Úloha 1.

Názov	Dĺžka	Počet vzorkov
maskoff_tone.wav	00:00:07.27	116395
maskon_tone.wav	00:00:07.19	115029

Úloha 2.

Názov	Dĺžka	Počet vzorkov
maskoff_sentence.wav	00:00:07.06	112981
maskon_sentence.wav	00:00:07.06	112981

Úloha 3.

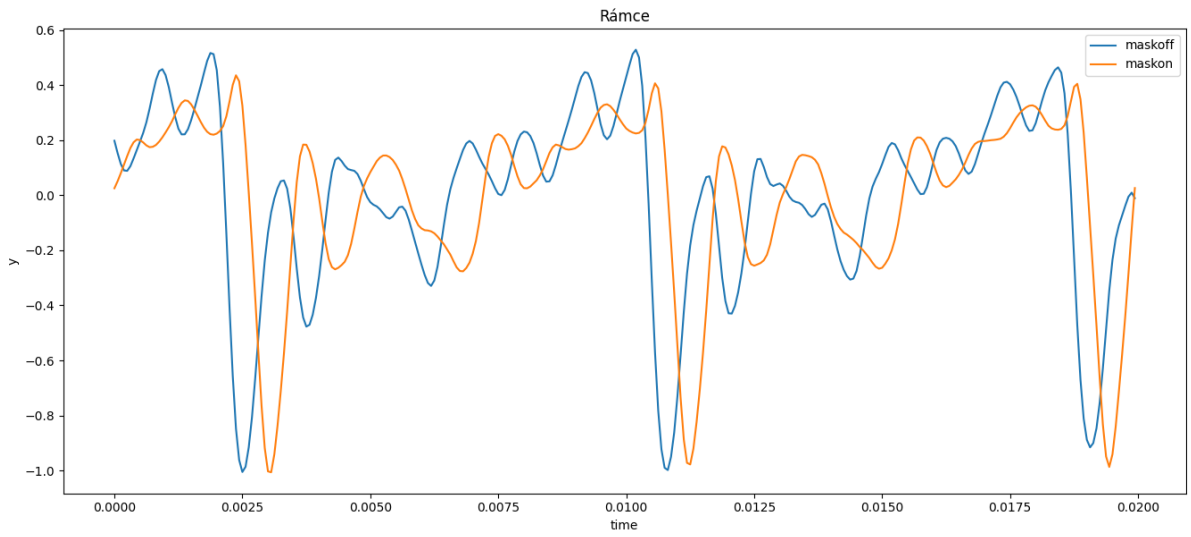
Výpočet pre dĺžku jedného rámca vo vzorkoch:

h = čas [sec] * vzorkovacia frekvencia [hz]

Pre konkrétne zadanie:

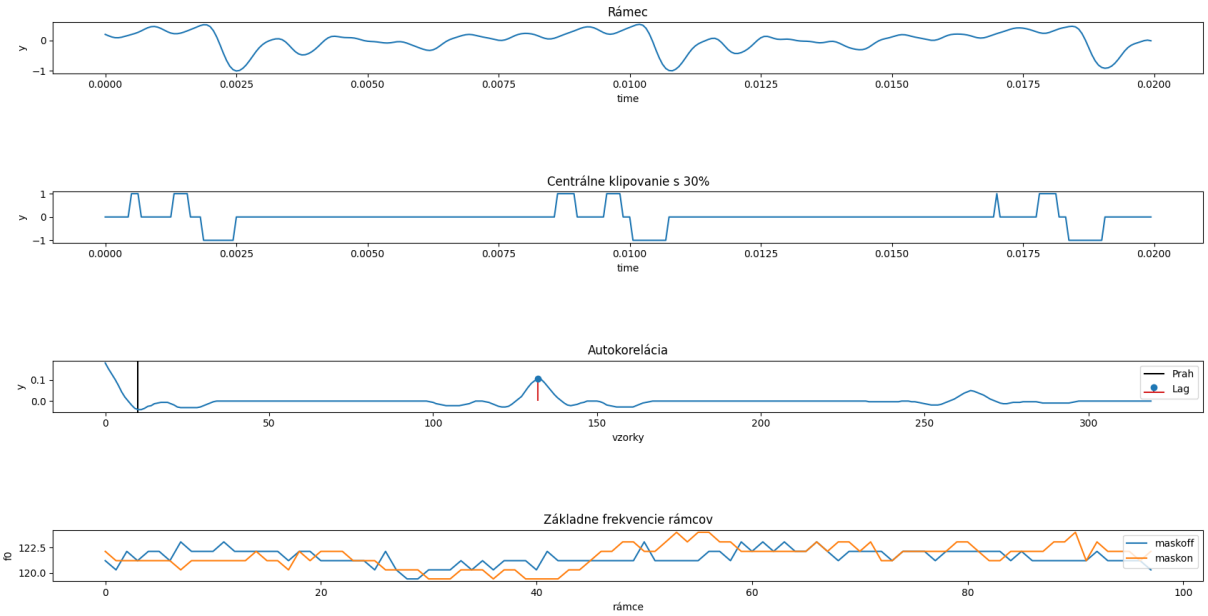
h = 0.02 * 16000 = 320 vzorkov

Graf:



Úloha 4.

Pri centrálnom klipovaní som musel znížiť prah až na 30% pretože väčšie hodnoty dokázali orezať signál natoľko že z neho skoro nič neostalo. (napr isté pasáže nahrávky mali max amplitúdu 0.6 po nomalizácii takže s prahom 70% by som dostal akurát 0 všade)



Stredná hodnota maskoff = 121.564

Rozptyl maskoff = 0.598

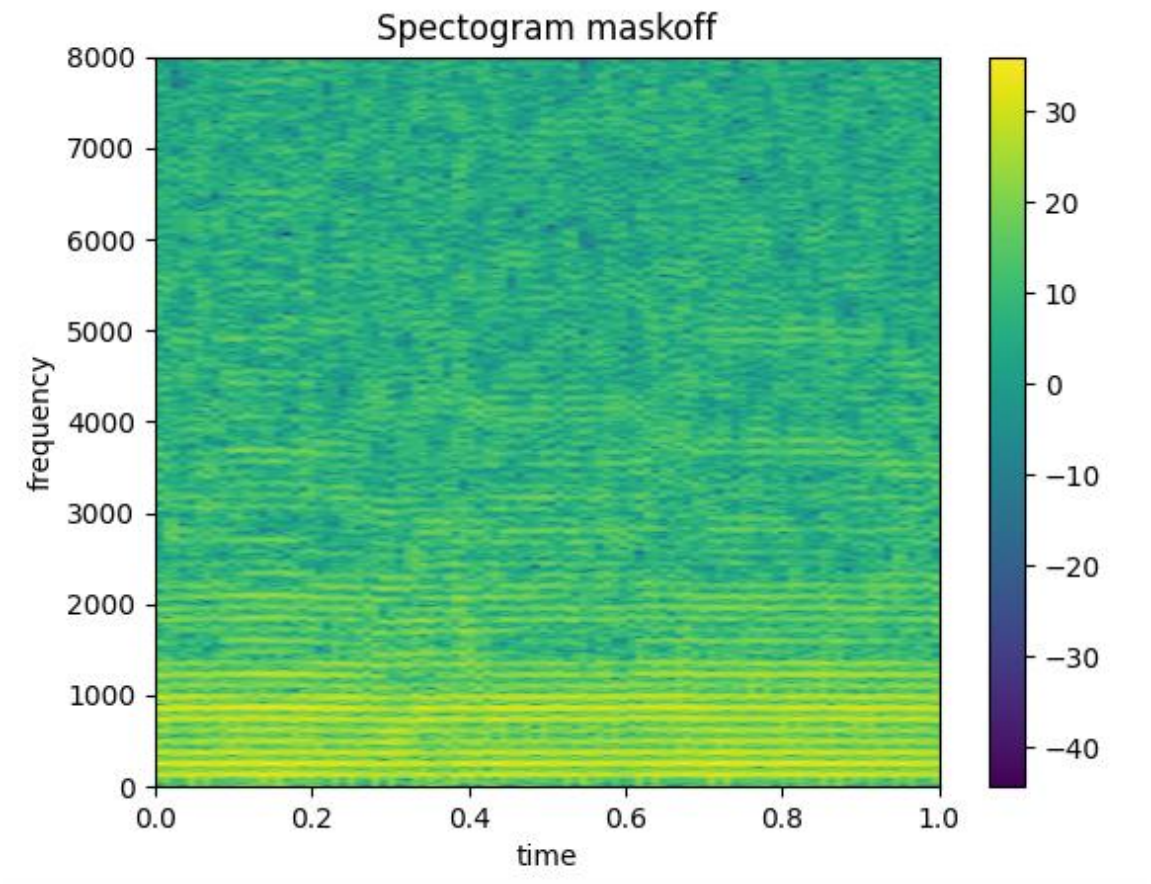
Stredná hodnota maskon = 121.655

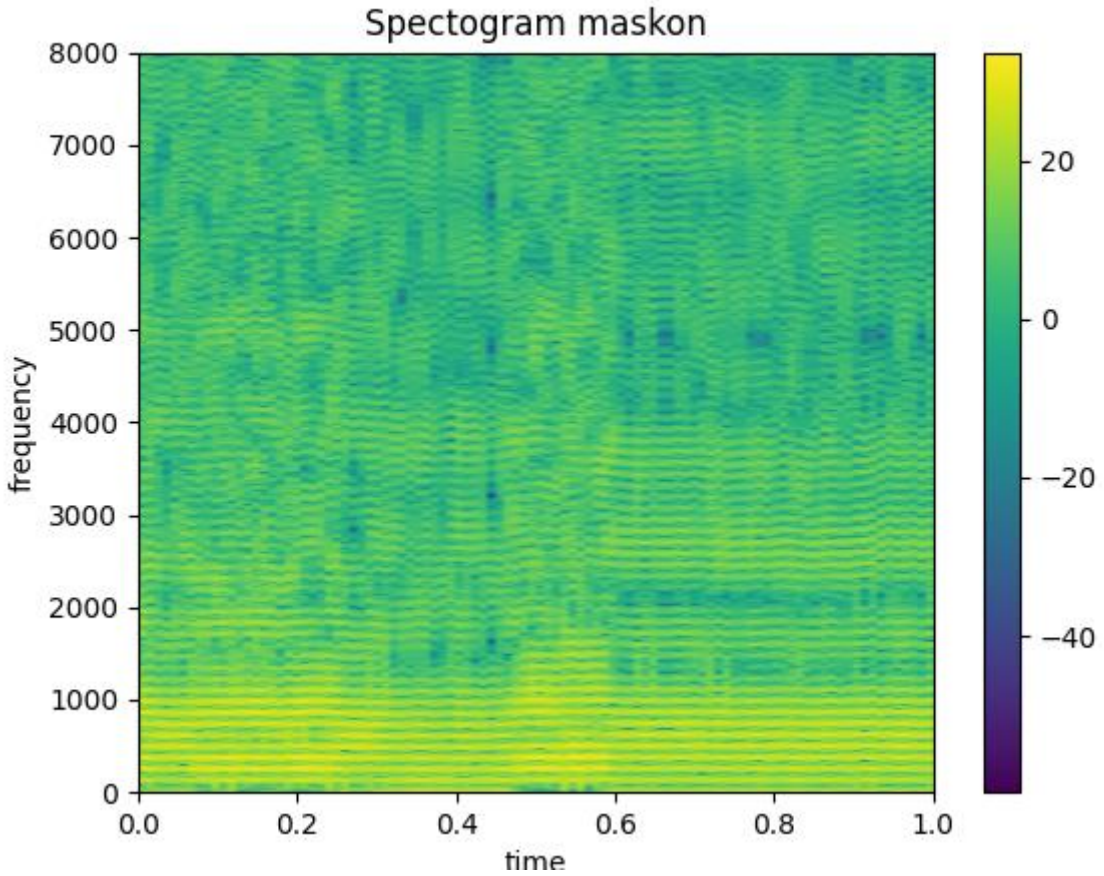
Rozptyl maskon = 1.325

Úloha 5.

Funkcia implementujúca rýchlu DFT (pozn. táto funkcia je dosť pomalá)

```
def my_dft(arr,N):

    if len(arr)< N:
        arr = np.pad(arr, (0,N-len(arr)),"constant")

    vys = []
    for i in range(0,N):
        vys.append((0))
        for j in range(0,N):
            vys[i] += arr[j] * np.exp(np.complex(0,-2*np.pi*i*j/N))

    return np.array(vys)
```





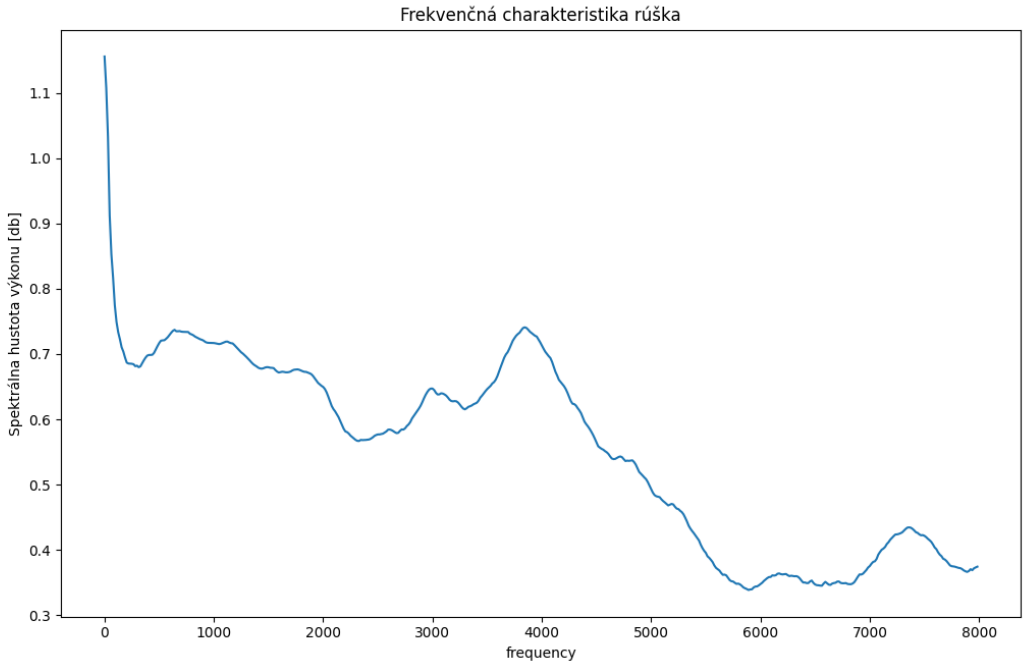
Úloha 6.

Použil som nasledovný vzorec

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{Q} b_k z^{-k}}{1 + \sum_{k=1}^{P} a_k z^{-k}} = \frac{B(z)}{A(z)},$$

$$H(e^{j\omega}) = H(z)|_{z=e^{j\omega}}$$

Pričom som ale ešte pred tým ako som čísla podelil aplikoval filter na koeficienty aby som ich jemne “vyžehlil“ (bez tohto kroku zbytočne vznikali prípady kedy sa zbytočne delilo velké číslo malým a výsledná charakteristika bola zašumená)

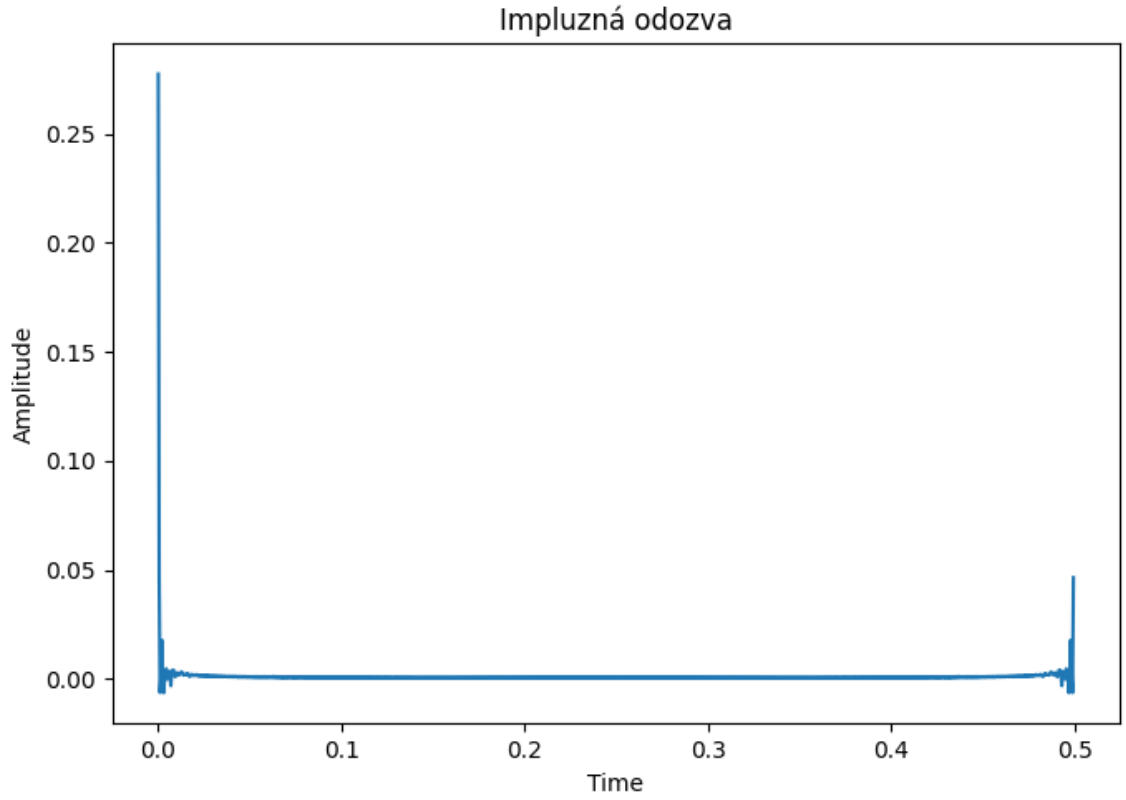


Filter potláča vysoké frekvencie viacej ako nízke, pričom toto je očakávaný výsledok.

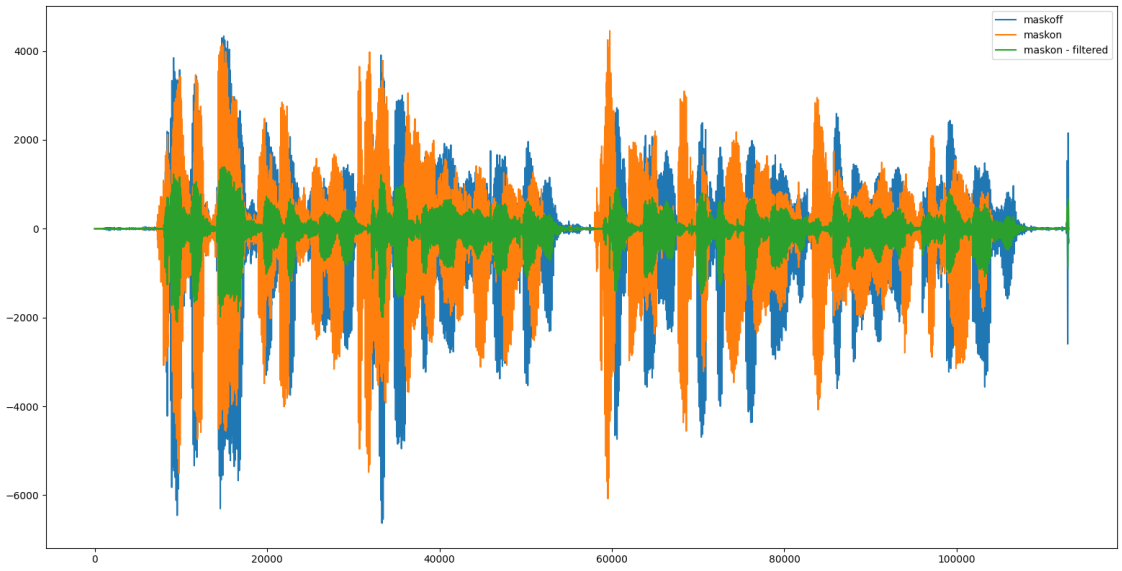
Úloha 7.

Funkcia implementujúca IDFT

```
def my_idft(arr,N):

    if len(arr)< N:
        arr = np.pad(arr, (0,N-len(arr)),"constant")


    vys = []
    for i in range(0,N):
        vys.append((0))
        for j in range(0,N):
            vys[i] += arr[j] * np.exp(np.complex(0,2*np.pi*i*j/N))

    return np.array(vys)/N
```



Úloha 8.



Simulovaný signál je viacej tichší ako reálny signál, povedal by som že signály sa najviac podobajú pri nízkych frekvenciach s nízkou amplitúdou a najviac sa líšia pri vysokých frekvenciach s vysokou amplitúdou.

Úloha 9. – záver

Myslím si že moje riešenie do istej miery funguje a je vidno istý výsledok.

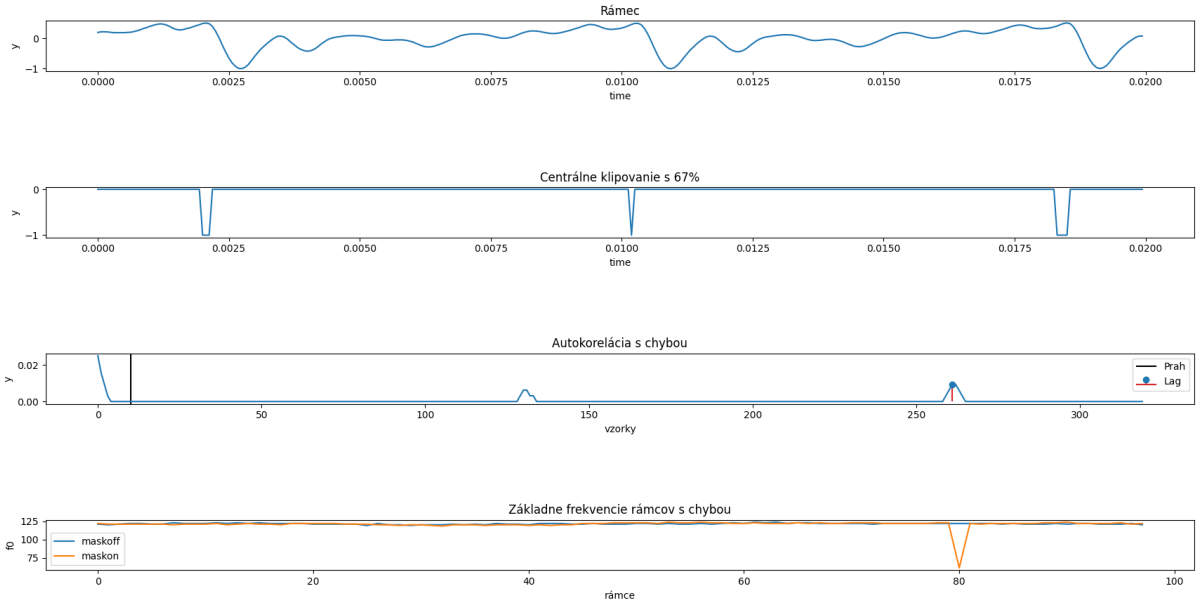
Mám za to že ak by som mal viacej a kvalitnejších dát tak by som mohol odhadnúť filter oveľa presnejšie (napr. viem si predstaviť mať rúšku cez reproduktor ktorý by prehrával frekvencie od 20Hz do 20kHz a prefiltrovný signál by som nahrával – tým by som vedel spočítať filter lepšie).

ISS project, Adam Fabo [xfaboa00]

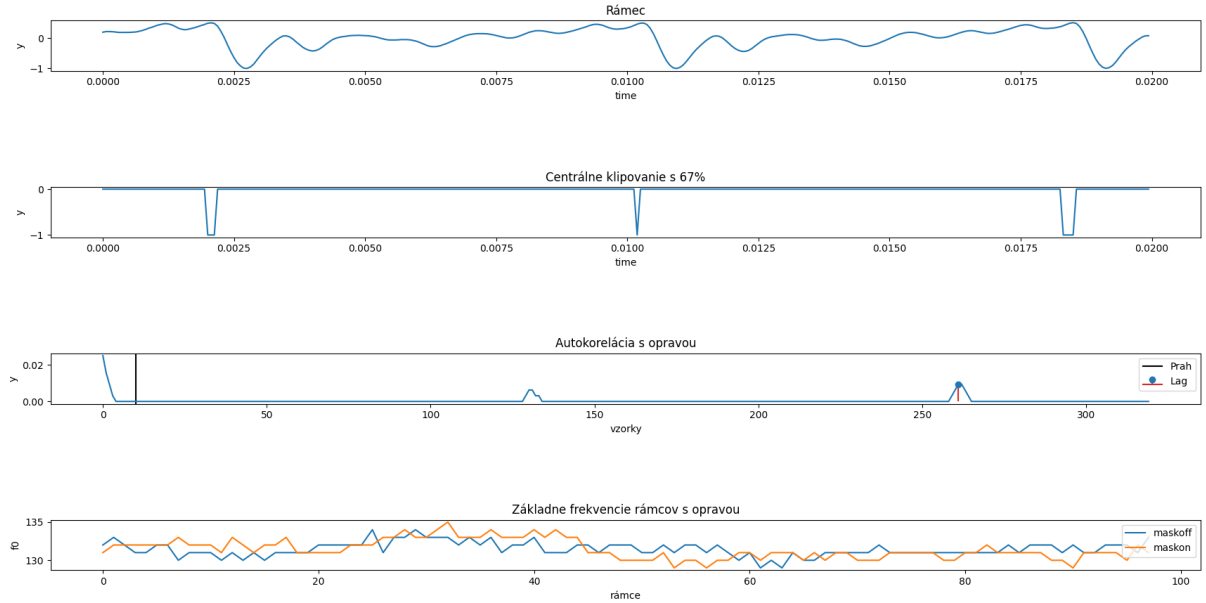
Doplňujúce úlohy

Úloha 12.

S chybou:



S opravou:



Detekcia chyby:

Pre to aby chyba mohla byť detekovaná musia byť najprv spočítané všetky základné frekvencie rámcov a z nich treba vypočítať priemernú hodnotu základnej frekvencie pre všetky rámce. Keďže očakávam že všetky základné frekvencie rámcov budú mať približne rovnakú hodnotu tak si zvolím istý prah nad a pod ktorým budem považovať hodnoty za chybné. Prah som zvolil na 25% z priemernej frekvencie, čiže ak je priemerna frekvencia 100hz tak hodnoty nad 125 a pod 75 budem považovať za chybné a hodné opravy.

Oprava chyby:

Keď už mám chybu detekovanú tak ju opravím tak, ako je napísané v opore ZRE a to nelineárnou filtráciou medíánovým filtrom.

7.6.1 Nelineární filtrace mediánovým filtrem

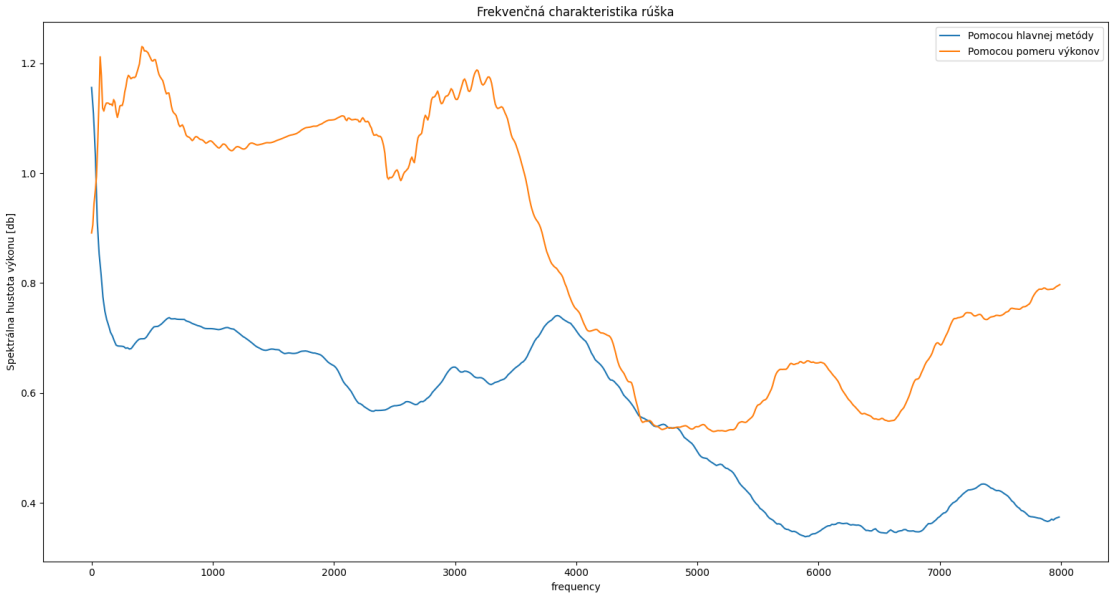
se provádí podle vztahu:

$$L(i) = \text{med}\,[L(i-k), L(i-k+1), \ldots, L(i), \ldots, L(i+k)] \qquad (7.24)$$

Medián seřadí hodnoty podle velikosti a vybere hodnotu, která se nachází uprostřed. Lagy z našeho příkladu tedy budou opraveny na 50, 50, 50, 50, 50.

Pri tejto metóde ešte treba taktiež zvoliť koeficient K, ktorý som si ja zvolil 10.

Úloha 14.



Myslím si že na obrázku je vidno že oba filtry tlmia viacej vyššie frekvencie ako nižšie. Ako si môžete všimnúť filter vypočítaný pomocou pomeru výkonov je vyššie v grafu ako filter z hlavnej úlohy, pričom dôvod za týmto nepoznám.

Pri počítaní filtru som sa inšpiroval linkom ktorý bol pri zadaní, šírka pásma pre ktoré som počítal výkon bola vždy 20hz pričom som počítal od 20hz do 8khz. Pri výpočtoch som využíval Simpsonov integrál.

Úloha 15.

Túto úlohu som neimplementoval ale rád by som odpovedal na bonusovú otázku.

Lag vlastne hovorí o perióde (T) signálu ale vo vzorkoch. Majme signály ktoré sú periodické a majú približne rovnakú základnú frekvenciu.

Na to aby sme získali fázový posun „doľava“ tak musíme nájsť začiatok najbližšej periódy doľava a odčítať ho od začiatku periódy posunutého signálu.

Na to aby sme získali fázový posun „doprava“ tak musíme nájsť začiatok najbližšej periódy doprava(čo je vlastne koniec periódy čo začína vľavo) a odčítať od neho začiatok periódy posunutého signálu.

Týmto spôsobom vlastne dostaneme vzdialenosť vo vzorkoch na začiatok periódy a na konec periódy – tým pádom keď ich sčítame tak dostaneme dĺžku jednej periódy čo je vlastne lag.