Vysoké učení technické v Brně
Fakulta informačních technologií

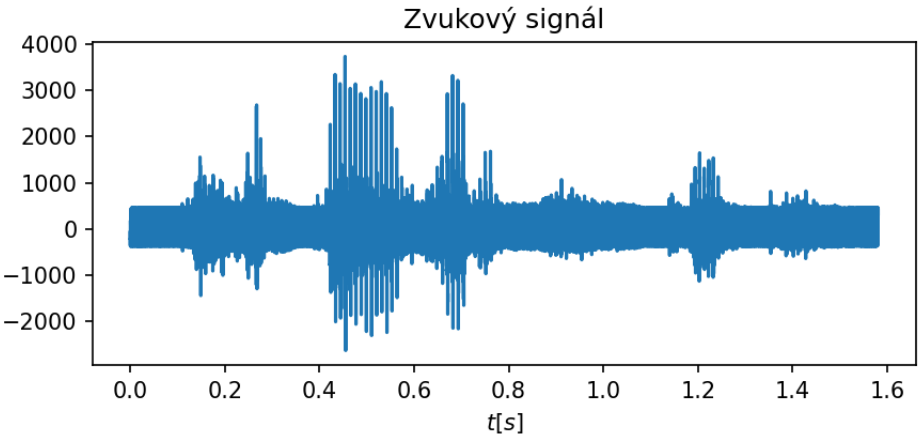
Signály a systémy

Filtrování signálu

29. prosince 2021 David Kocman, xkocma08

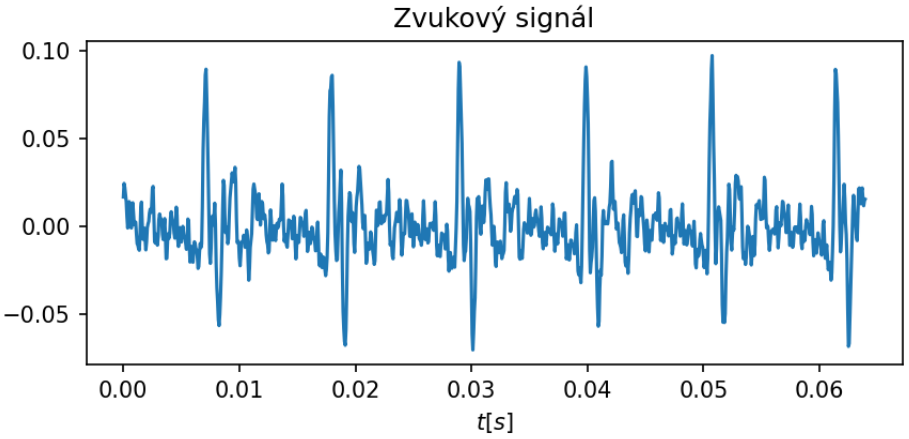
1 Základy

Pro načtení signálu jsem použil z knihovny `scipy.io` funkci `wavfile.read` a pro délku signálu z knihovny `librosa` funkci `librosa.get_duration`. Délka signálu je 1,5808125 s. Ve vzorcích je to 25 293 vzorků. Maximální hodnota je 3 733 a minimální je -2 629.



2 Předzpracování a rámce

Po normalizaci maximální absolutní hodnotou 2^{15} jsem signál ustřednil střední hodnotou a jednoduchým cyklem rozdělil na rámce, které jsem si nechal zobrazit, a z výsledných 48 jsem si vybral 16. rámec, zobrazen níže:



3 DFT

K implementaci vlastní DFT jsem zvolil cyklický průchod.

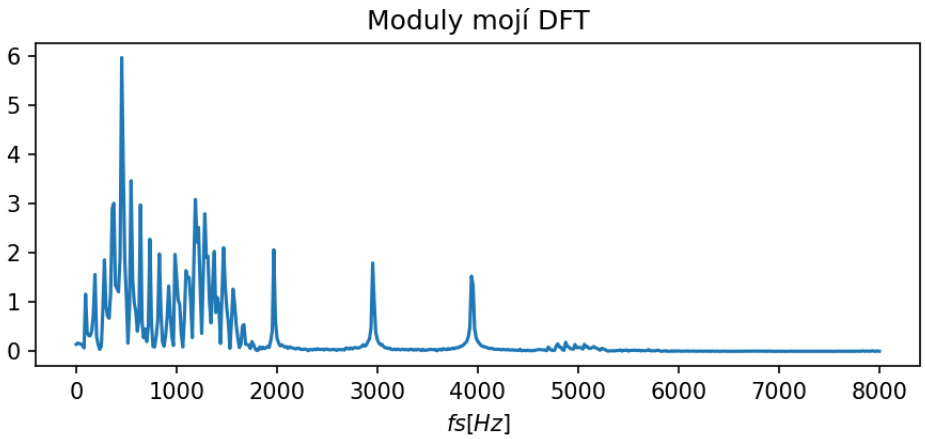
```
for z in my_fft[0]:  # vypocet DTF
    h = 0
    for y in Matrix[15]:
        my_fft[0][g] = my_fft[0][g] + (Matrix[15][h] * (math.e ** (-(2j*math.pi/N)*h*g)))
        h = h + 1
    g = g + 1
```

Obrázek 1: Zanořené `for` cykly pro DFT

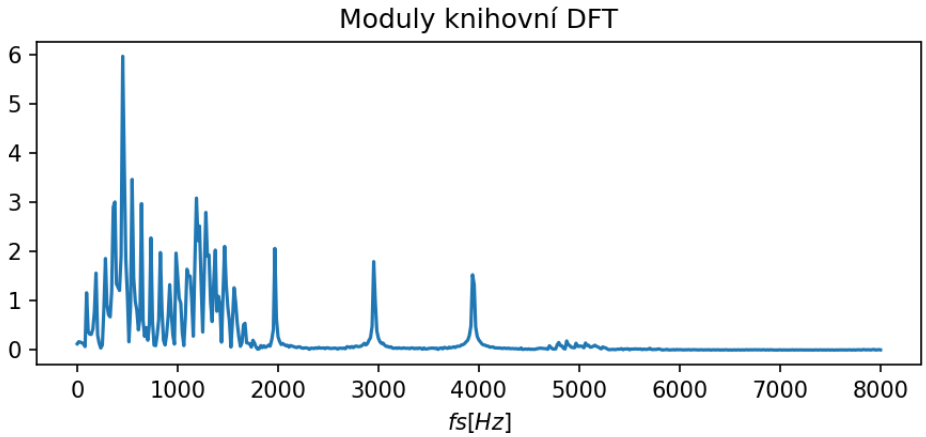
Do matice `my_fft` se postupně přidávají prvky upravené podle vzorce pro DFT:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-\frac{j2\pi}{N}kn}$$

Dále jsem výsledné koeficienty zobrazil na polovině vzorkovací frekvence 8000 Hz:



Pro kontrolu jsem si spočítal i knihovní DFT: `np.fft.fft`, kterou jsem zobrazil níže, a také porovnal koeficienty knihovní a mojí funkce funkcí `np.allclose`



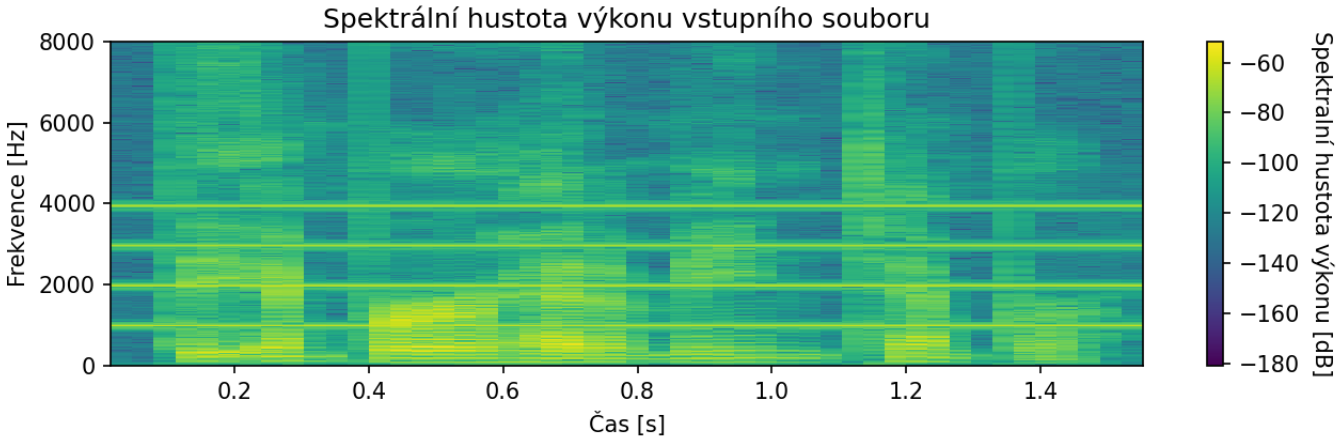
Jak pohledem, tak i díky výstupu funkce `np.allclose`, která vrací `True`, můžeme usoudit že jsem DFT funkci naimpementoval správně.

4 Spektogram

Pro zobrazení spektogramu vstupního signálu jsem zvolil knihovní funkci `scipy.signal.spectogram`. Jako arguemnty jsem zadal ustředněné a normalizované vstupní data, vzorkovací frekvenci 16 000 Hz a délku okna 1024 vzorků s překrytím 512 vzorků. Před znázorněním jsme koeficienty DFT museli zlogaritmovat rovnicí:

$$P[k] = 10 \log_{10} |X[k]|^2$$

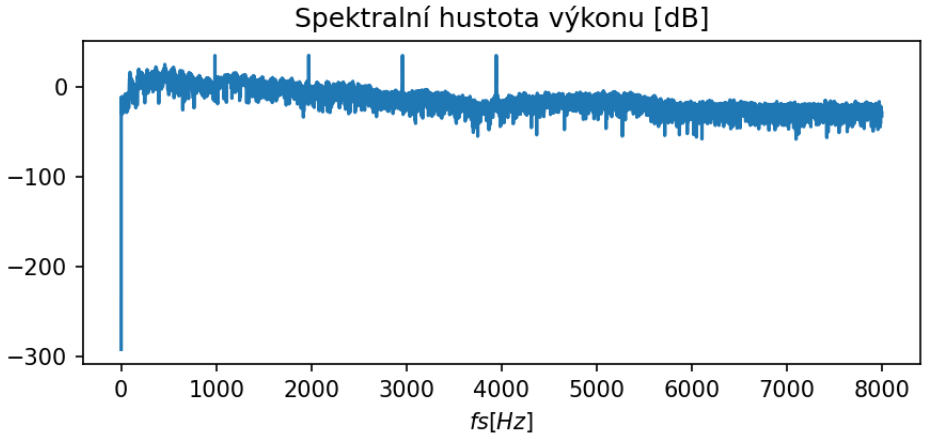
Spektogram je znázorněn níže:



Na obrázku můžeme krásně vidět 4 rovnoběžné linie, které znázorňujou rušivé komponenty. Jejich frekvence zjistíme v následujícím úkolu.

5 Určení rušivých frekvencí

Pro přesné určení rušivých frekvencí jsem se rozhodl používat jedno spektrum. Postup je stejný jako u spektogramu. Spočítal jsem si DFT, zlogaritmoval a zobrazil:



Při vyčtení výběžků jsem zjistil, že frekvence rušivých prvků jsou následující:

$$f_1 = 986Hz$$

$$f_2 = 1972Hz$$

$$f_3 = 2958Hz$$

$$f_4 = 3944Hz$$

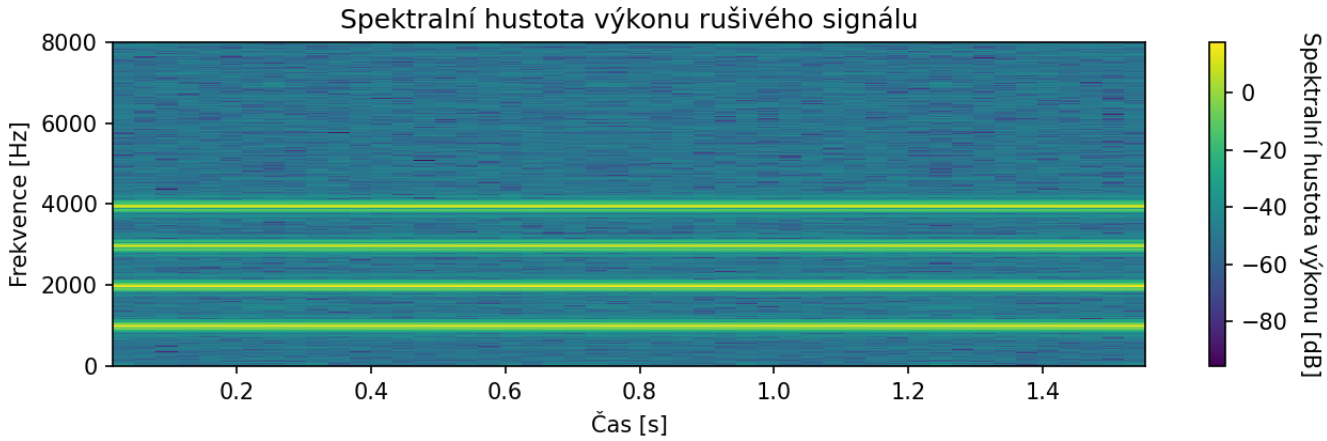
Také jde poznat, že frekvence jsou harmonicky vztažené. f_2, f_3 a f_4 je dvoj, troj a čtyřnásobek f_1.

6 Generace signálu

Amplidudy výsledných cosinusovek jsem spočítal podle vzorečku pro cosinus:

$$x(k) = \cos(2\pi fk)$$

Nadále jsem všechny amplitudy sečetl a vytvořil nový soubor `4cos.wav`. Níže je zobrazen jeho spektogram s velikostí okna 1024 vzorků a překrytím 512 vzorků:



Poslechem i porovnáním spektogramů jsem usoudil, že jsem frekvence určil správně.

7 Čisticí filtr

Na návrh filtru jsem zvolil třetí variantu: **návrh 4 pásmových zádrží**. využil jsem funkcí `scipy.signal.buttord` [2] a `scipy.signal.butter` [1]. Délku závěrného pásma jsem zvolil 30 Hz a šíři přechodu do propustného pásma 50 Hz na každé straně. Hodnoty jsem musel normovat Nyquistovou frekvencí.

```
N1, Wn1 = signal.buttord([0.117, 0.1295], [0.121375, 0.125125], 3, 40, False)
b1, a1 = signal.butter(N1, Wn1, 'bandstop', False)
```

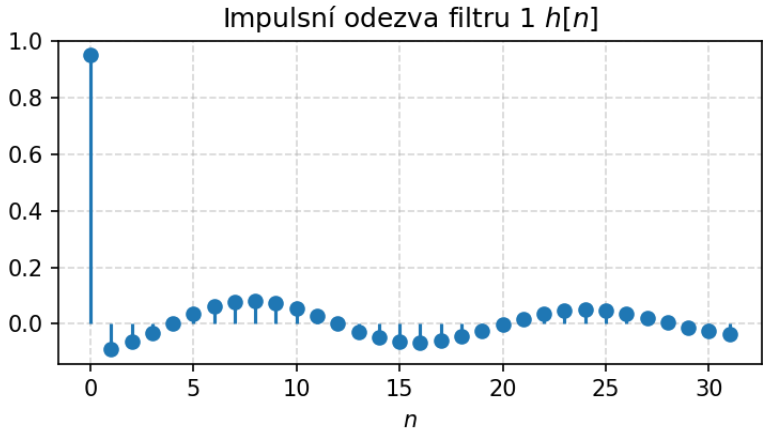
Obrázek 2: Návrh pásmové propusti na frekvenci 986 Hz

Všechny impulzní odezvy jsou omezeny na 32 prvků.

Koeficienty a impulzní odezva filtru na frekvenci 986 Hz:

$$a = [1 \quad -7.31494736 \quad 23.96633038 \quad -45.86339438 \quad 56.02595203 \quad -44.72646787 \quad 22.79284042 \quad -6.78432819 \quad 0.90447155]$$

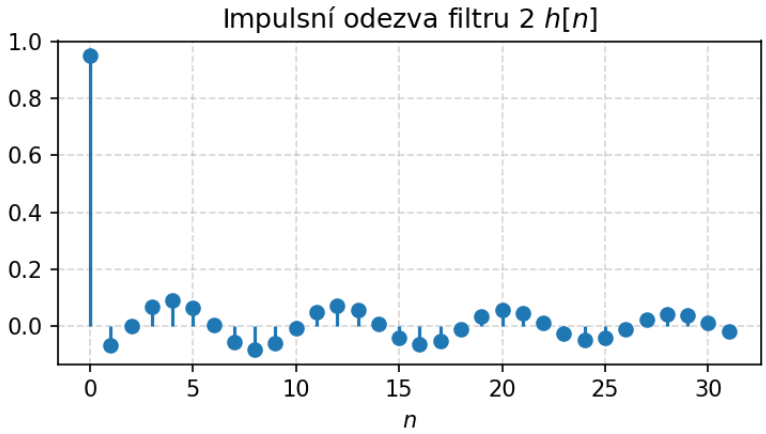
$$b = [0.95103709 \quad -7.0451983923.37547509 \quad -45.29937052 \quad 56.03657001 \quad -45.29937052 \quad 23.37547509 \quad -7.04519839 \quad 0.95103709]$$



Koeficienty a impulzní odezva filtru na frekvenci 1972 Hz:

$$a = [1 \quad -5.6461884 \quad 15.85380847 \quad -27.76165743 \quad 32.97986866 \quad -27.06513539 \quad 15.06826209 \quad -5.23178087 \quad 0.90335749]$$

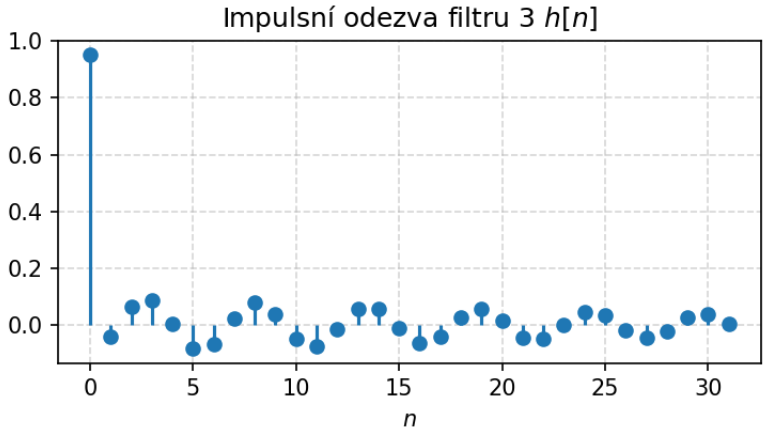
$$b = [0.95045121 \quad -5.43547497 \quad 15.45852692 \quad -27.41690608 \quad 32.98734045 \quad -27.41690608 \quad 15.45852692 \quad -5.43547497 \quad 0.95045121]$$



Koeficienty a impulzní odezva filtru na frekvenci 2958 Hz:

$$a = [1 \quad -3.14240286 \quad 7.60106073 \quad -11.12717692 \quad 13.29861115 \quad -10.8465889 \quad 7.22255135 \quad -2.91061587 \quad 0.90288298]$$

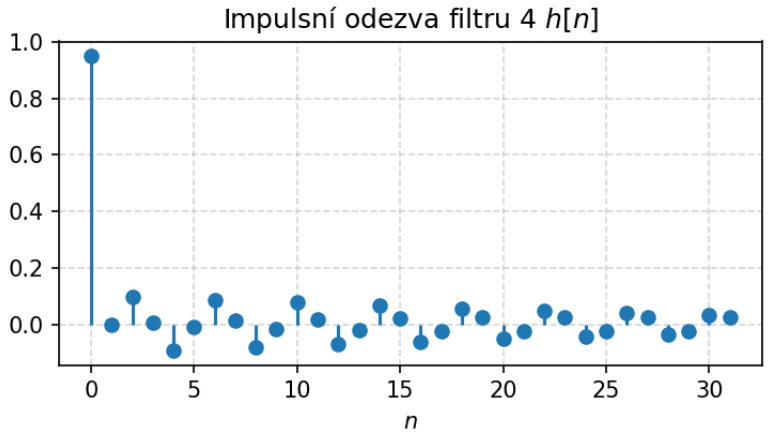
$$b = [0.95020154 \quad -3.02453619 \quad 7.41102151 \quad -10.98885608 \quad 13.30266011 \quad -10.98885608 \quad 7.41102151 \quad -3.02453619 \quad 0.95020154]$$



Koeficienty a impulzní odezva filtru na frekvenci 3944 Hz:

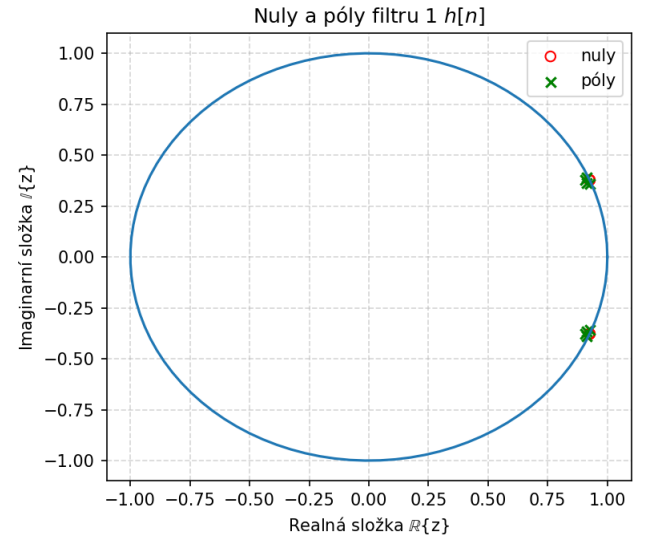
$$a = [1 \quad -0.1736804 \quad 3.90878936 \quad -0.50807035 \quad 5.71970698 \quad -0.49521297 \quad 3.71346895 \quad -0.16082471 \quad 0.90254748]$$

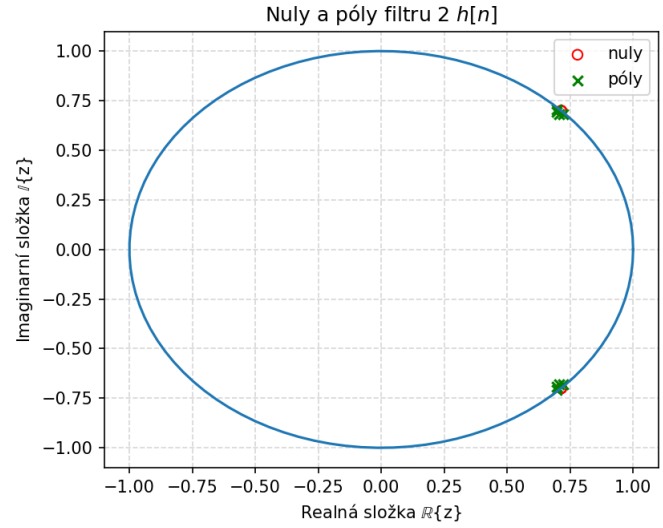
$$b = [0.95002499 \quad -0.16714271 \quad 3.8111273 \quad -0.50175149 \quad 5.72220819 \quad -0.50175149 \quad 3.8111273 \quad -0.16714271 \quad 0.95002499]$$

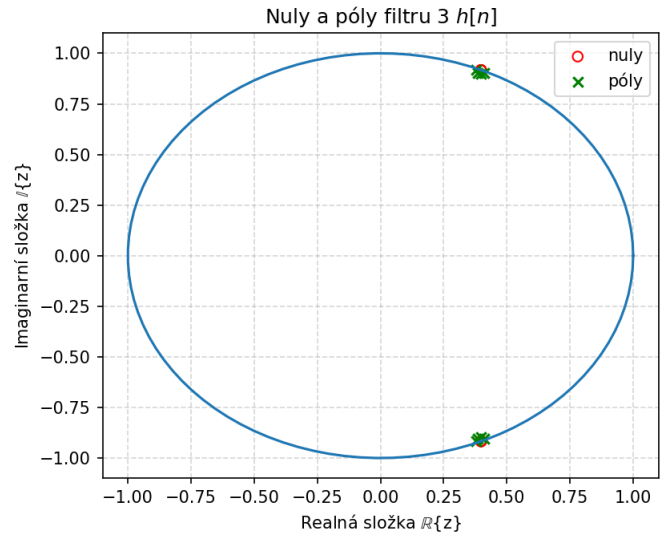


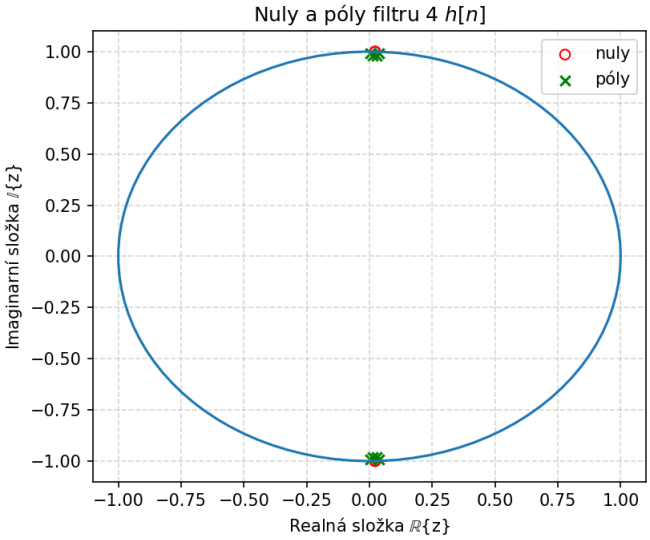
8 Nulové body a póly

Pro výpočet nulových bodů a pólů jsem použil funkci `scipy.signal.tf2zpk`. Můžeme vidět, že filtry jsou stabilní, jelikož všechny póly se nacházejí uvnitř kružnice.



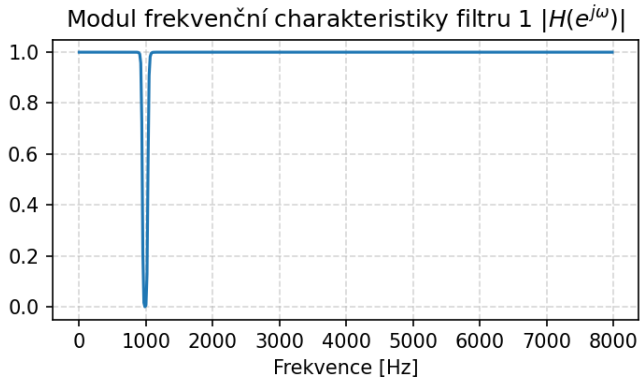


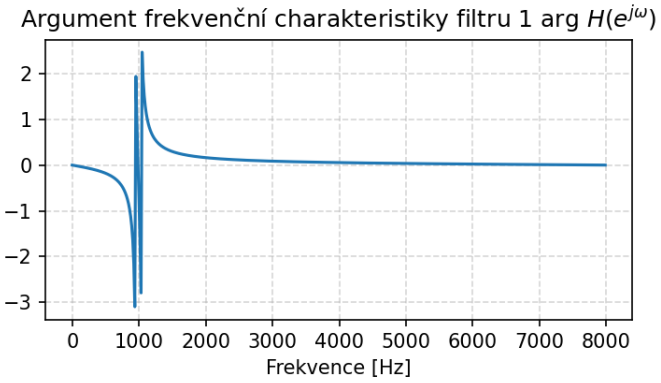


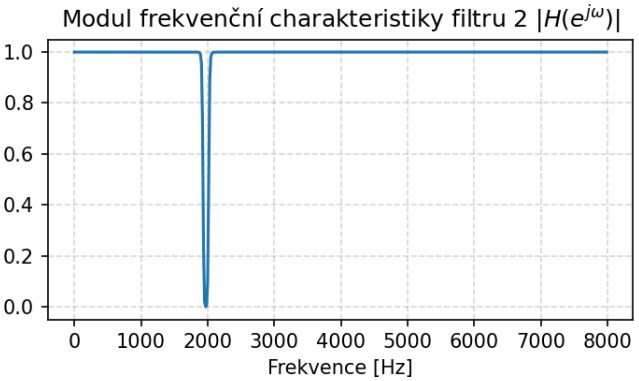


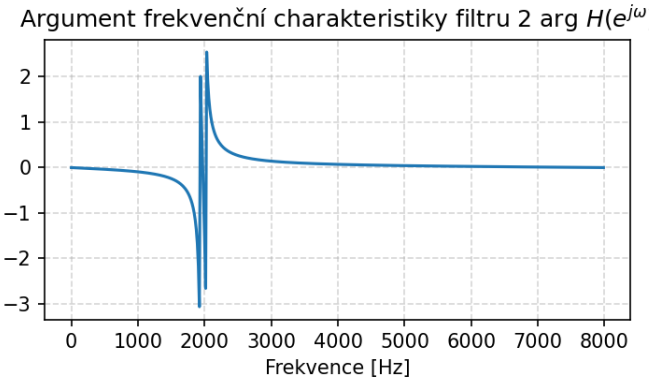
9 Frekvenční charakteristika

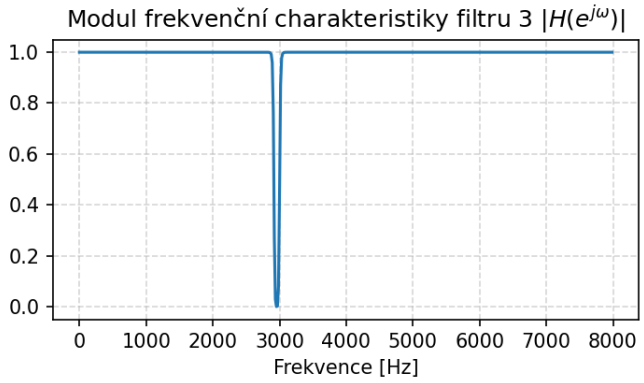
Na vypočítání frekvenční charakteristiky filtrů jsem použil funkci `scipy.signal.freqz`. Níže jsou moduly a argumenty všech filtrů:

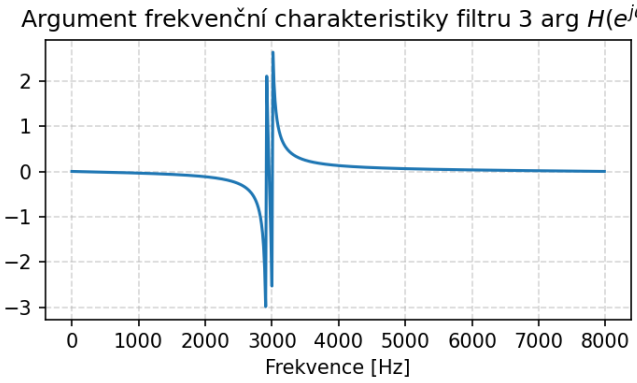


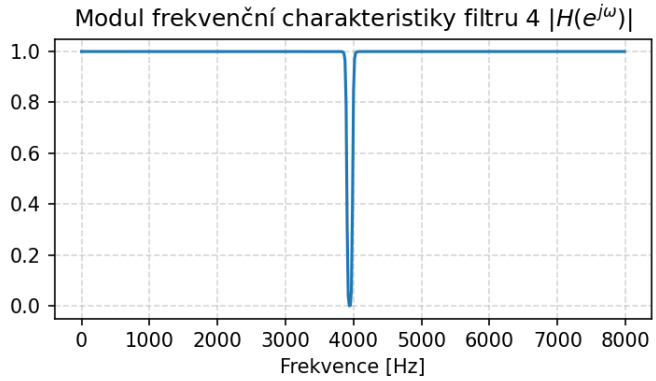


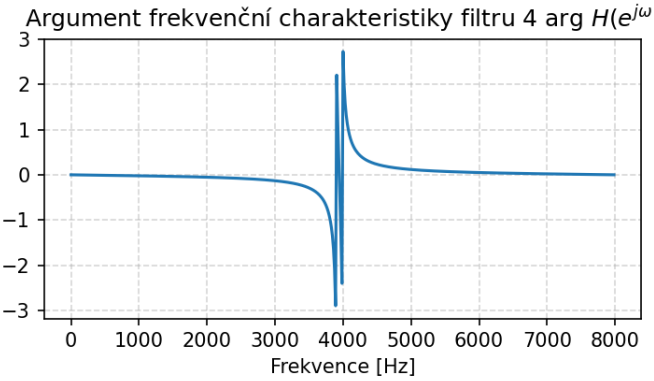








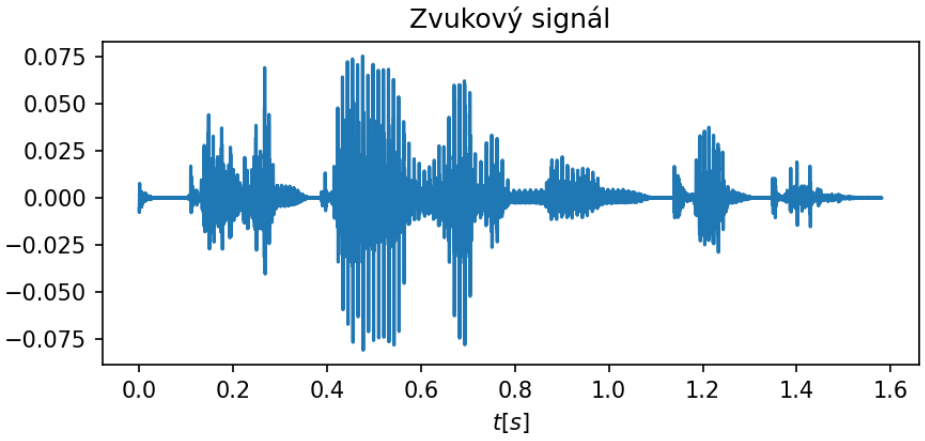




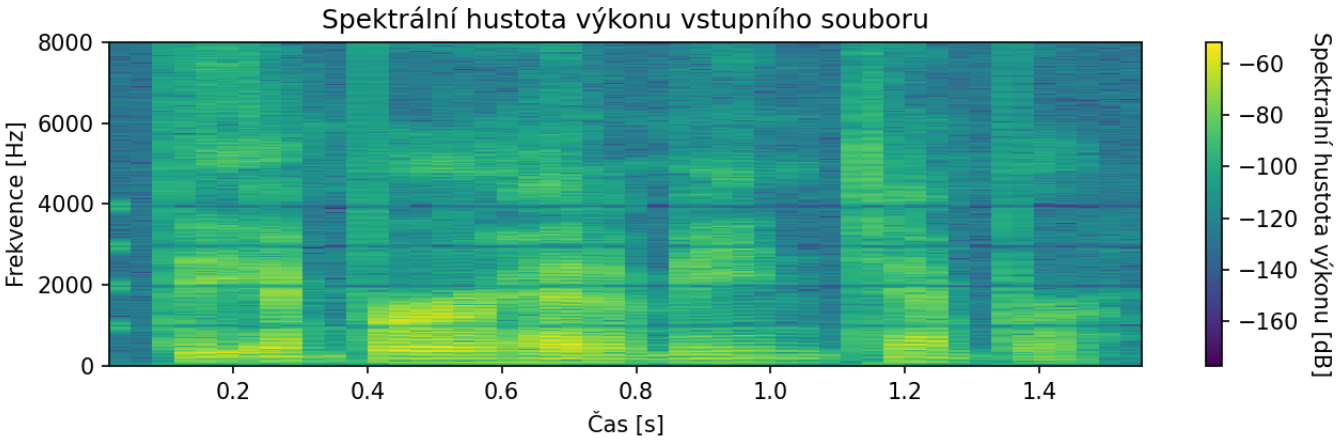
Můžeme vidět, že filtry potlačují rušivý signál na správných frekvencích.

10 Filtrace

Výsledný vyfiltrovaný signál jsem dostal tak, že ustředněná a normovaná vstupní data jsem prohnal přes navržené filtry a uložil do souboru `clean_bandstop.wav`. Poslechem jsem se přesvědčil, že signál jsem úspěšně vyčistil. Pro jistotu jsem si to ale také zobrazil:



Zde si můžeme všimnout absence šumu



Obrázek 3: Spektogram vyfiltrovaného signálu

Ve spektogramu jde poznat, že jsme 4 rovnoběžné čáry úspěšně vyfiltorvali.

citace

[1] docs.scipy: *scipy.signal.butter reference*. [online], [viděno 6.12.2021].
URL <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html>

[2] docs.scipy: *scipy.signal.buttord reference*. [online], [viděno 6.12.2021].
URL <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.buttord.html>