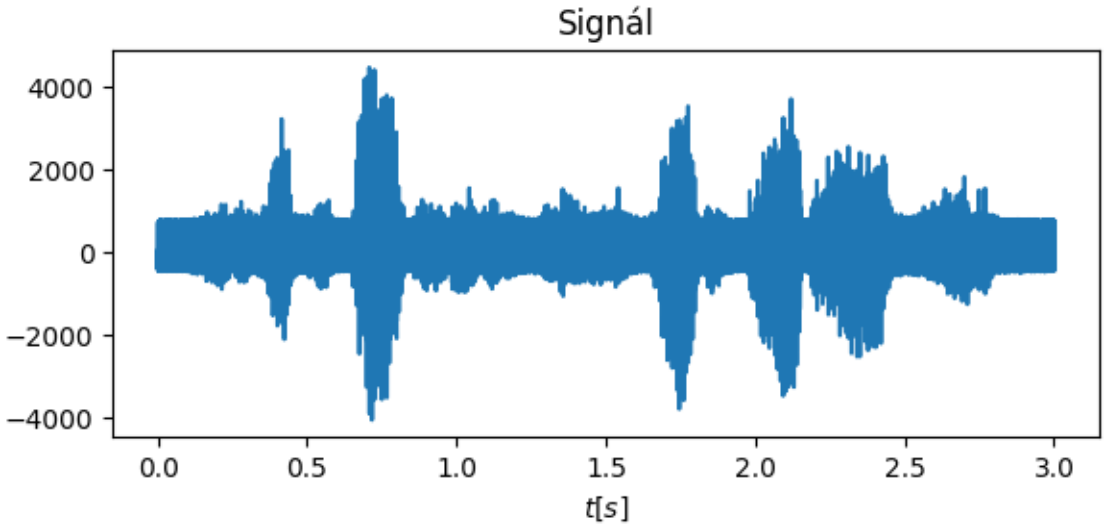
Úloha 1: Základy

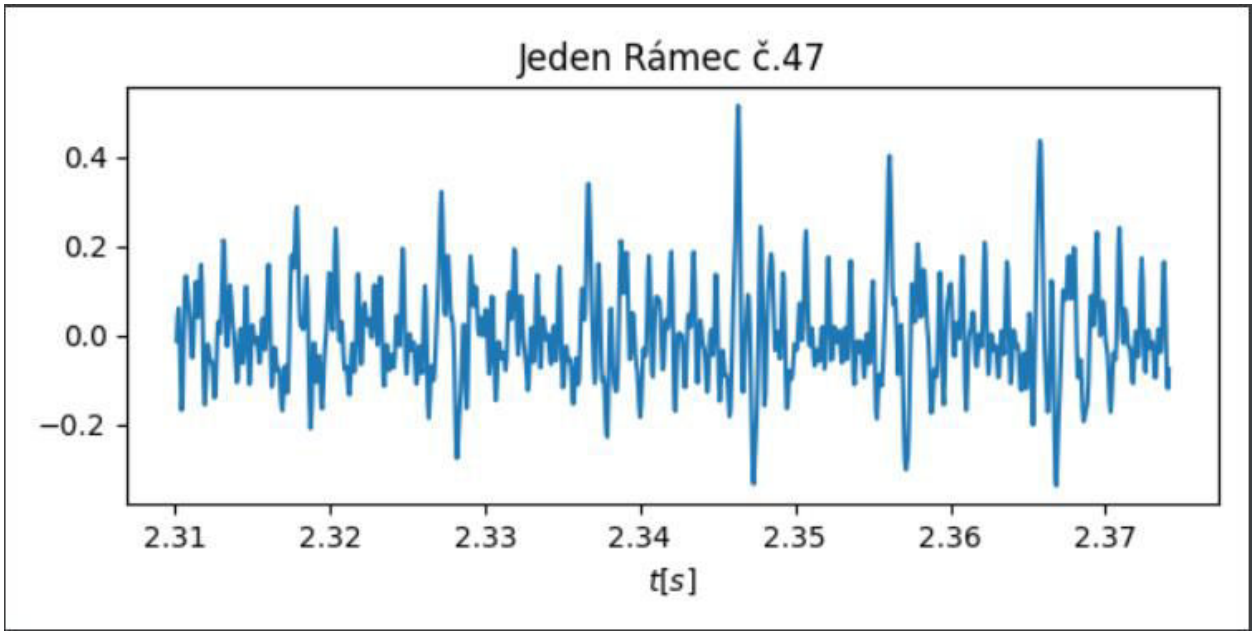
Vstupný signál má dĺžku 48026 vzorkov alebo 3.001625 sekúnd, jeho minimálna hodnota je -4052 a maximálna hodnota je 4486.

Zobrazenie signálu so slušnou časovou osou v sekundách:



Úloha 2: Predspracovanie a rámce

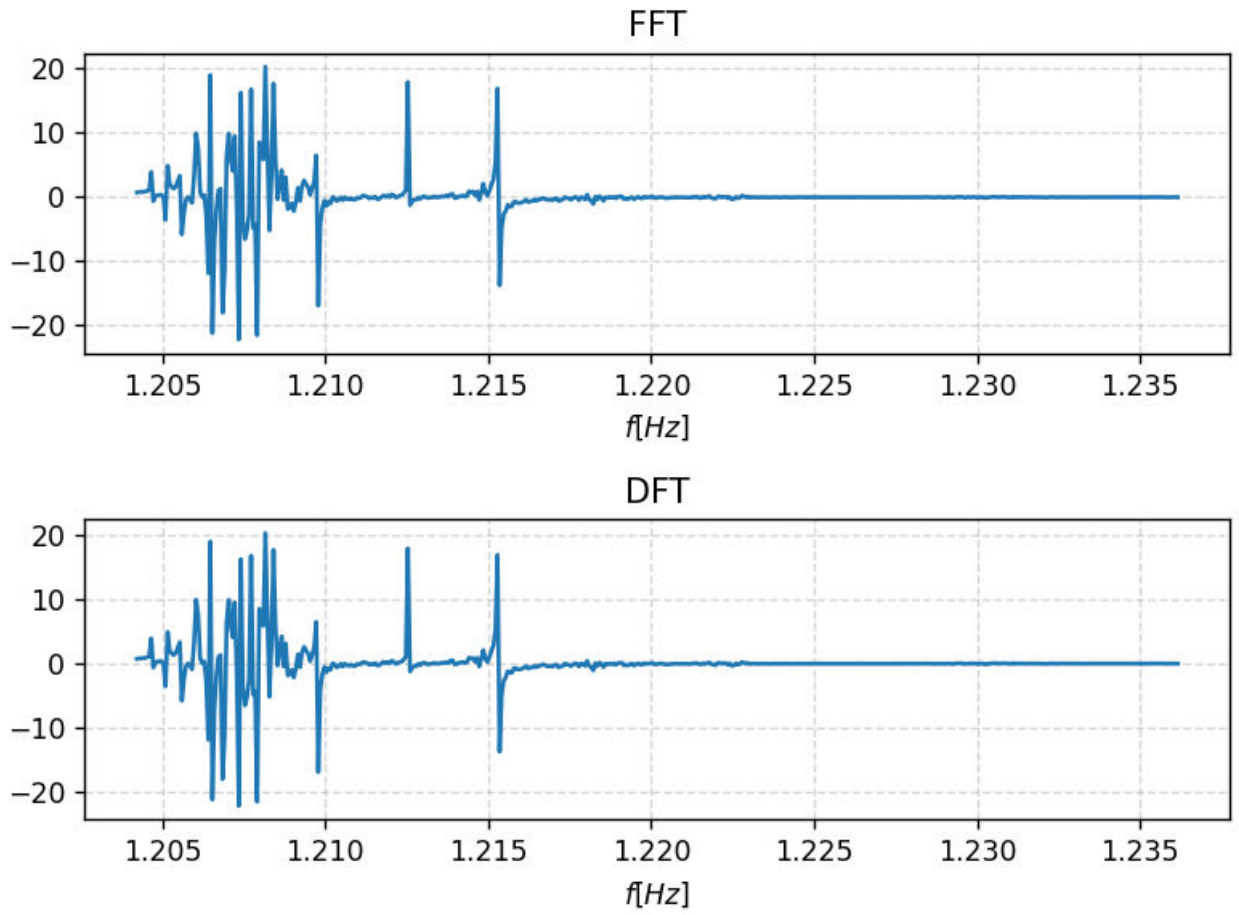
Rámec[47] podľa zadania:



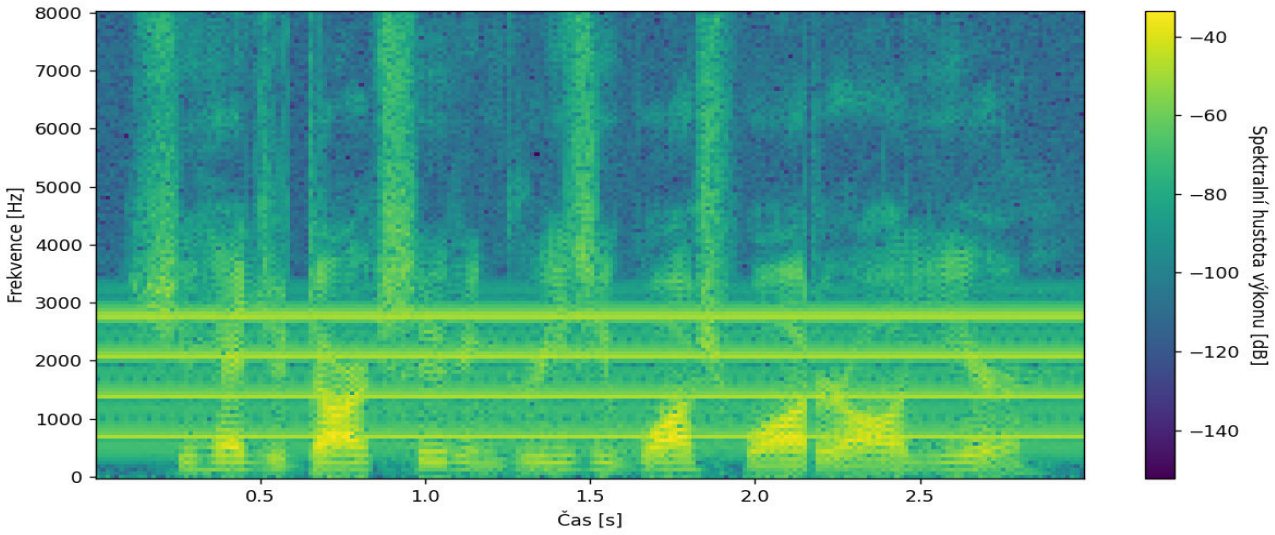
Úloha 3: DFT

Porovnanie FFT s mojou DFT implementovanou pomocou cyklov.

Vypadajú identicky ale moja implementácia DFT trvá poznateľne dlhšie na výpočet.



Úloha 4: Spektogram – Spektogram celého signálu

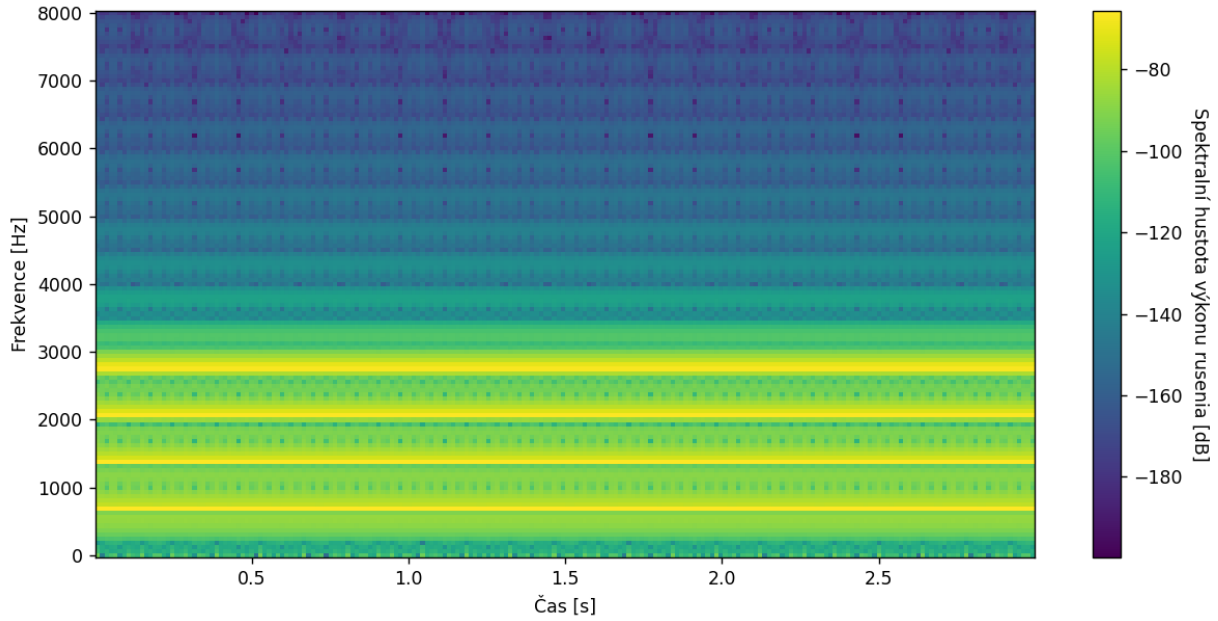


Úloha 5: Určenie rušivých frekvencí

Podľa spektogramu vieme približne určiť rušivé frekvencie (okolo násobkov 700Hz) a následne ich presne učiť pri implementácí filtrov pásmovej zádrže, rušivé frekvencie teda nastávajú na frekvenciách 693 Hz, 693*2 Hz, 693*3 Hz a 693*4 Hz, sú teda harmonicky vztažené.

Úloha 6: Generovanie signálu

Generujem kosínusovku pre každú rušivú frekvenciu a následne tieto cosínusovky sčítam a pomocou funkcie sf.write() vygenerujem daný signál.



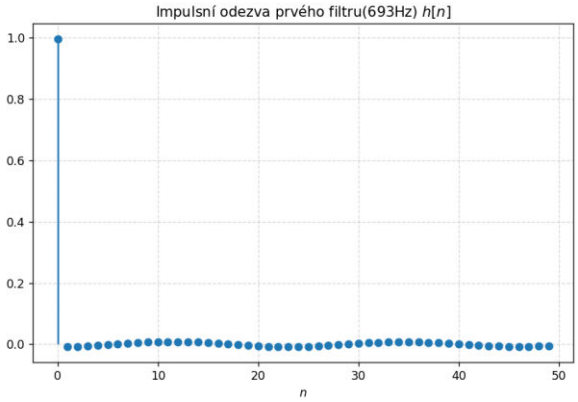
Úloha 7: Čistiaci filter

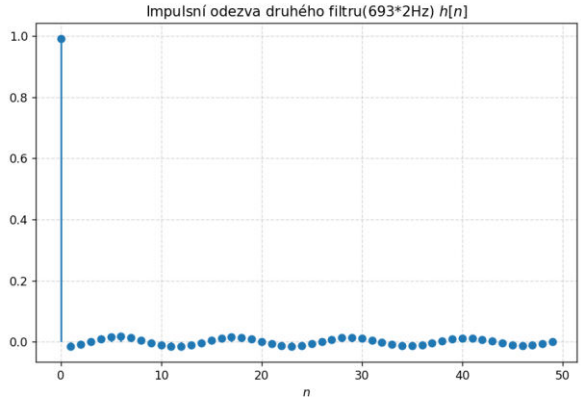
Zostrojil som štyri filtre typu pásmová zádrž, čiže pre každú rušivú frekvenciu jeden filter.

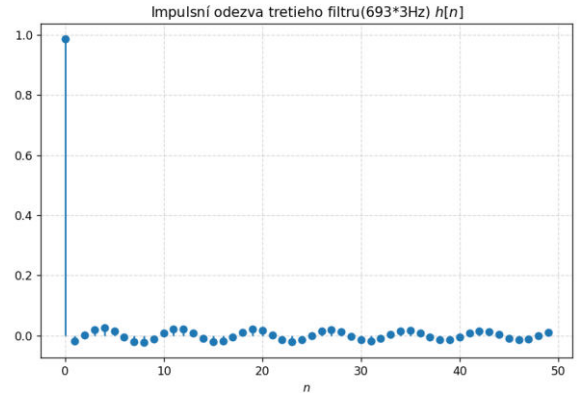
Návod na zostrojenia filtra typu pásmová zádrž v pythone som čerpal zo stránky Stack Overflow a generovanie grafov impulzných odoziev podľa Jupyteru.

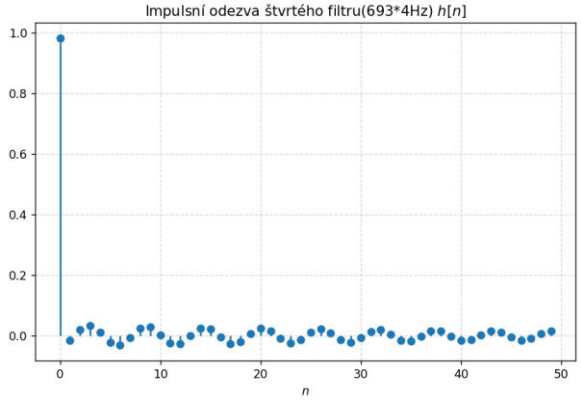
```
Koeficienty 1. Filtru a: [ 1.         -1.91769741  0.99096955] b: [ 0.99548477 -1.91769741  0.99548477]
Koeficienty 2. Filtru a: [ 1.         -1.69561774  0.98201991] b: [ 0.99100996 -1.69561774  0.99100996]
Koeficienty 3. Filtru a: [ 1.         -1.35127862  0.97314965] b: [ 0.98657482 -1.35127862  0.98657482]
Koeficienty 4. Filtru a: [ 1.         -0.91099034  0.96435735] b: [ 0.98217867 -0.91099034  0.98217867]
```

Impulzné odozvy filtrov:





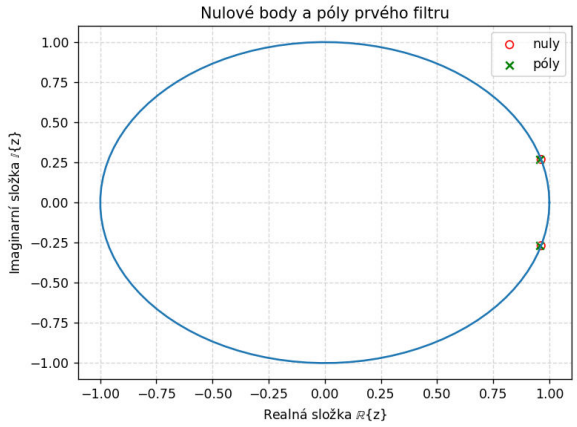


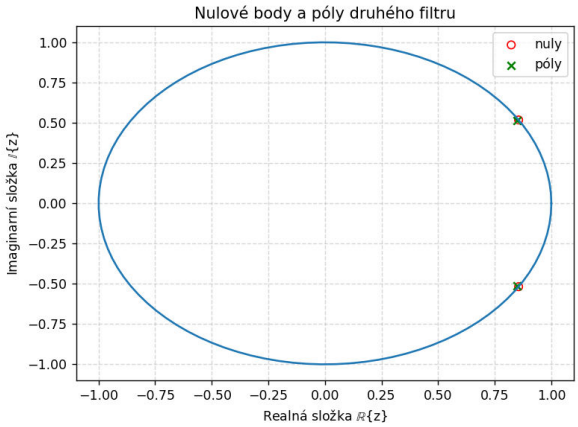


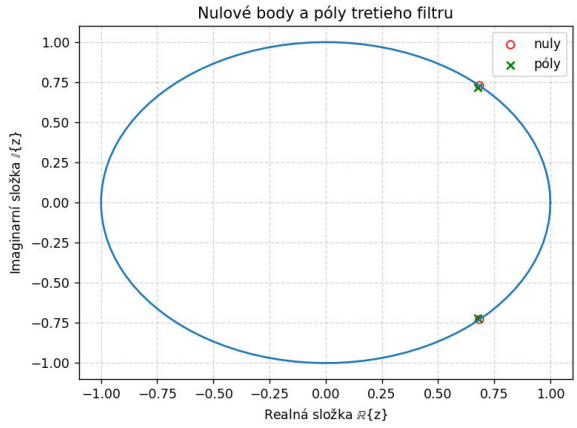
Marián Backa (xbacka01)

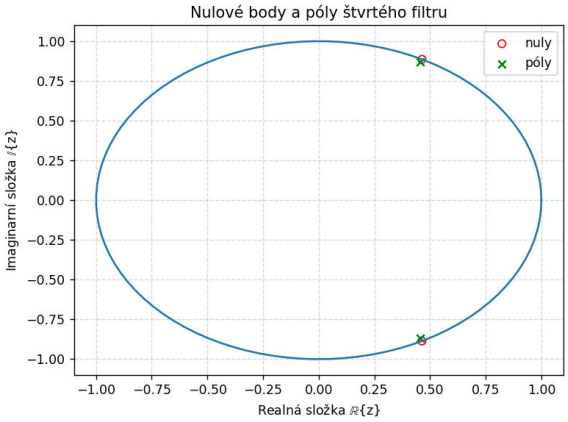
Úloha 8: Nulové body a póly

Postupoval som podľa Jupyteru.





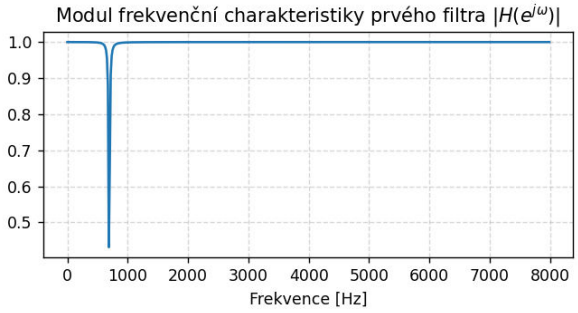


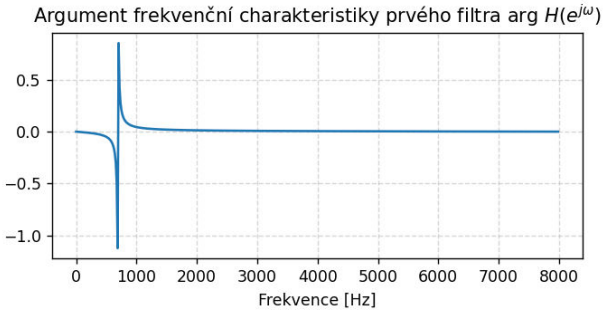


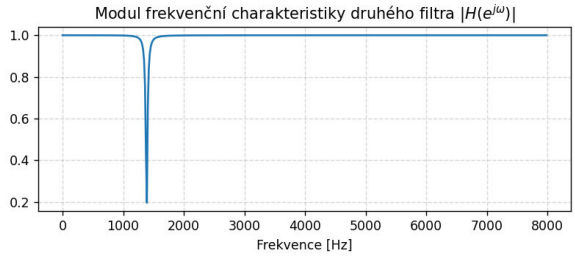

```
Prvy filter: Poly:  [0.9588487+0.26754198j 0.9588487-0.26754198j]  Nuly:  [0.96319776+0.26879374j 0.96319776-0.26879374j]
Druhy filter: Poly:  [0.84780887+0.51306923j 0.84780887-0.51306923j]  Nuly:  [0.85549985+0.51780306j 0.85549985-0.51780306j]
Treti filter: Poly:  [0.67563931+0.71879147j 0.67563931-0.71879147j]  Nuly:  [0.68483332+0.72869975j 0.68483332-0.72869975j]
Štvrtý filter: Poly:  [0.45549517+0.86998937j 0.45549517-0.86998937j]  Nuly:  [0.46375999+0.88596088j 0.46375999-0.88596088j]
```

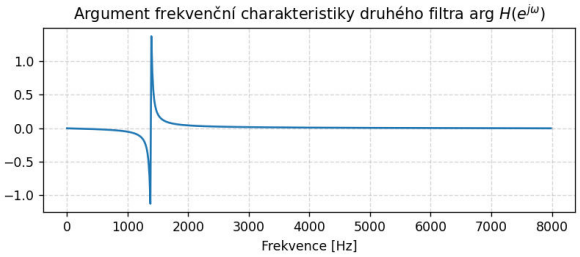
Úloha 9: Frekvenčná charakteristika

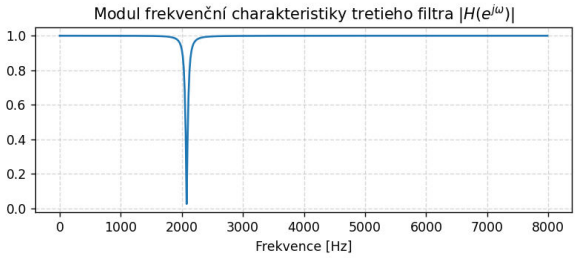
Postupoval som podľa Jupyteru.

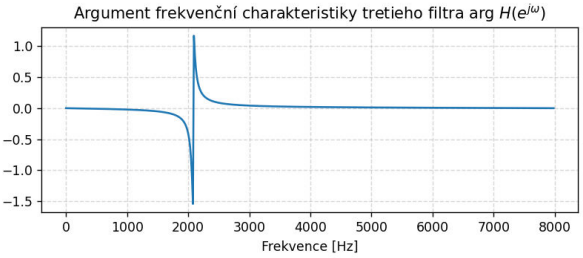


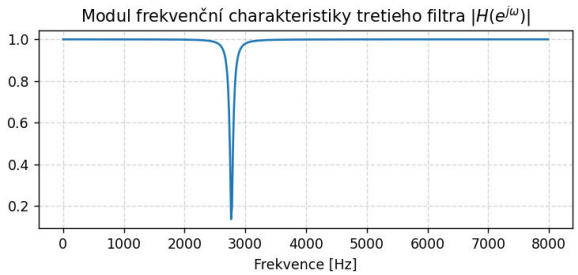


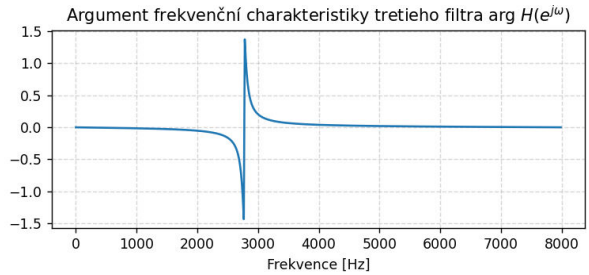






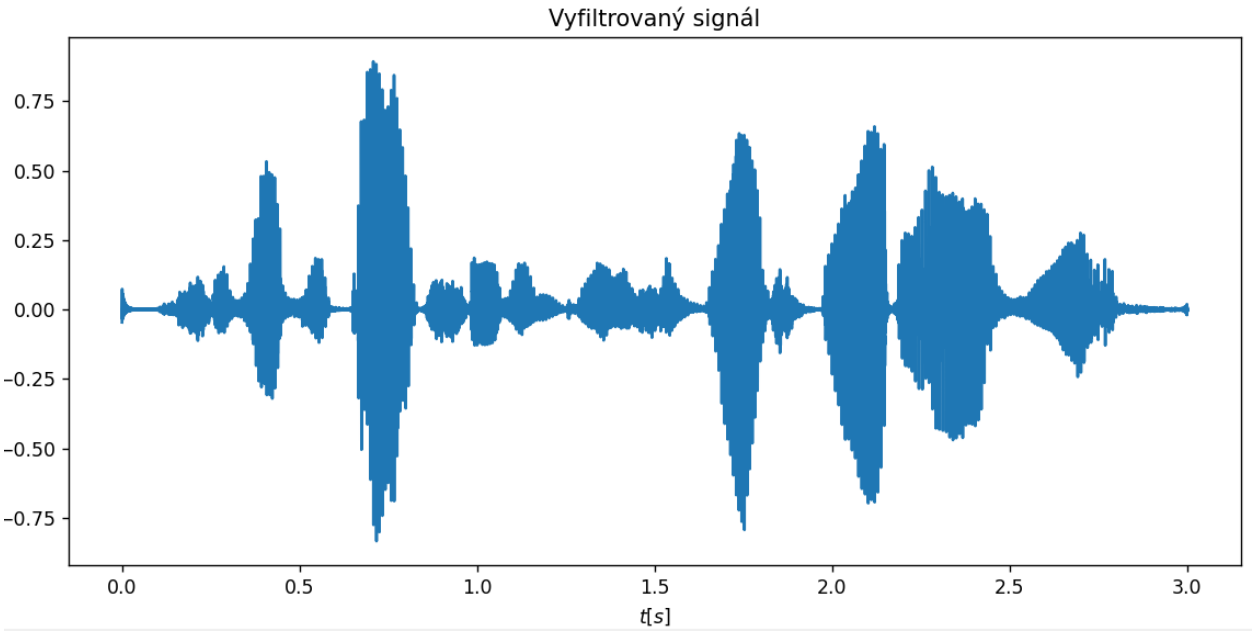






Ako vidíme filter správne potlačuje práve naše štyri rušivé frekvencie a prepúšťa ostatné nezmenené.

Úloha 10: Filtrace



Signál zostal v rozsahu -1 až 1 aj po filtrovaní. Na grafe môžeme vidieť, že sme sa zbavili stále prítomnej rušivej zložky(štyroch kosínusoviek tak, že medzi slovami sa signál hodnotou blíži k 0 a neostáva tam žiadne konštantné rušenie (viz. graf úlohy 1) alebo sa pozrieme na Spektogram a vidíme, že rušivé frekvencie sú úplne potlačené.