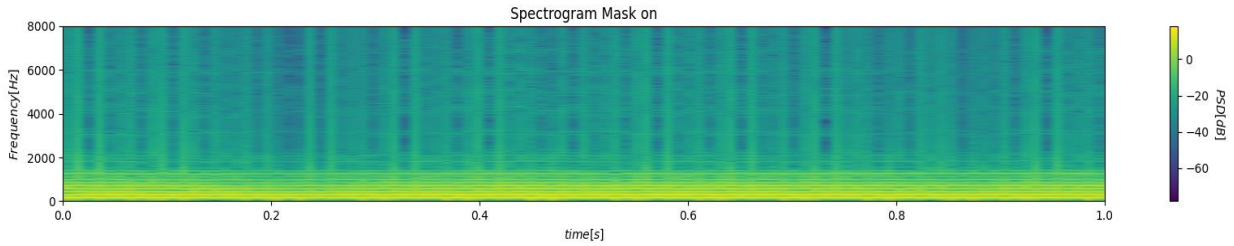
Spektrogramy vykresleme pouze pro k = [0..512]

(a) Implementace DFT podle vzorce[2]: $X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi kn}{N}}$

```
def DFT(frame):
    frame = np.asarray(frame, dtype=complex)  # nastaveni typu pro
spravny vypocet
    size = frame.size
    dft = np.ndarray(size, dtype=complex)  # vysledna dft funkce

    for q in range(size):
        s = 0
        for j in range(size):
            s += frame[j] * np.exp(-2j * np.pi * j * q * np.divide(1,
size, dtype=complex))  # vzorec
        dft[q] = s

    return dft
```

(b)





6. Rozdělíme DFT rámce nahrávky s rouškou na DFT rámce nahrávky bez roušky.

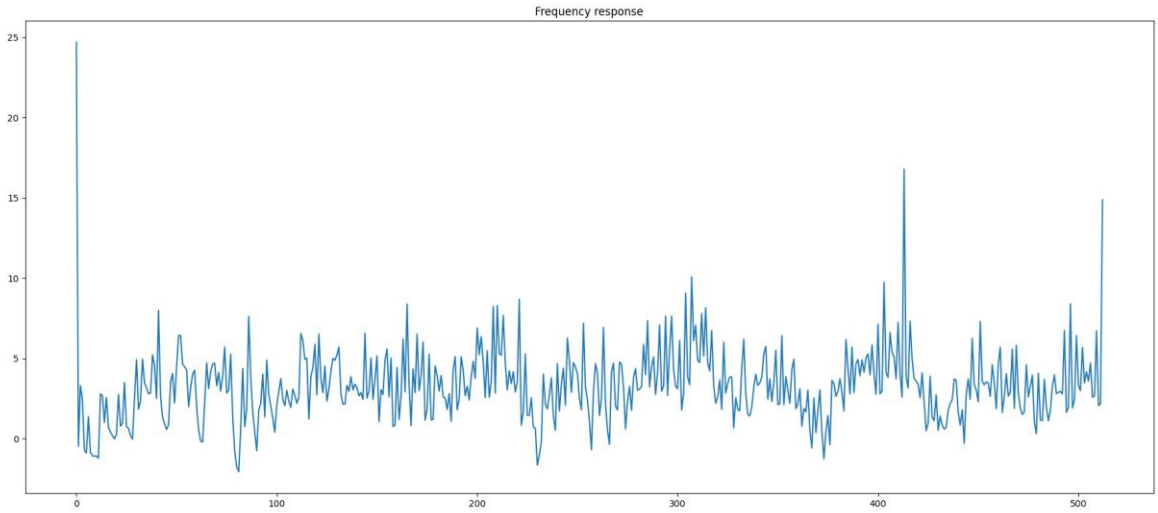
   Zprůměrujeme do jedné funkce s absolutnimi hodnotami.

   Aplikujeme logaritmický vzorek pro výkonové spektrum.

   Vykreslime půl funkce kvůli symetrii.

   (a) H(e^jw) = *Y(e^jw) / X(e^jw)*

   (b)

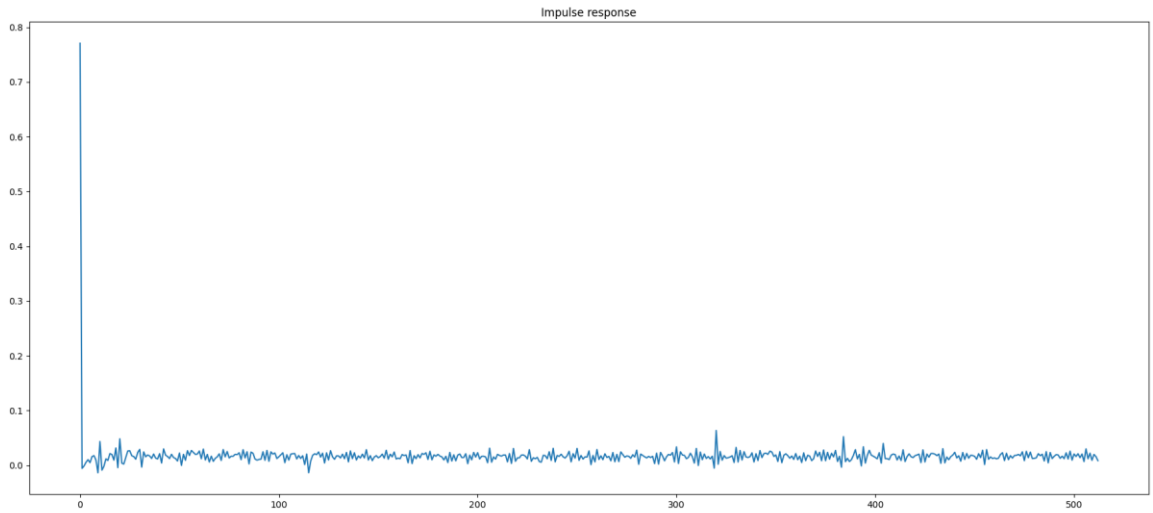


(c) (x = Hz, y = dB). Filtr není stabilní. Nemužeme zdůraznit dobrý úsek frekvence. Jsou několik drsných zvuku. Střední hodnota je malá – 5-7 dB.

7. (a) Implementace IDFT podle vzorce[2]: $$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi kn}{N}}$$

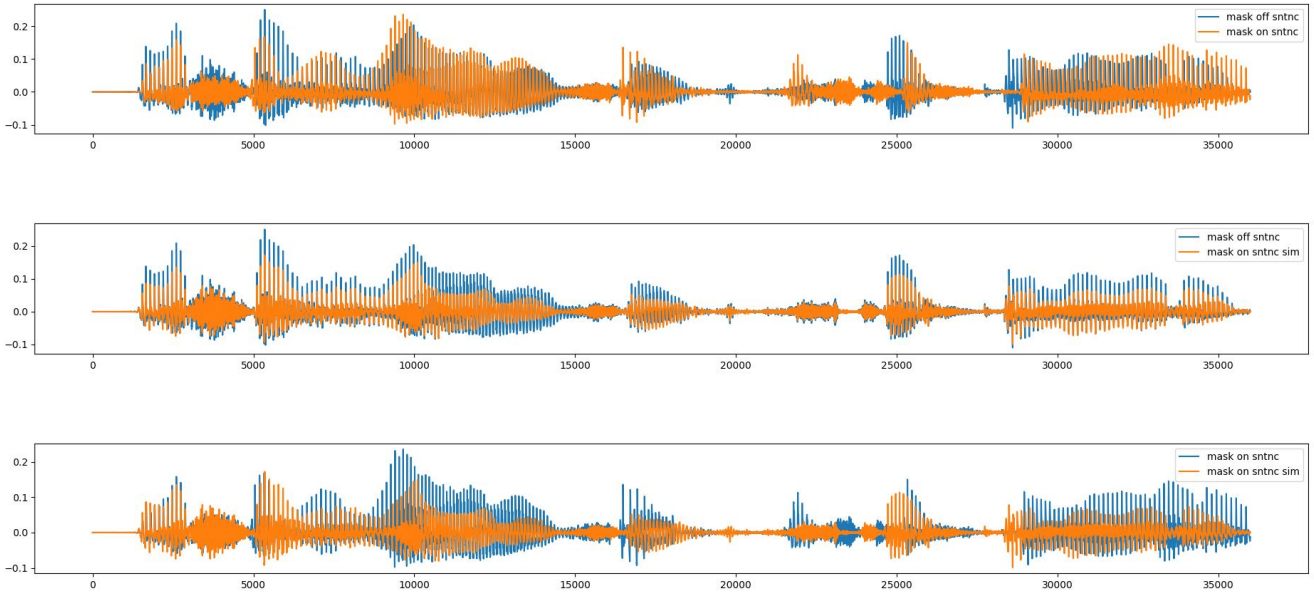
```
    def IDFT(dft):
        dft = np.asarray(dft, dtype=complex)  # nastaveni komplexniho typu
        size = dft.size
        idft = np.ndarray(size, dtype=complex)

        for q in range(size):
            s = 0
            for j in range(size):
                s += dft[j] * np.exp(2j * np.pi * q * j * np.divide(1,
size, dtype=complex))  # zakladni cast inverzniho vzorce DFT
            idft[q] = np.divide(s, size, dtype=complex)  # deleni na N

        return idft
```

(b)



8. Provádíme simulaci pomocí *scipy.signal.lfilter*.

(a)



(b) Na třetím grafu signály jsou nejvíce podobné na nízkých hodnotách. Simulovaná rouška má jakoby omezení do hodnoty 1,5, proto se nejvíce liší na při vysokých hodnotách. Na 20 000-21 000 frekvencích simulované roušky vidíme vůbec ztracený signál ve srovnání s původní rouškou.

9. Závěr

Mám nevýkonnou funkci počítající DFT, spočítání trvá asi 10 min. Myslím, že na našich slabých vlastnostech signálů vhodně používat FFT.

Nepříjemná je frekvenční charakteristika. Používal bych studiové zařízení nebo speciální programy.

Můžeme vidět, že původní filtr skoro ne filtruje řeč ve 8. na prvním grafu, a to může vést k problémům.

Koeficienty roušky nevypadají přesně na porovnání.

Věřím, že moje řešení mírně funguje.

---

1. http://ssl-iitg.vlabs.ac.in/Signal%20and%20their%20properties%205(theory).html
2. https://www.slideshare.net/parveztrina/dsp-lecture-vol-2-dft-fft