



Protokol k projektu do předmětu ISS

Ivan Tsiareshkin (xtsiar00)

07. 01. 2022

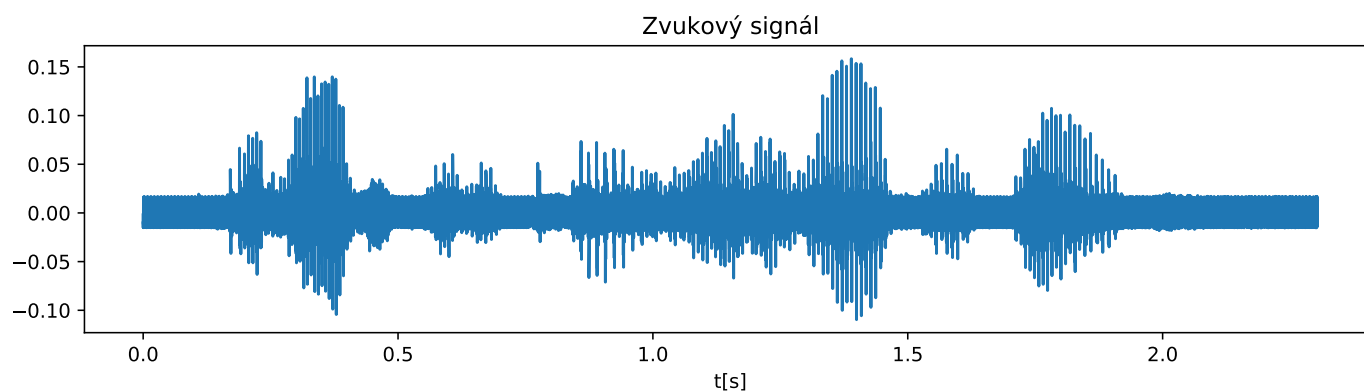
Úvod

Projekt jsem vypracoval v jazyce Python 3.6.9 pomocí knihoven `numpy`, `matplotlib`, `scipy` a `soundfile`. Funkčnost projektu implementována do jednoho hlavního souboru `proj.py`.

1. Základy

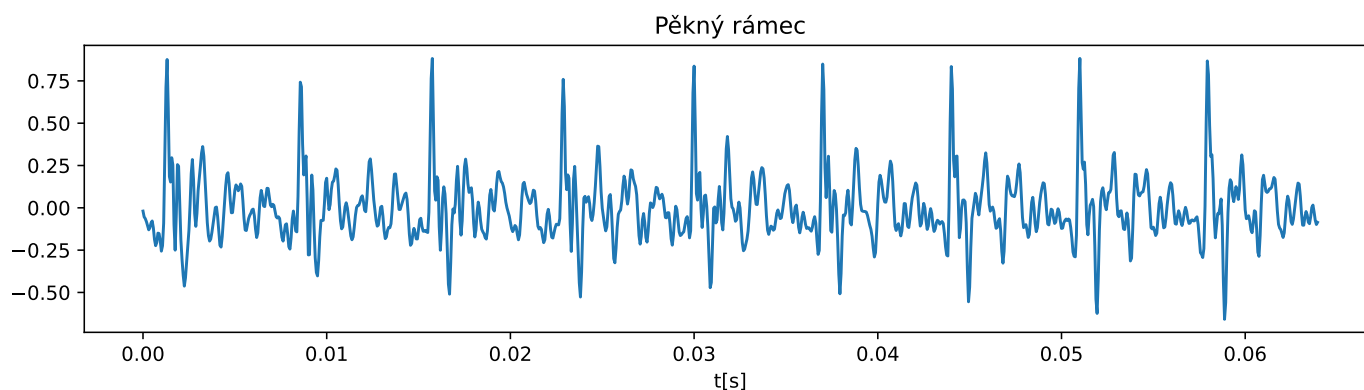
Signál	Délka ve vzorcích	Délka v sekundách
xtsiar00.wav	36864	2.304

Minimální hodnota	Maximální hodnota
-0.109588623046875	0.158355712890625



2. Předzpracování a rámce

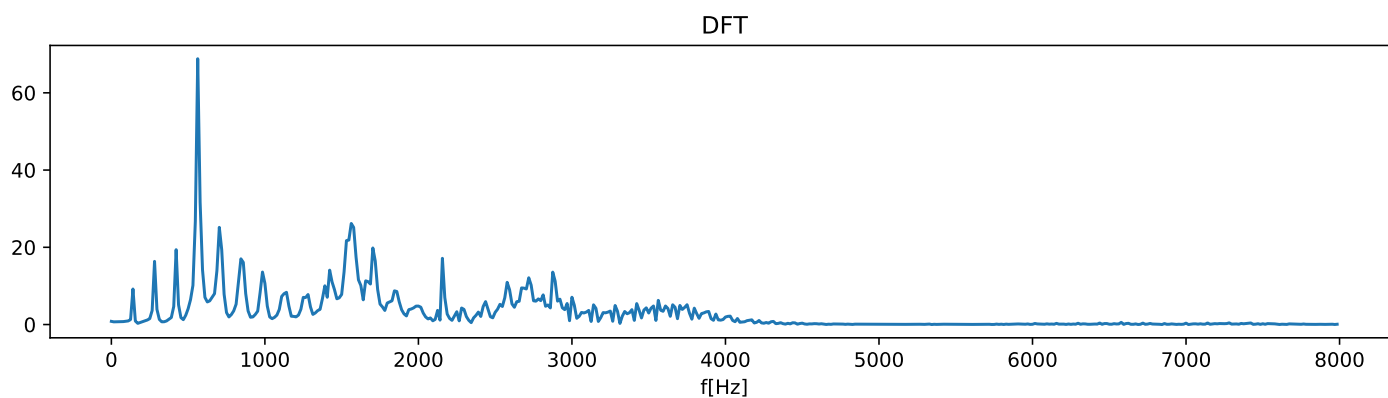
Rozdělení na rámce jsem provedl ve `for` loopu, který prochází po celé délce signálu ve vzorcích. První rámec bude v rozsahu $(0; 1024)$, druhý $(512; 1536)$, třetí $(1024; 2048)$ atd., takže jsem použil tuto konstrukci k výpočtu rozsahu jednoho rámce: `[j*512:1024+j*512]`. Pěkným a znělým rámcem byl rámec č. 10



3. DFT

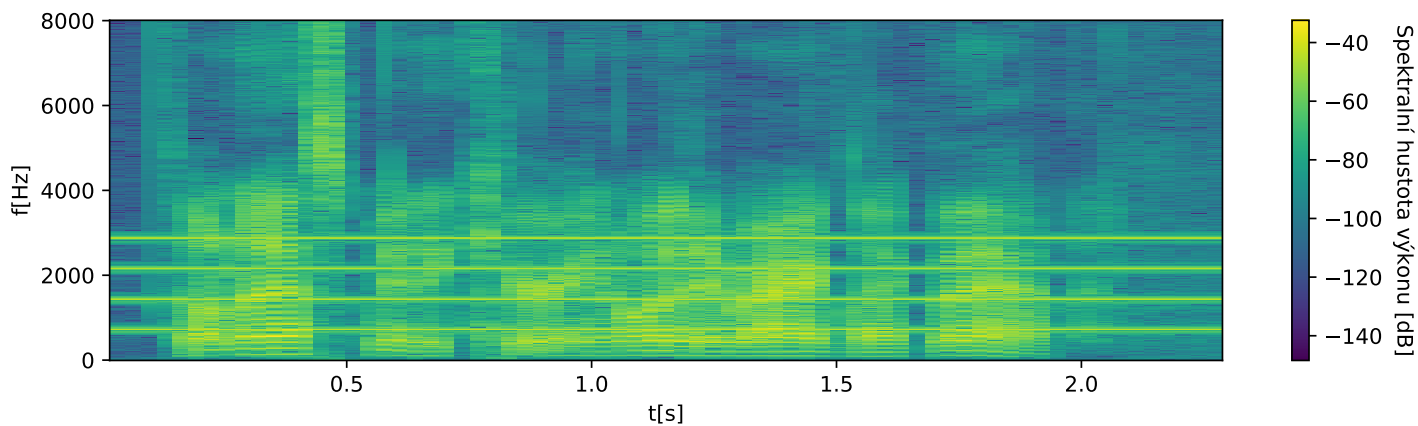
Moje implementace DFT, která vypočítá stejný výsledek jako `np.fft.fft`, vyzerá takto:

```
def DFT(x):
    x = np.asarray(x, dtype=float)
    N = x.shape[0]
    n = np.arange(N)
    f = n.reshape((N, 1))
    M = np.exp(-2j * np.pi * f * n / N)
    return np.dot(M, x)
```



4. Spektrogram

Při tvorbě spektrogramu jsem inspiroval Jupyter Notebookem Katky Žmolíkové a použil jsem funkci `spectrogram`.



5. Určení rušivých frekvencí

Rušivé frekvence jsem odečetl ručně ze spektrogramu a dostal jsem následující frekvence: 720 Hz, 1440 Hz, 2160 Hz a 2880 Hz.

6. Generování signálu

Pomocí vzorku jsem vygeneroval 4 kosinusy a ty pak postupně sečetl do jednoho signálu.

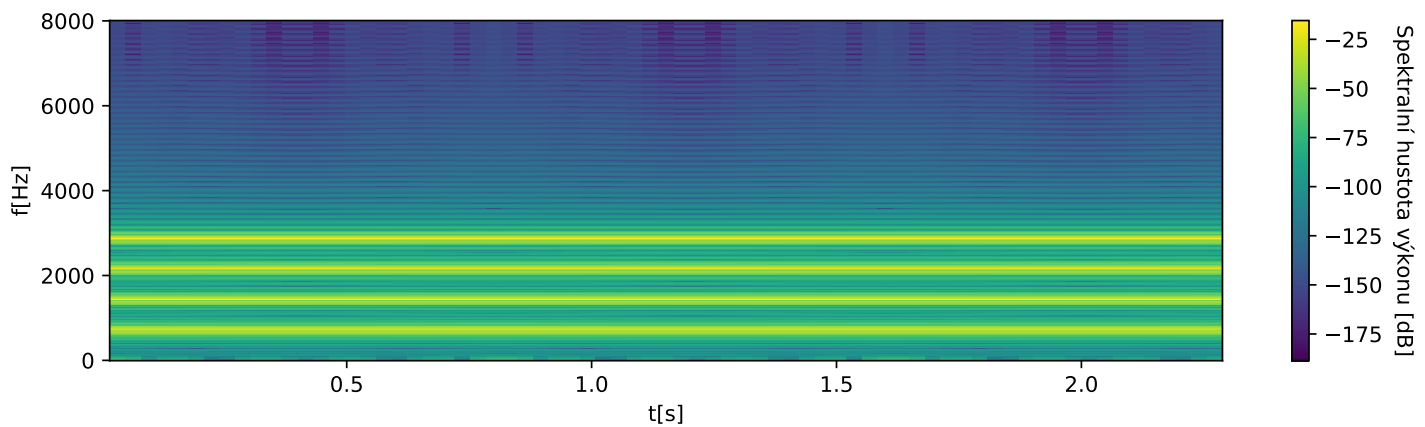
```
arr = []
t = np.arange(len(fs)) / data

for i in range(len(fs)):
    arr.append(i / data)

out1 = np.cos(2 * np.pi * fq0 * np.array(arr))
out2 = np.cos(2 * np.pi * fq1 * np.array(arr))
out3 = np.cos(2 * np.pi * fq2 * np.array(arr))
out4 = np.cos(2 * np.pi * fq3 * np.array(arr))

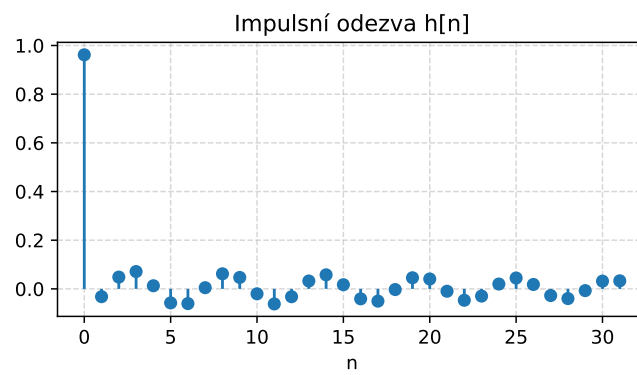
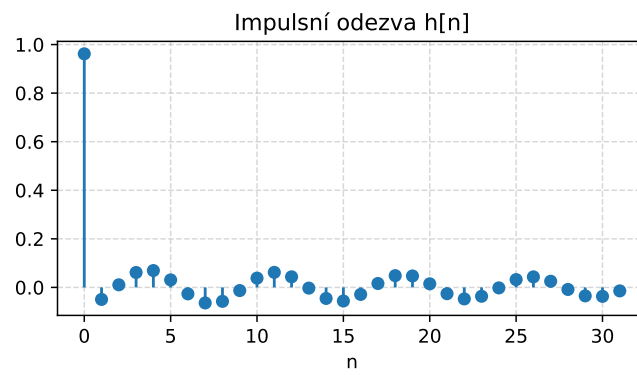
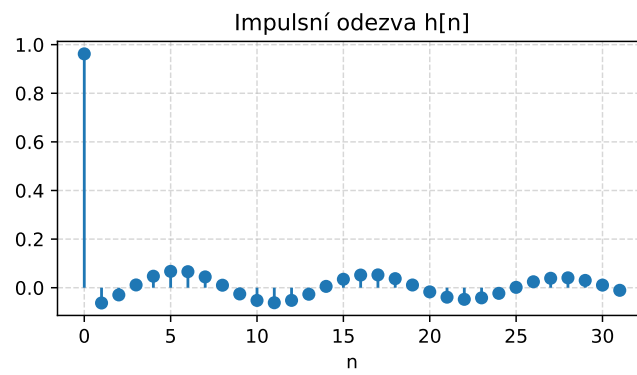
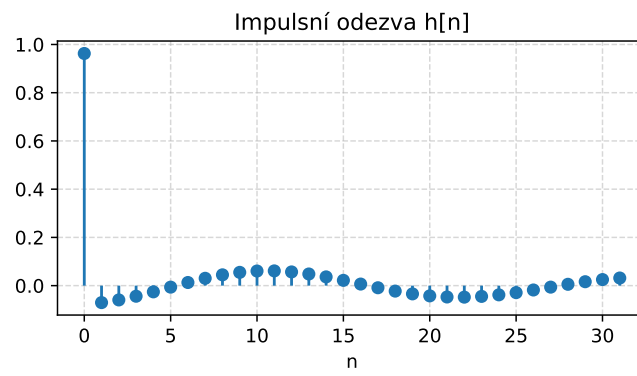
out = out1+out2+out3+out4
```

Dostal jsem takový spektrogram rušivého signálu:



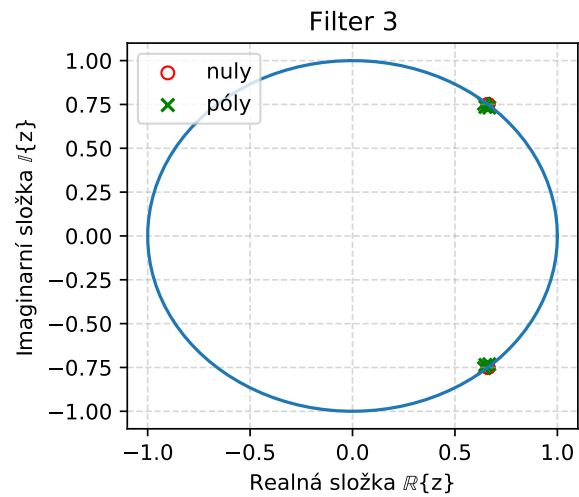
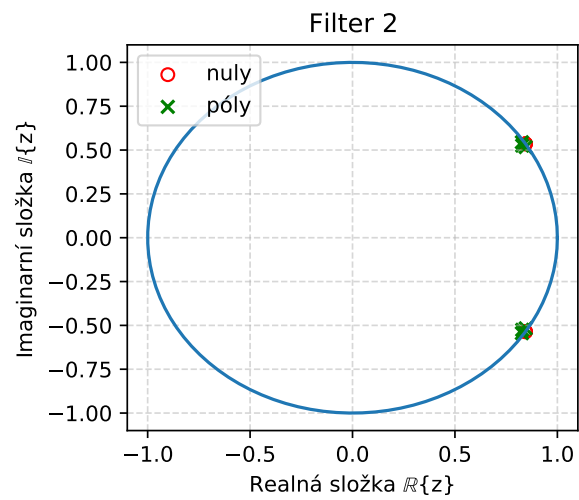
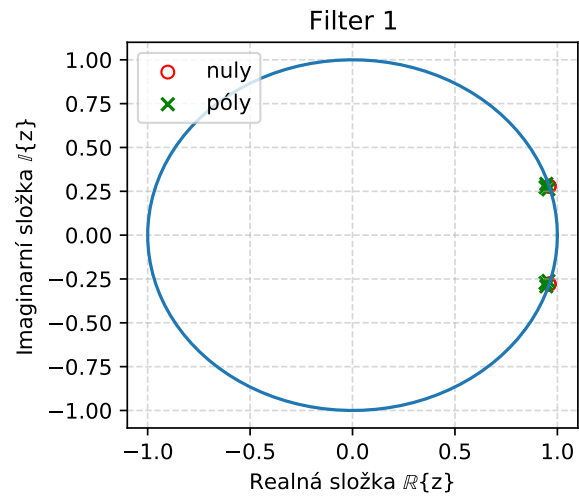
7. Čisticí filtr

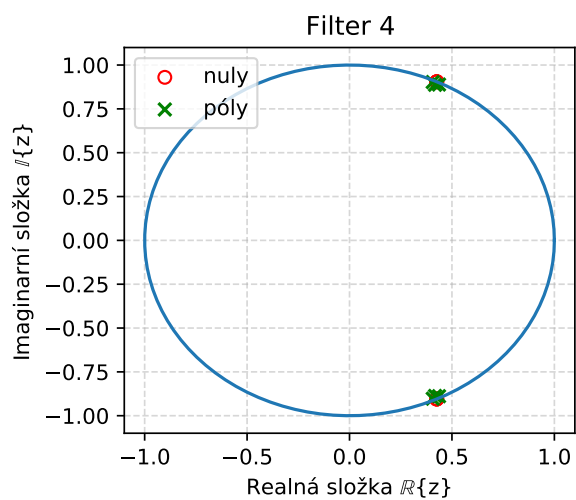
Jako čistící filtr jsem použil 3. variantu, tj. návrh 4 pásmových zádrží. Držel jsem se zadání, ale změnil jsem šířku závěrného pásma na 20 Hz. Impulsní odezvy navrhnutých filtrů:



8. Nulové body a póly

Nulové body a póly návrhnutých filtrů:

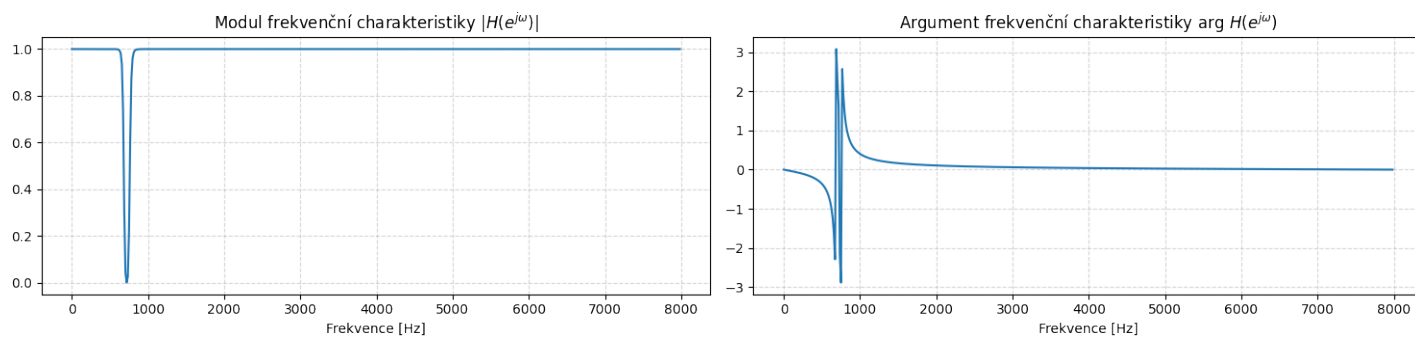




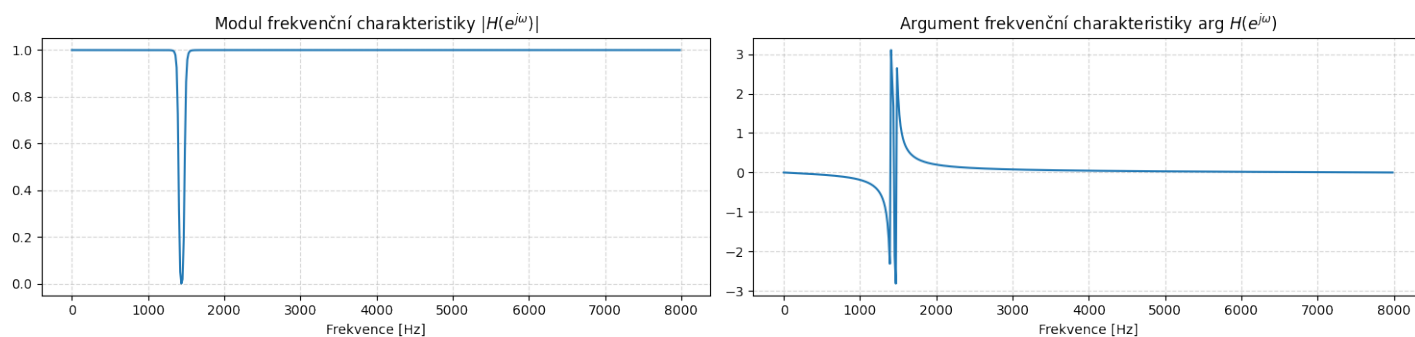
9. Frekvenční charakteristika

Frekvenční charakteristiky návrhnutých filtrů:

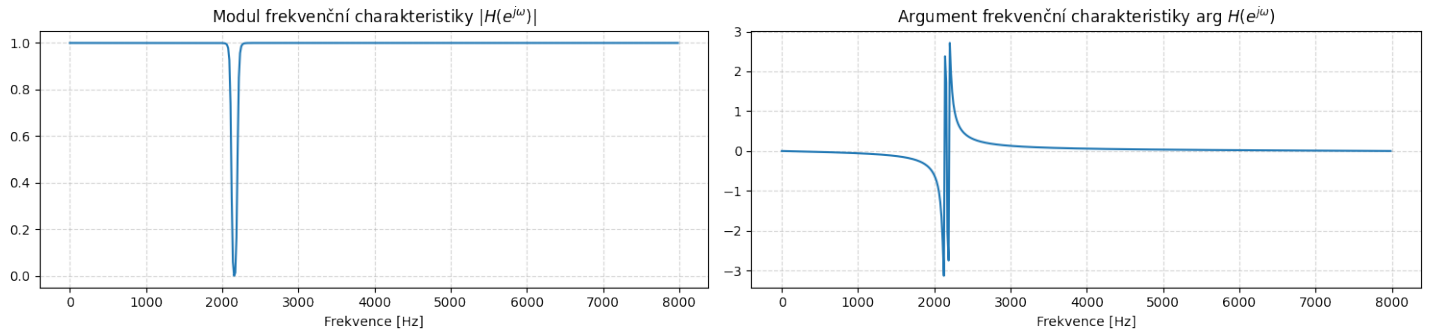
Filter 1



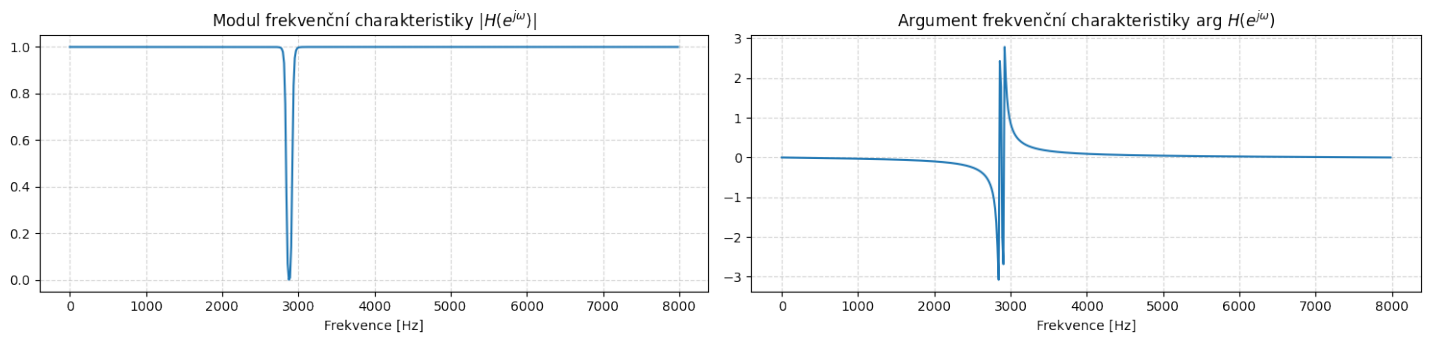
Filter 2



Filter 3



Filter 4



10. Filtrace

Na finální filtraci signálu jsem použil funkci `scipy.signal.filtfilt`, která aplikuje lineární digitální filtr dvakrát, jednou vpřed a jednou vzad. Spektrogram vyfiltrovaného signálu:

