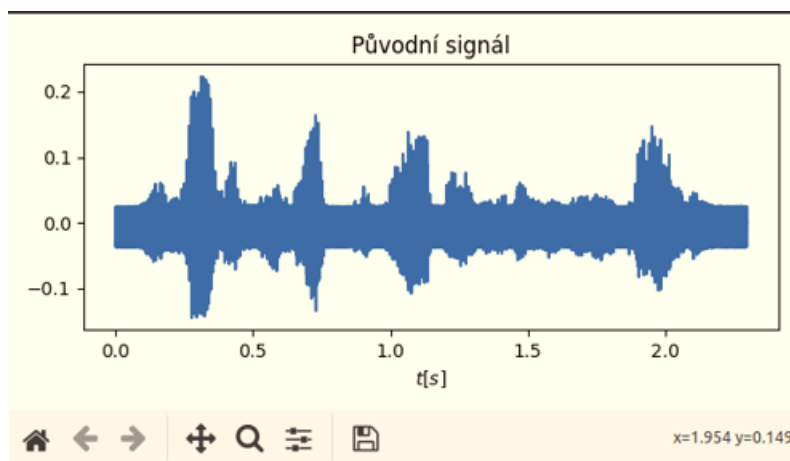


Protokol k projektu ISS

1. Základy

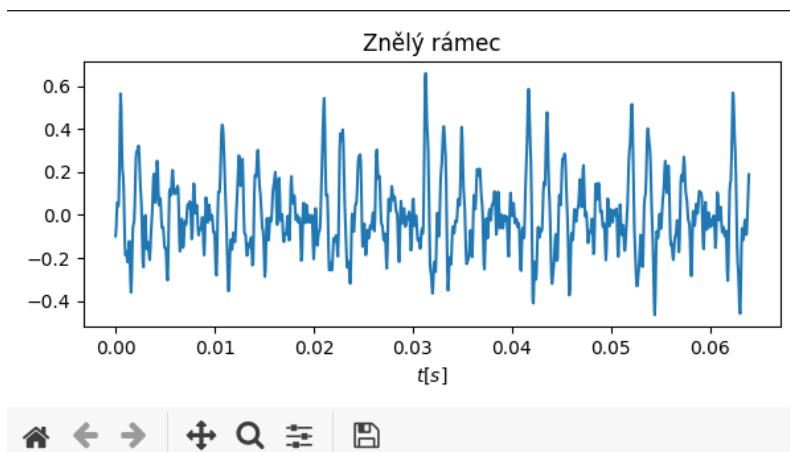
Originální signál jsem načetl pomocí knihovny *soundfile* a její funkce *soundfile.read*. Požadované hodnoty jsem poté zjistil pomocí návratových hodnot této funkce a vypsal je. Nakonec jsem pomocí knihovny *matplotlib* signál zobrazil.

```
PŮVODNÍ SIGNÁL  
minimum je: -0.14569091796875  
maximum je: 0.222991943359375  
delka je 2.297625 sekund nebo 36762 vzorku
```



2. Předzpracování rámce

Střední hodnotu jsem zjistil jednoduše pomocí funkce *np.mean*, rozdělení signálu probíhá přes jednoduché vnořené cykly *for*. Pomocí dalšího cyklu *for* jsem si postupně vykresloval všechny rámce, z nichž jsem vybral ten, který se mi zdál jako nejvíce znělý.

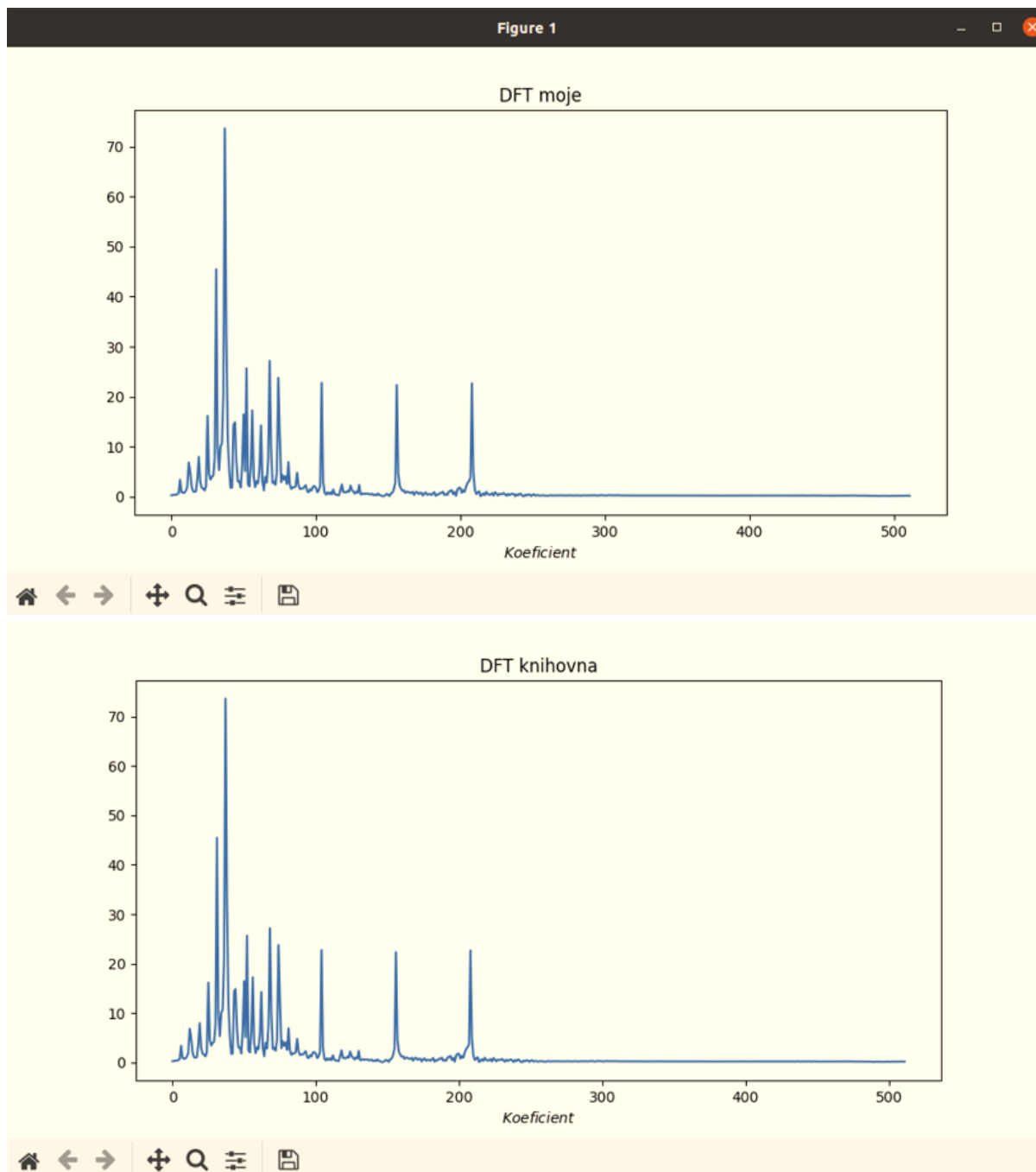


3. DFT

DFT jsem vytvářel podle vzorce

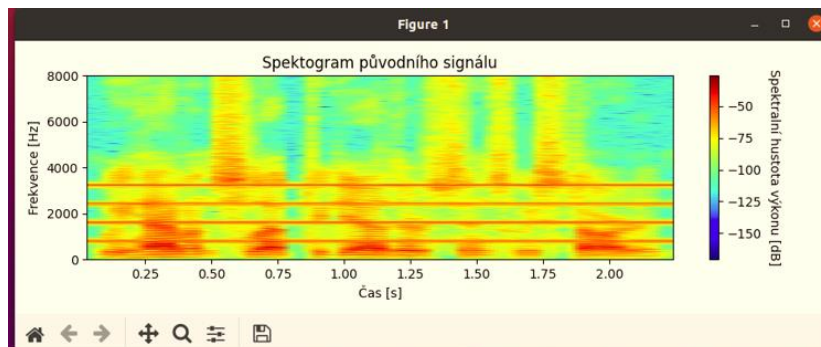
$$x[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}$$

Při implementaci jsem využíval maticových operací z knihovny numpy. Z výsledného DFT jsem smazal druhou polovinu a graf následně zobrazil a porovnal s DFT vytvořenou pomocí `np.fft.fft`



4. Spektrogram

Na spektrogram jsem využil funkci *spectrogram*, kam jsem přímo zadal požadovanou délku okna a překrytí. Následně jsem jej zobrazil opět pomocí knihovny *matplotlib*.



5. Určení rušivých frekvencí

Rušivé frekvence jsem určil jednoduše pomocí přiblížení spektrogramu.

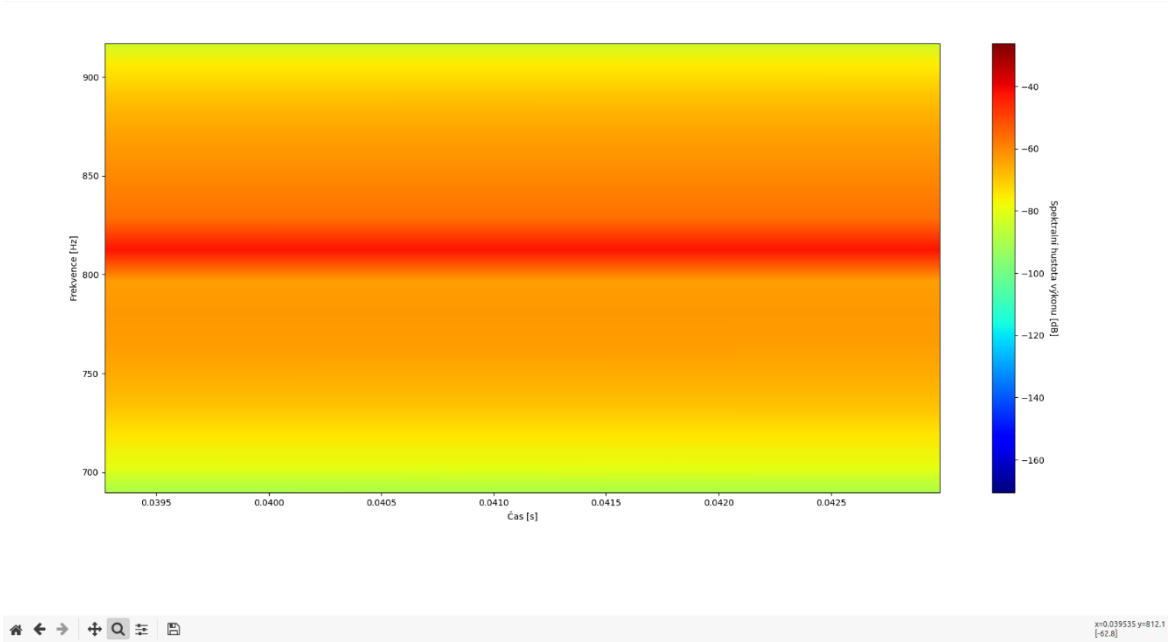
$f_1 = 812 \text{ Hz}$

$f_2 = 1624 \text{ Hz}$

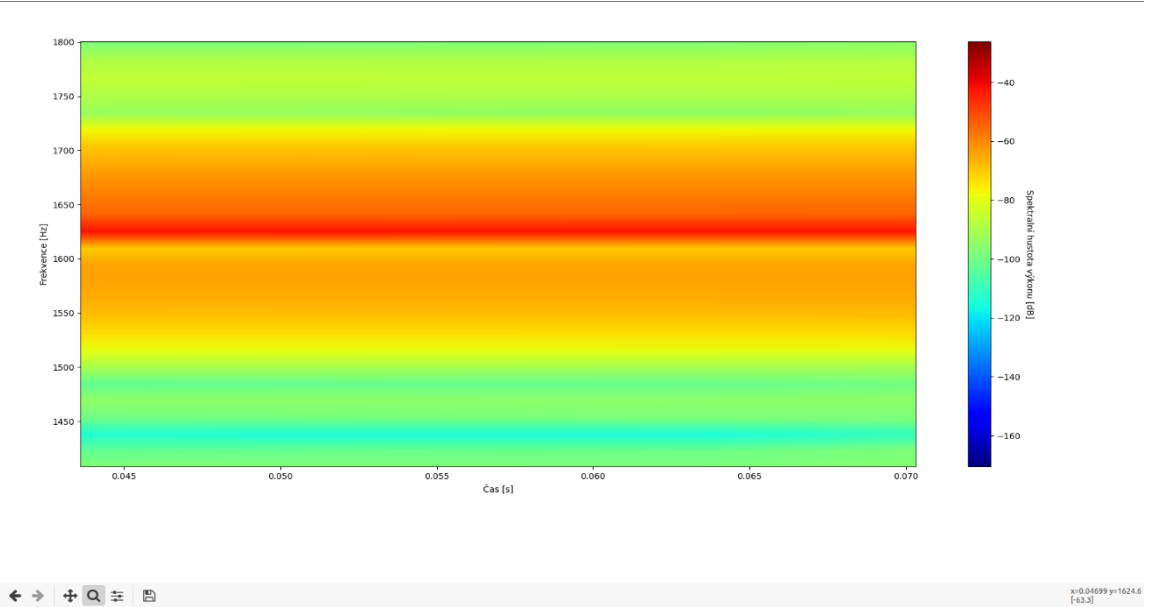
$f_3 = 2436 \text{ Hz}$

$f_4 = 3248 \text{ Hz}$

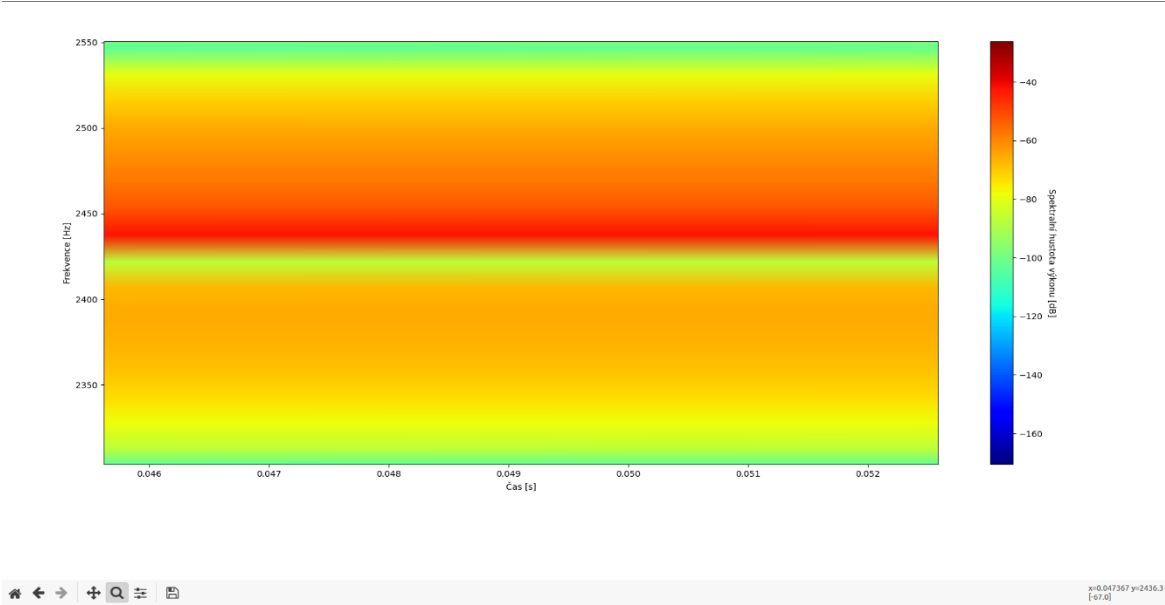
Z vyčtených hodnot vyplývá, že jsou cosinusovky harmonicky vztažené.



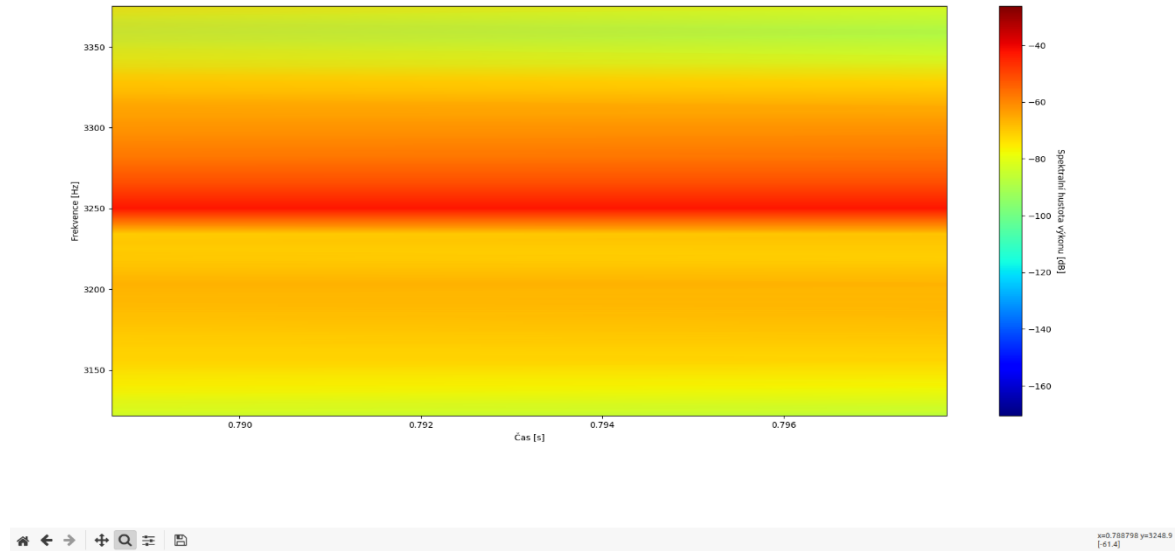
Frekvence 1 – 812 Hz



Frekvence 2 – 1624 Hz



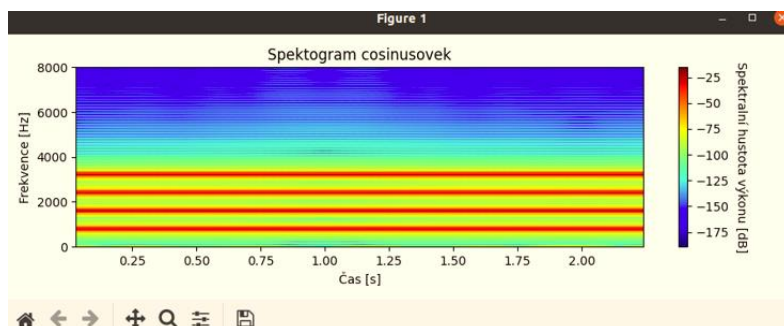
Frekvence 3 – 2436 Hz



Frekvence 4 – 3248 Hz

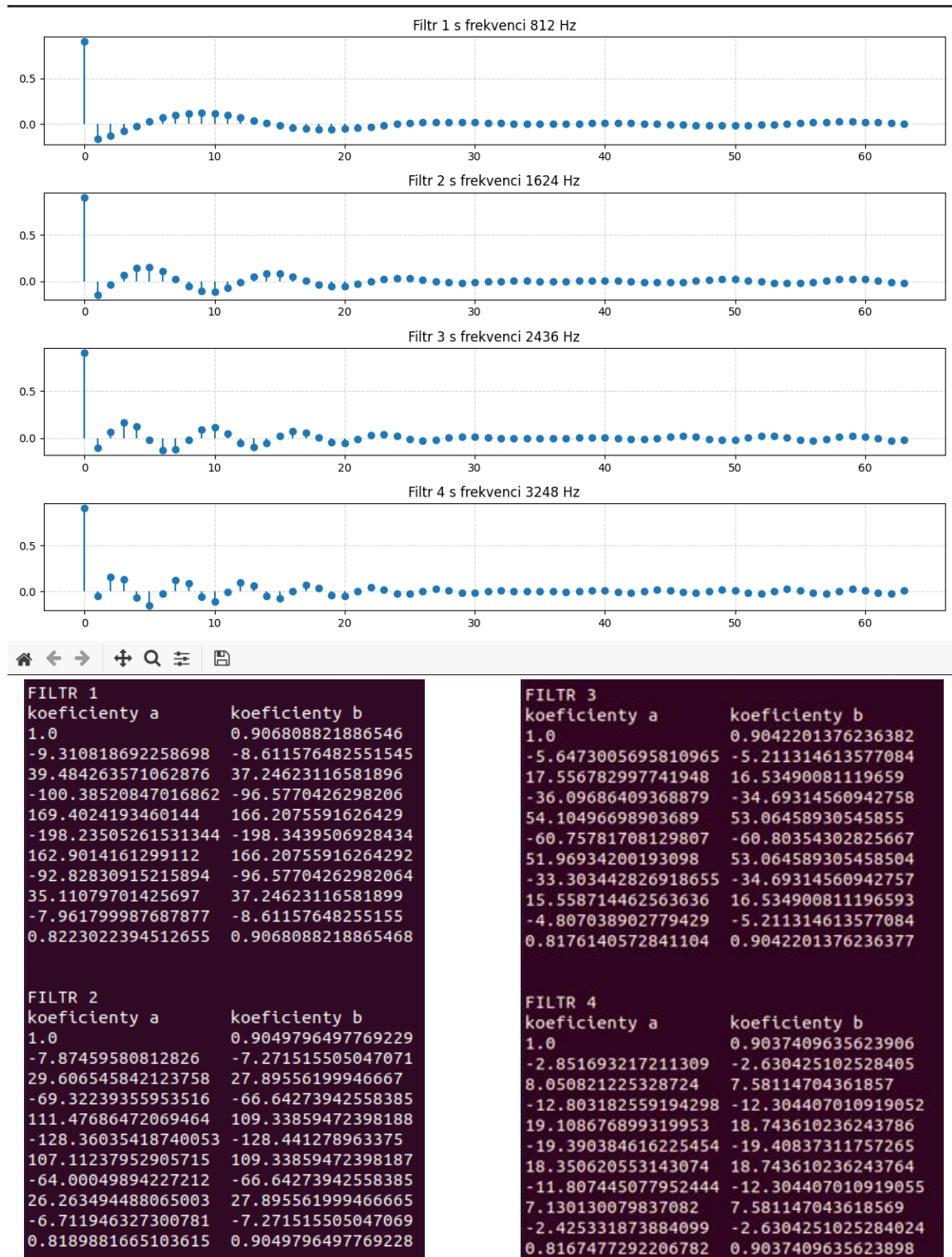
6. Generování signálu

Nejprve jsem si vytvořil pole, které jsem pomocí cyklu zaplnil vzorky. Následně jsem vytvořil 4 cosinusovky pomocí *numpy* funkce *np.cos*, kde jsem jako vstupní parametry, vyjma dvojky a π , využil již zmiňované pole se vzorky a vyčtenou frekvencí. Poté jsem všechny cosinusovky sečetl. Spektrogram jsem zobrazil stejně jako v úkolu 4 a pro zapsání signálu byla využita funkce *soundfile.write*.



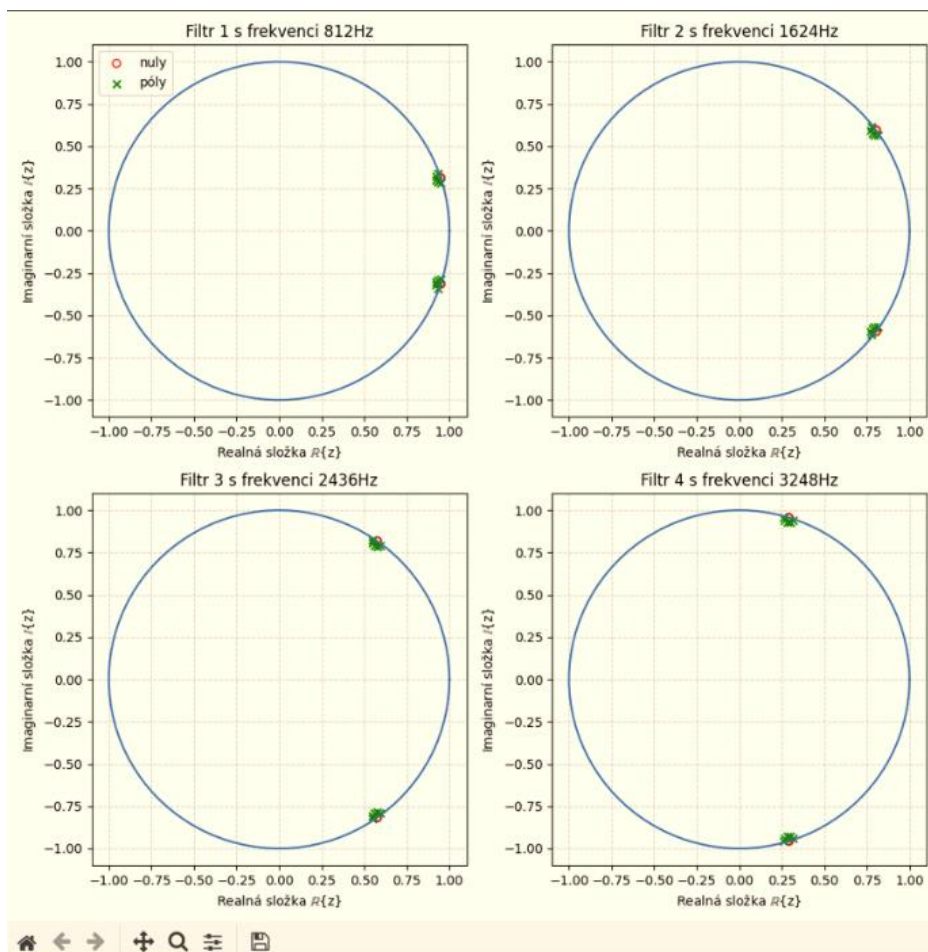
7. Čistící filtr

Vybral jsem třetí alternativu ze zadání, tedy návrh 4 pásmových propustí. Na toto jsem využil knihovnu *scipy*, konkrétně funkce *signal.buttord* a *signal.butter*. Při tvorbě jsem se řídil jejich dokumentací. Parametry jako závěrné pásmo, šíři přechodů do propustného pásma, zvlnění a potlačení šumu jsem nastavil dle hodnot v zadání. Impulzní odezvu jsem zjistil pomocí jednotkového impulzu a poté ji zobrazil. Koeficienty se mi vrátily po použití funkce *butter* a vzhledem k tomu, že mám 4 samostatné filtry, ukládal jsem si je postupně do pole, která jsem vypsal.



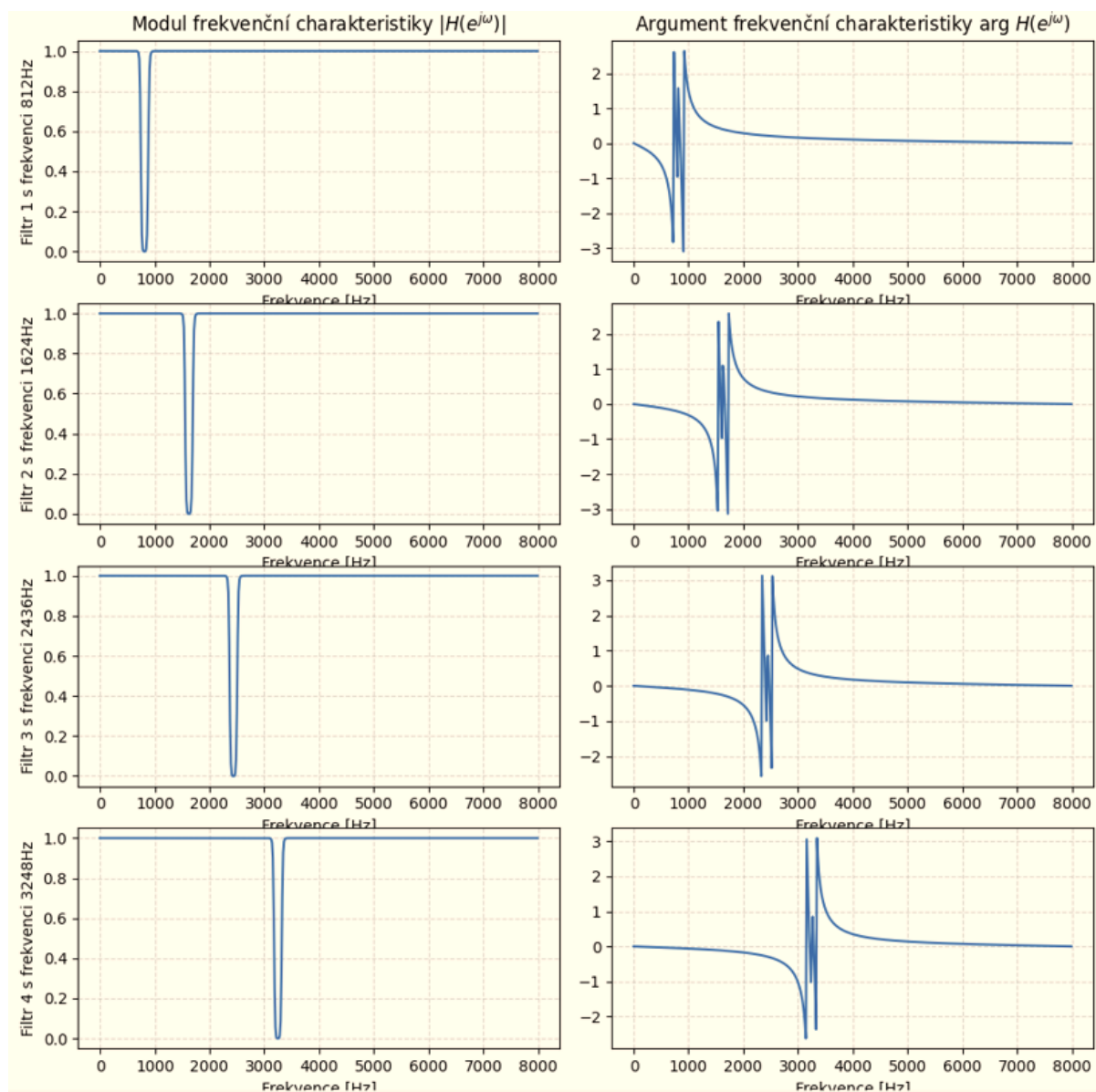
8. Nulové body a póly

Nuly a póly jsem získal opět funkcí *butter*, akorát s tím rozdílem, že jsem změnil output z *ba* na *zpk*, což mi namísto koeficientů vrácí nuly a póly. Graf jsem vytvářel tak, že jsem vytvořil kružnici, do které jsem promítl nalezené hodnoty pomocí funkce *scatter*.



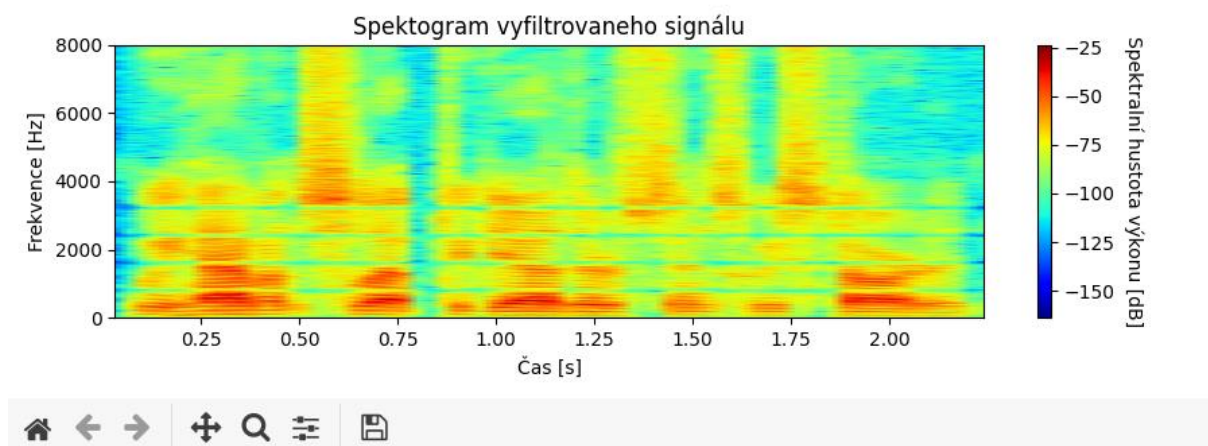
9. Frekvenční charakteristika

Frekvenční charakteristiky a jejich argumenty jsem získal pomocí funkce *freqz*, kam jsem nahrál získané koeficienty b, a . Poté jsem všechny frekvenční charakteristiky zobrazil. O jaký filtr se jedná je件zobrazeno na ose Y před grafem modulů frekvenčních charakteristik.



10.Filtrace

Filtraci jsem provedl pomocí postupného aplikování všech filtrů na původní signál. Signál ještě nebyl v rozsahu od -1 do +1, upravil jsem to tedy opět vydělením maximem absolutní hodnoty. Správné vyfiltrování jsem kromě poslechu ještě ověřil spektrogramem. Po spuštění nahrávky se zde stále nacházelo malé lupnutí na začátku, které potvrdil i spektrogram, vynuloval jsem tedy prvních 0.05 sekund nahrávky, kde se pípnutí dle spektrogramu nacházelo.



11.Zdroje

Převážná část informací pochází z příkladů od Katky Žmolíkové a dokumentací knihoven scipy, numpy a matplotlib