VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISS Projekt 2021/22

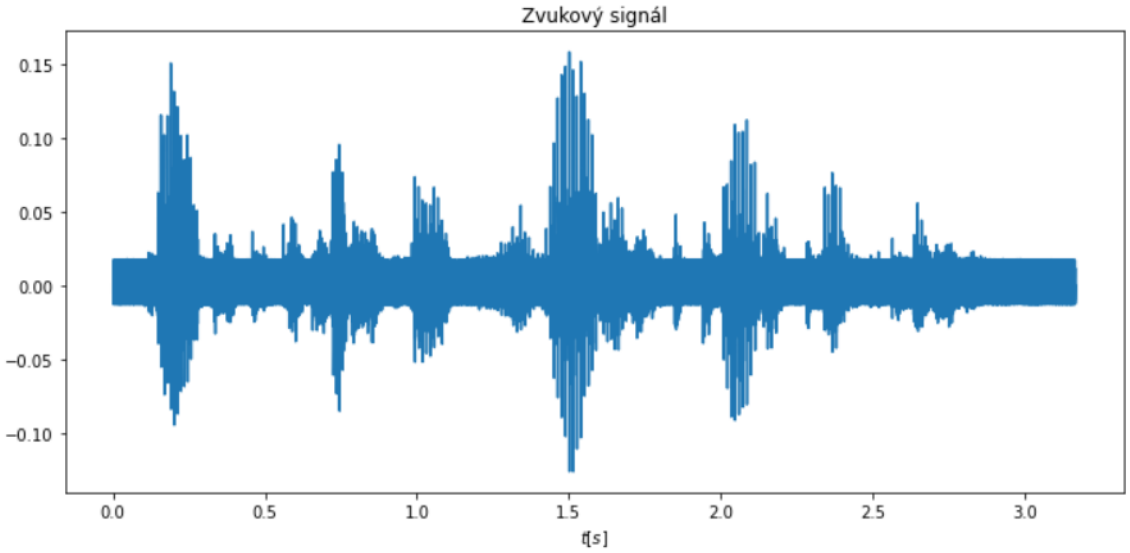
7. ledna 2022

Simona Češková (xcesko00)

1 Základy

Můj načtený signál pomocí funkce `wavfile.read` má vlastnosti:

- Délka ve vzorcích: 50688
- Délka v sekundách: 3.168
- Maximální hodnota: 2530
- Minimální hodnota: -2015

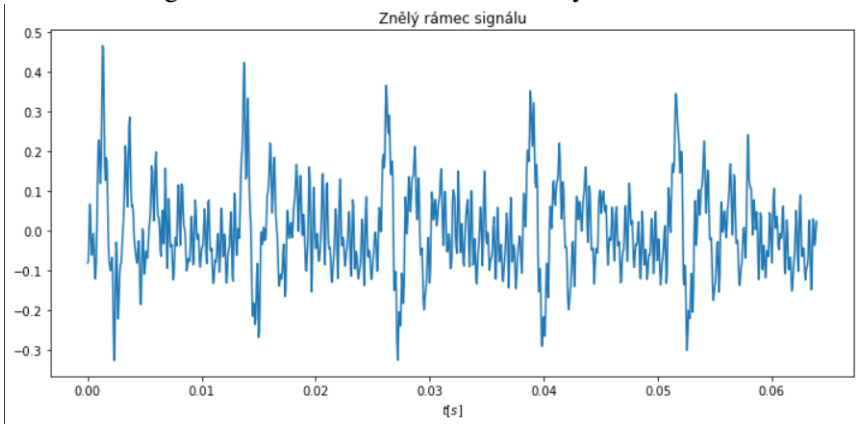


2 Předzpracování a rámce

Ustřednění signálu jsem udělala pomocí funkce `np.mean(DATA)`, která mi vypočte střední hodnotu. Normalizováním do dynamického rozsahu jsem získala interval $(-0,7964283993204706; 1,0)$.

Aby nedošlo k vynechání posledního vzorku, tak jsem zaokrouhlila nahoru pomocí funkce `math.ceil(DATA.size/frame)`, kde zároveň rozděluji signál na rámce o 1024 vzorcích.

Ukázka signálu v daném intervalu s rozdělenými rámci:



3 DFT

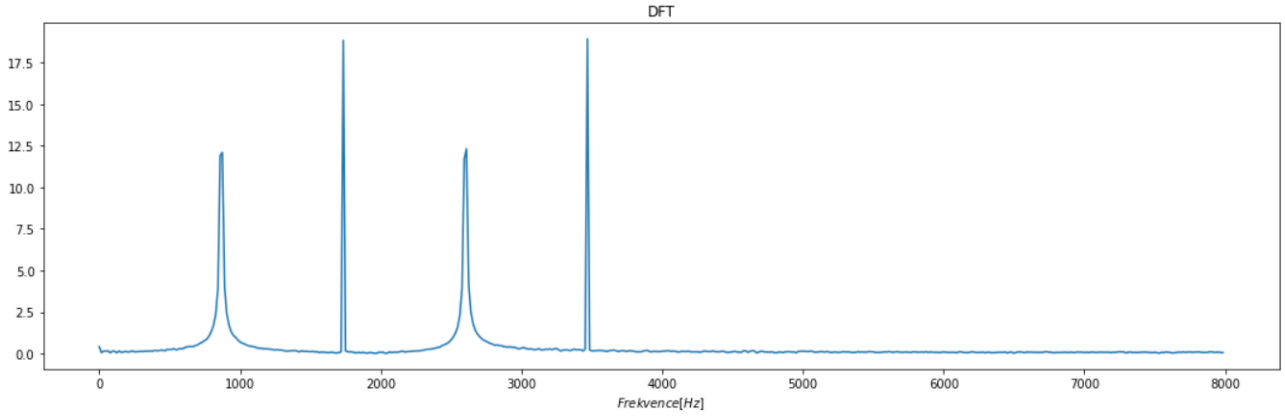
Implementovala jsem funkci `DFT_function` podle vzorce, který jsem našla na stránce (odkaz).

Spustila jsem funkci na vybraném nejvíce perodickém rámci a zobrazila modul DFT na půlce frekvence, aby se odstranila dvonásobnost rušivých frekvencí. Porovnala jsem svůj výsledek s knihovní implementací:

`np.allclose(DFT_MATRIX.real,(np.fft.fft(MATRIX[0])).real)`

Výsledek vyšel stejně, tak jako v mém řešení.

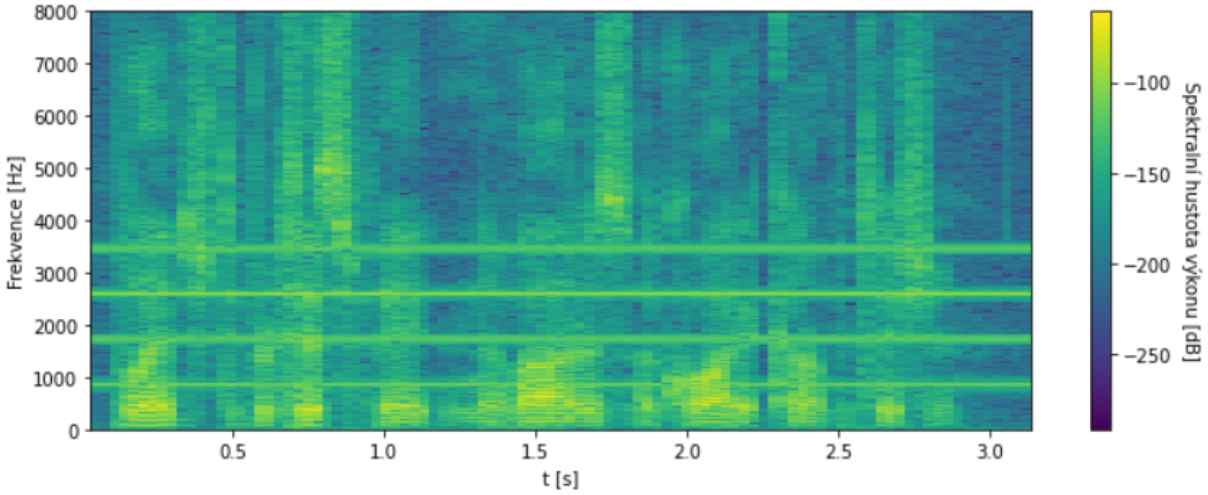
Modul DFT pro frekvence $< 0; 8000 >$:



4 Spektrogram

Využila jsem funkci `spectrogram(DATA, FS, nperseg=1024, noverlap=512)`, kde nperseg je délka okna a noverlap je překrytí. Dále jsem upravila hodnoty jednotlivých koeficientů DFT pomocí vzorce ze zadání `sgr_log = 10 * np.log10((sgr+1e-20)**2)`

Logaritmický výkonový spektrogram:



5 Určení rušivých frekvencí

Rušivé frekvence jsem našla díky 3. úkolu, kde jsem podle grafu zjistila, že právě jen ty rušivé frekvence sahají k vysokým hodnotám. Proto jsem zvolila např hodnotu 12, kterou přesáhnou pouze ony.

Ukládám je postupně do pole `cos_freqs.append(F[i])`, které po zavolání funkce `cos_collect()` pak uložím do matice `cos_matrix`.

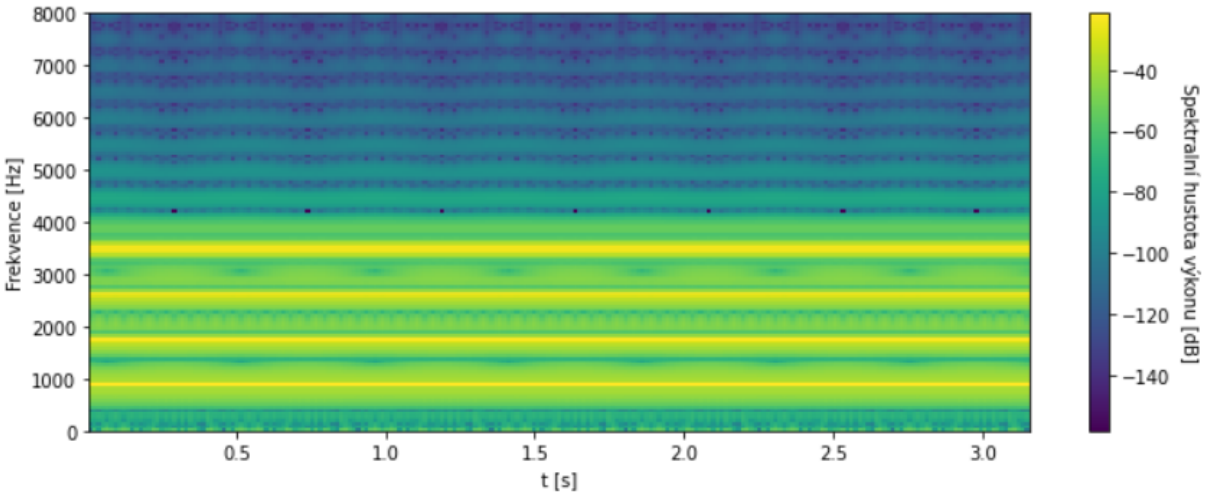
- f_1 = 875.0 [Hz]
- f_2 = 1734.375 [Hz]
- f_3 = 2609.375 [Hz]
- f_4 = 3468.75 [Hz]

Vypočítané rušivé frekvence f_1, f_2, f_3 a f_4 mně vyšly jako násobky s lehkou odchylkou.

6 Generování signálu

Nejdříve jsem si vytvořila pole `cos_omegas = []`, kam jsem ukládala postupně 4 cosinusovky, které jsem spočítala podle vzorce a potom je sečetla dohromady do proměnné `cos_data`. Tuto proměnnou jsem poté uložila do souboru `"../audio/4cos.wav"`. V porovnání se spektrogram v úkolu 4, lze vidět, že rušivé frekvence jsou stejné.

Výsledný spektogram rušivých signálů:



7 Čisticí filtr

7.1 Výroba filtru v z-rovině

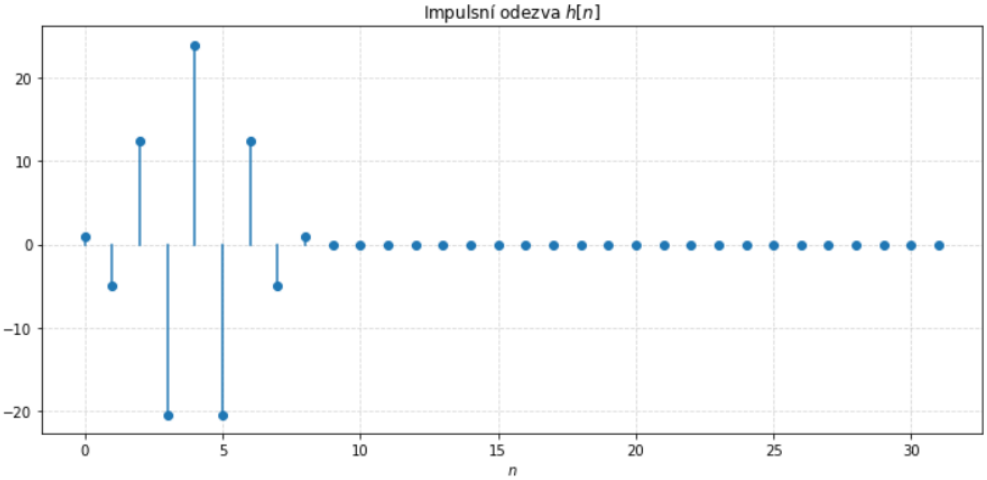
Na potlačení rušivých frekvencí jsem si vybrala pásmovou zádrž 7.1

Postupovala jsem tak, že na začátku jsem našla normovanou kruhovou frekvenci
`omega_filter = 2*np.pi*cos_matrix[i]/FS`
Poté jsem pomocí ní vypočítala nulové body `n_matrix.append(np.e**(omega_filter*1j))`.

Sdružené body na opačné straně osy y k `n_matrix` jsou `n_friend.append(np.conj(n_matrix[i]))`.
Nakonec jsem převedla nuly na koeficienty filtru pomocí `np.poly`.

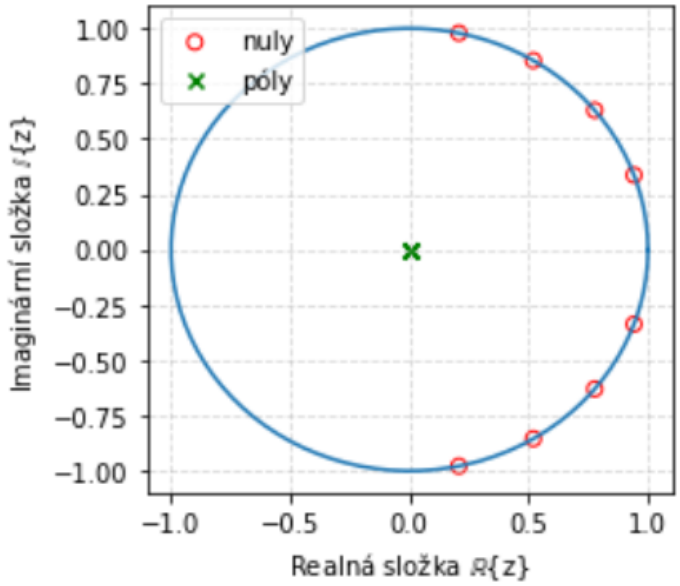
Dostala jsem FIR filtr s 9 koeficienty:
$[1; -4,88979979; 12,34969765; -20,39927867; 23,95828711; -20,39927867; 12,34969765; -4,88979979; 1]$

Na grafu impulzní odezvy lze vidět 9 získaných hodnot:



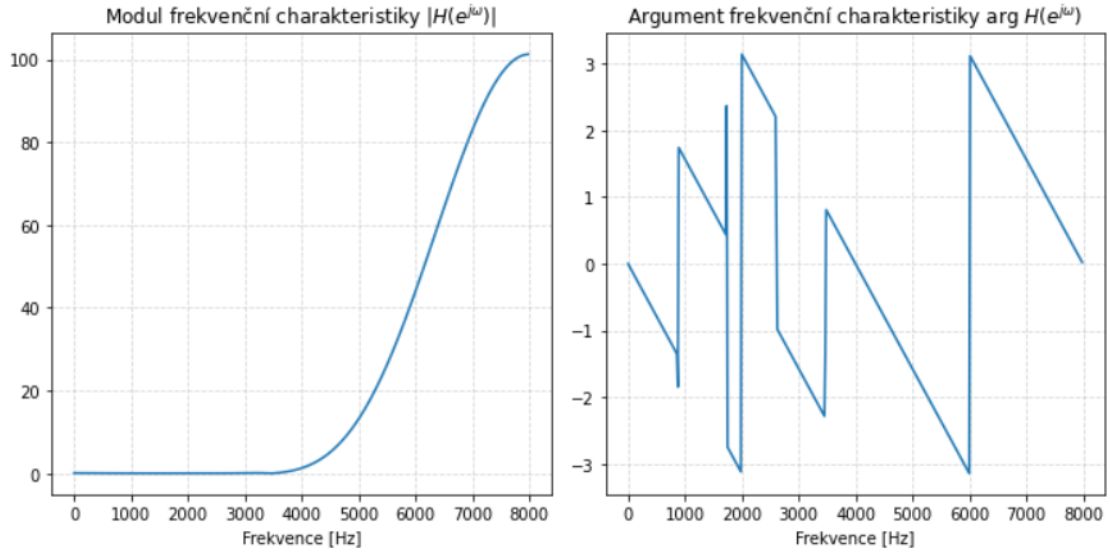
8 Nulové body a póly

Zde jsem znovu také využila již vytvořeného kódu z materiálů od Žmolíkové a za proměnnou b jsem dosadila svoje vyfiltrované koeficienty `b=K_filter`. Pouze jsem vyměnila svou proměnnou normované frekvence `fs` za `FS`, protože jsem ji tak pojmenovala už v minulých úlohách.



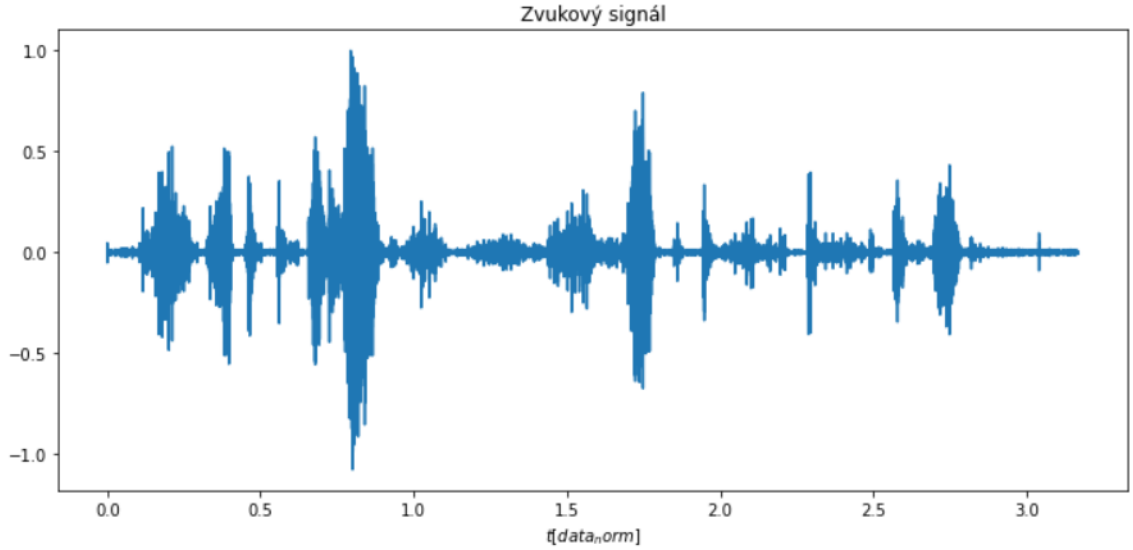
9 Frekvenční charakteristika

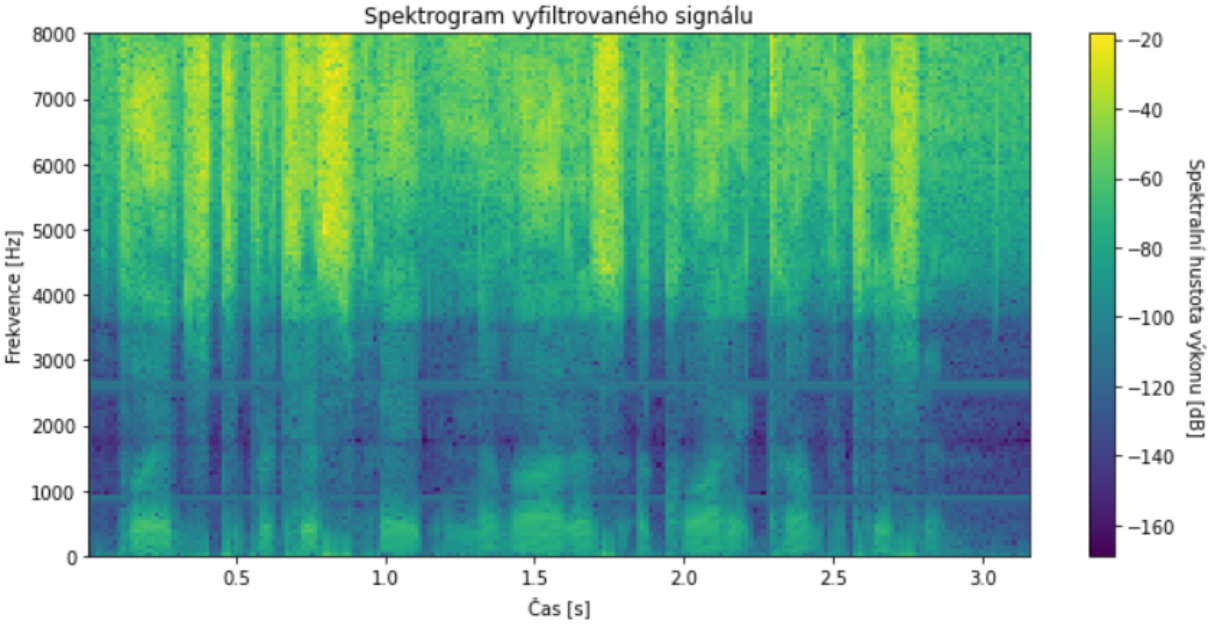
Zjistila jsem, že filtr potlačuje rušivý signál, což je vidět i z grafu, podle toho, jak ten signál v tom grafu "skáče".



10 Filtrace

Provedla jsem ostatní kroky a zkontrolovala, zda je výsledný signál v rozmezí $(-1; 1)$, což nebyl a tak jsem ho znovu znormovala pomocí dělením maximem absolutní hodnoty.




Spektrogram vyfiltrovaného signálu
Frekvence [Hz]
8000
7000
6000
5000
4000
3000
2000
1000
0
0.5
1.0
1.5
2.0
2.5
3.0
Čas [s]
−20
−40
−60
−80
−100
−120
−140
−160
Spektrální hustota výkonu [dB]

11 Použité materiály

11.1 Odkazy:

Obecné Python tipy - Žmolíková

Zvuk spektra filtrace - Žmolíková

Rovnice: https://pythonnumericalmethods.berkeley.edu/notebooks/chapter24.02-Discrete-Fourier-Transform.html