

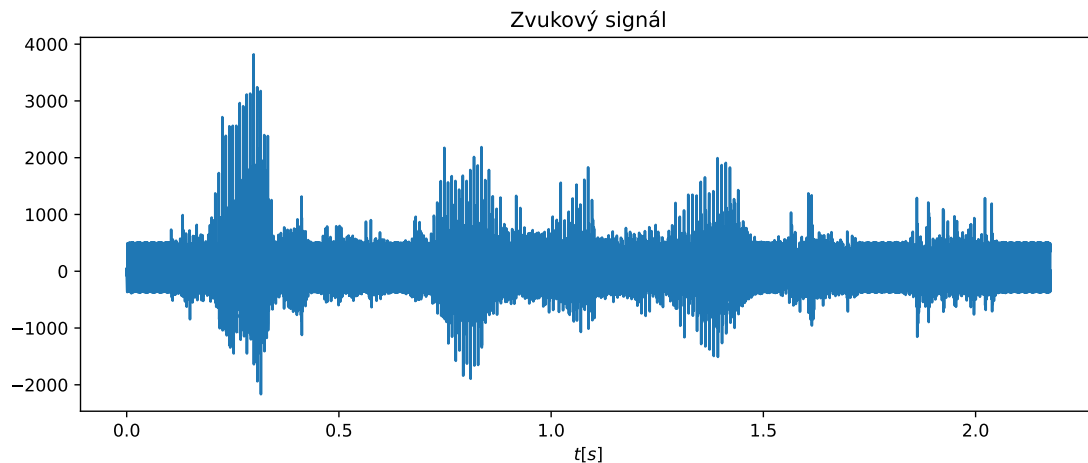
Protokol

Jméno: Mikheda Vladislav

Login: xmikhe00

1 Základy

Načetl jsem signál pomocí „wavfile.read“. Signál byl načten nenormalizovaný.



Délka ve vzorcích = 34816[vzorků]

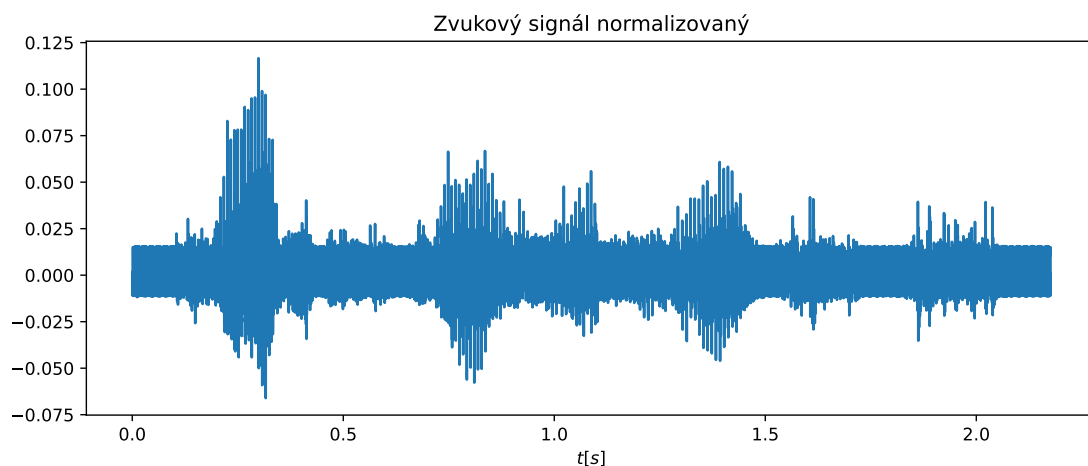
Délka v sekundách = 2.176[s]

Minimální hodnota = -2166

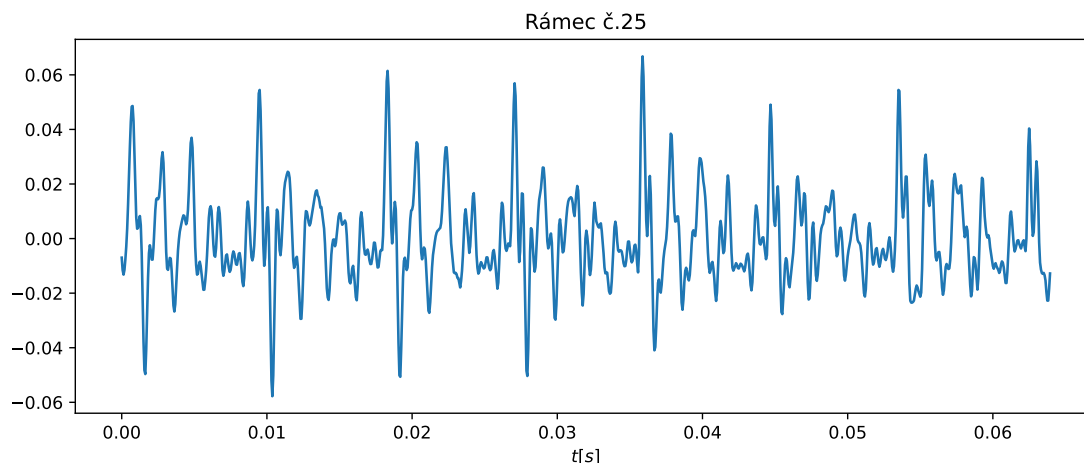
Maximální hodnota = 3820

2 Předzpracování a rámce

Normalizoval jsem signál pomocí dělením na $2^{15} = 32768$. Je to maximální hodnota 16 bytného integera se šířkou kterého signál byl načten. Po normalizaci signál má hodnoty v rozhraní od -1 do 1.



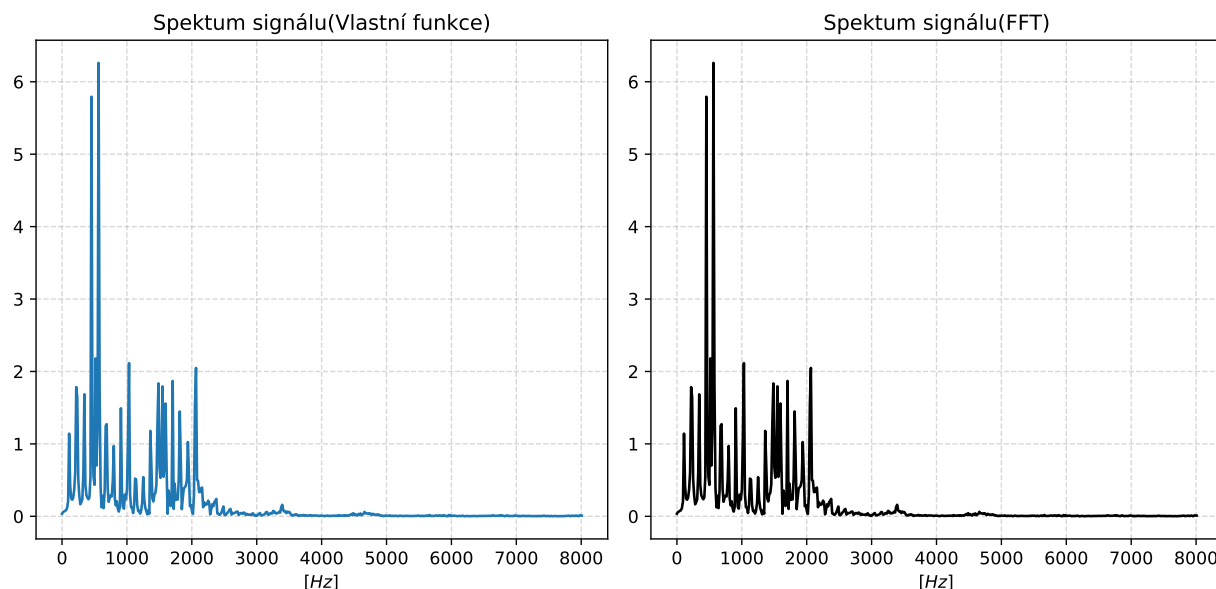
Dal signál byl rozdělen na úseky. Pro zobrazení jsem vybral rámeček číslo 25.



3 DFT

DFT byla definovaná pomocí vlasti funkce která na vstup přijímá pole s signálem. Pro vypracování DFT potřebujeme dvě kvadratické matice $N \times N$ kde N je počet vzorku přijímaného signálu, ve kódu matice X a Y . Matice X má v řádcích čísla od 1 do N a reprezentuje n z formule, matice Y v řadě obsahuje N stejných čísel a s každým řádkem se tato čísla zvyšují o 1 a reprezentuje k z formule. Dal čísla v maticích vynásobíme číslem na stejné pozice a dostaneme matici která ve sobe pro každý vzorek signálu má $k * n$. Dále musíme každý vzorek v matici použít jako mocninu pro $e^{-j\frac{2\pi}{N}}$ a dostaneme $e^{-j\frac{2\pi}{N}kn}$ teď nám jen zůstává vynásobit signál této matice a budeme mít N koeficienty DFT. Navržena funkce neobsahuje cykly.

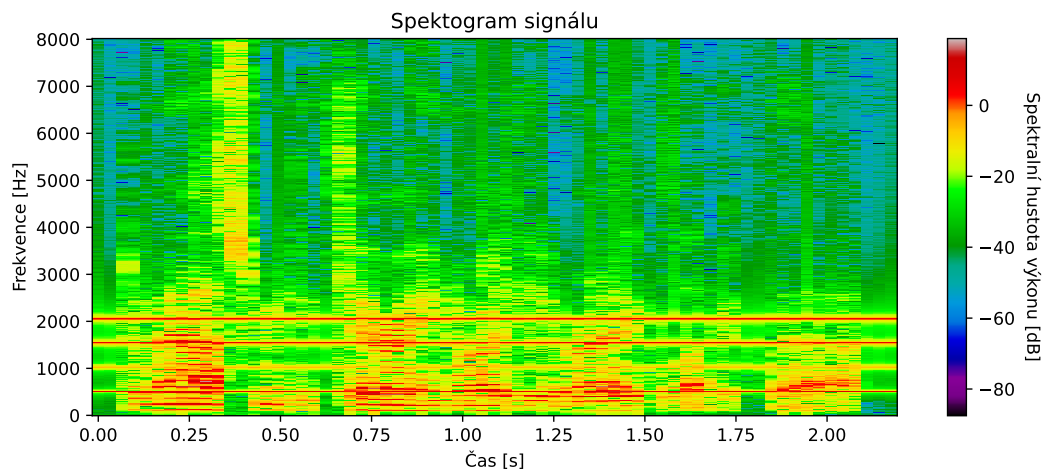
$$\text{Použitý vzorek } X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn}$$



Vidíme z grafů že výsledky jsou stejné, také jsem použil pro srovnání funkci „np.allclose“ a dostal jsem *True*

4 Spektrogram

Pro výpočet DFT celého signálu jsem použil „np.fft.fft“, pak modul koeficientů jsem upravil pomocí formule a zobrazil jsem spektrogram pomocí „plot.pcolormesh“ pro zobrazení jsem použil půlku frekvence a koeficientů



5 Určení rušivých frekvencí

Pro určení frekvence jsem navrhl vlastní funkci, která na vstup přijímá rámec signálu. Použil jsem rámec číslo 2 protože není tam řeč, jen 4 rušivé cosinusovky. Funkce bere maximální hodnotu signálu převádí ji na index v pole signálu, na stejném indexu v pole frekvenci nachází frekvence pro daný vzorek. Po výpočtu jsem dostal:

Frekvence_1 = 516[Hz] Frekvence_2 = 1032[Hz] Frekvence_3 = 1548[Hz] Frekvence_4 = 1548[Hz]

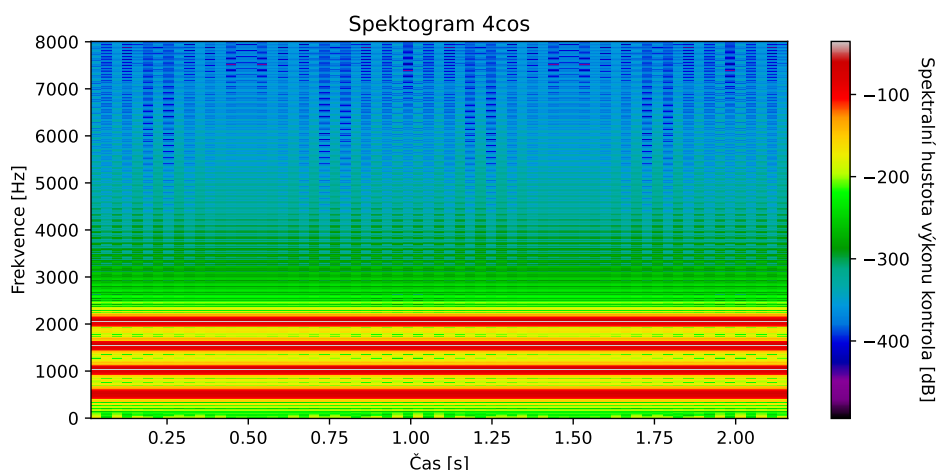
Zkontrolujeme harmonickou vztaženost cosinusovek:

Frekvence_2 = 516 · 2 Frekvence_3 = 516 · 3 Frekvence_4 = 516 · 4

Vidíme že cosinusovky jsou harmonicky vztaženy.

6 Generování signálu

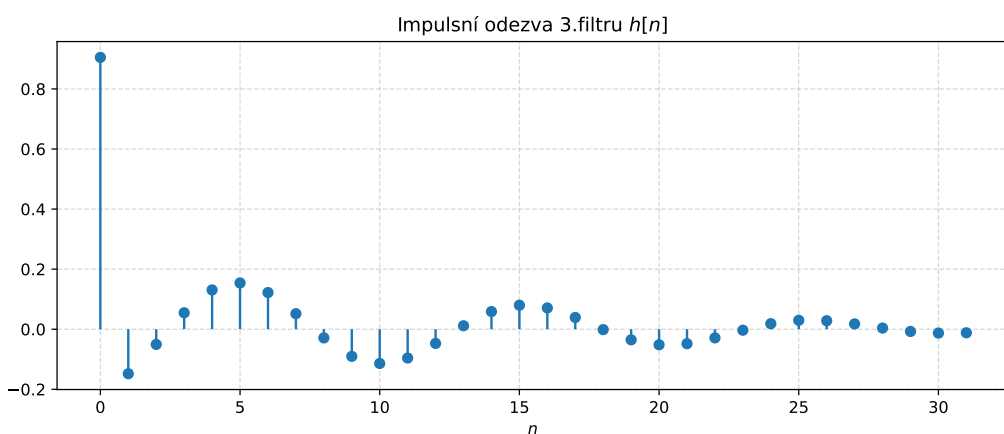
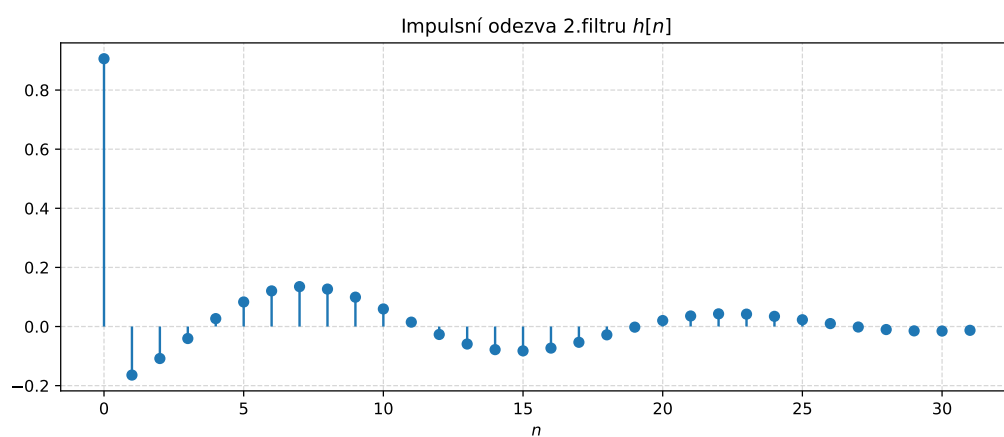
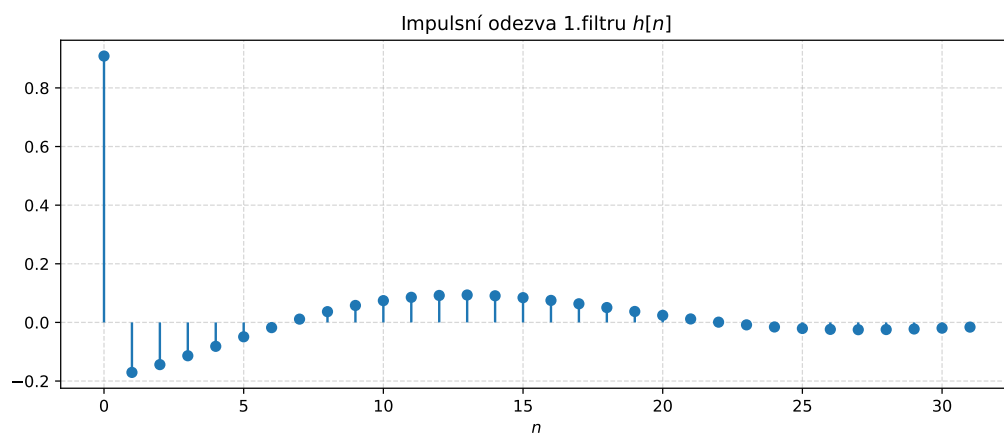
Vygeneroval jsem každou cosinusovku zvlášť pak je spojil. Spekter byl navržen pomocí funkce „spectrogram“



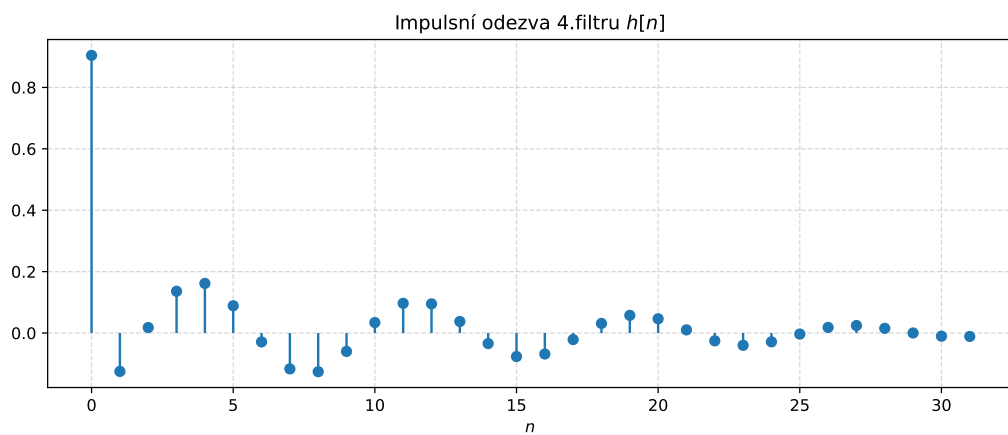
Ze spektrogramu a z poslechu jde vidět že frekvence byly určeny správně.

7 Čisticí filtr

Vygeneroval jsem čtyři filtry typu pásmová zádř. Pro generování a i b koeficientů jsem použil „`scipy.signal.butterd`“ a „`scipy.signal.butter`“. Mam hodně koeficientů a proto jsou v textovém dokumentu¹. Impulsní odezva byla vygenerována pomocí filtrací jednotkového impulsu, pro filtraci byla použita funkce „`scipy.signal.lfilter`“

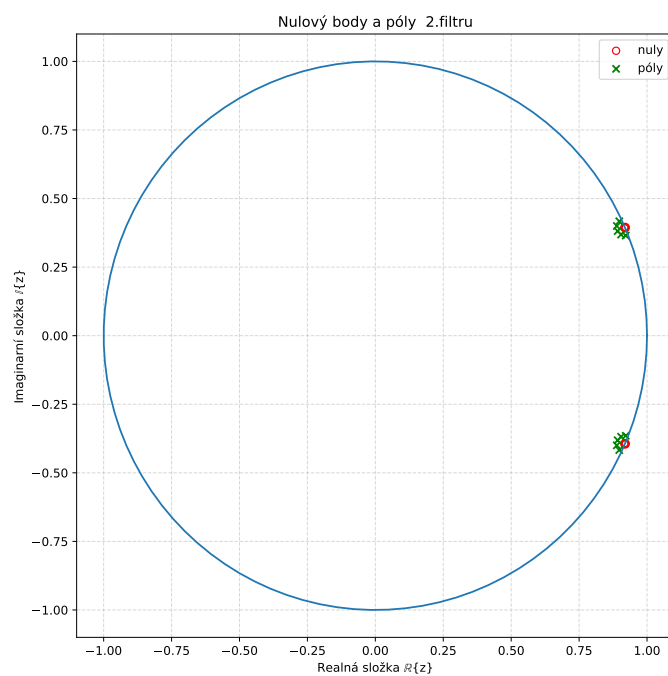
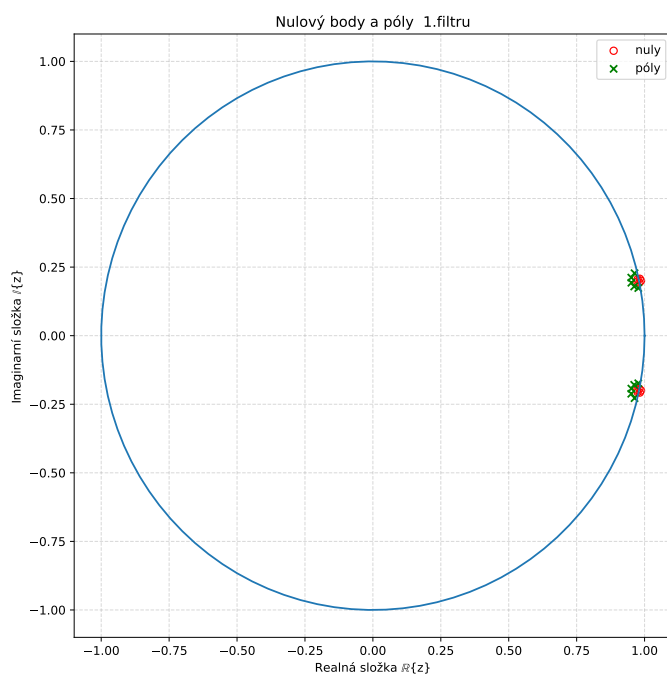


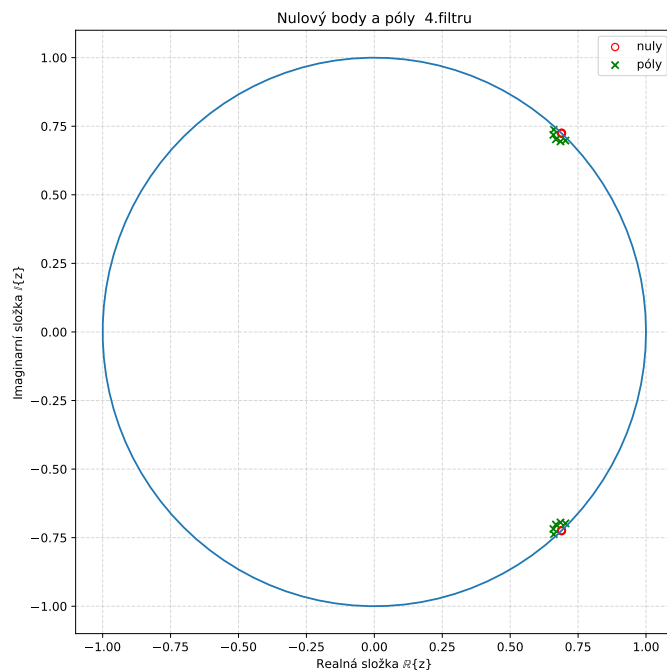
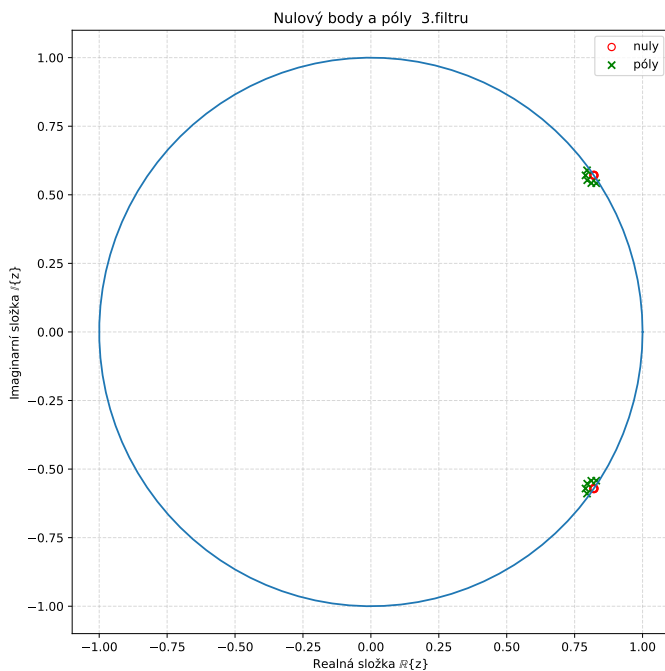
¹/src/A_B.txt



8 Nulové body a póly

Nulové body a póly byly vypočteny pomocí funkce „`scipy.signal.tf2zpk`“

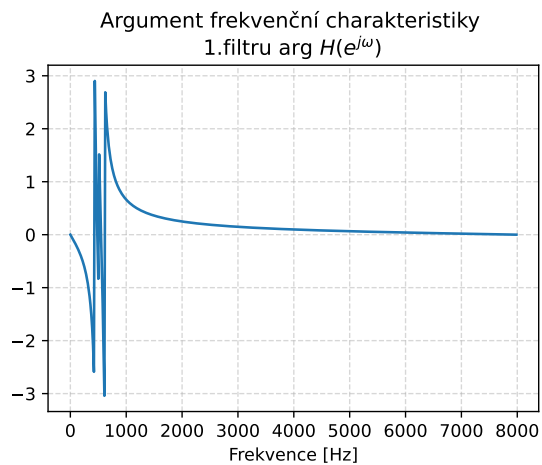
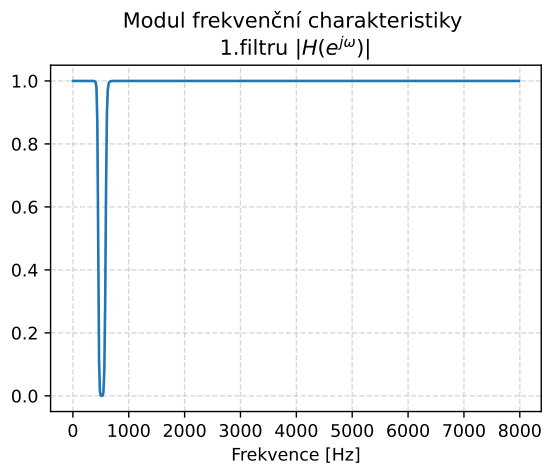


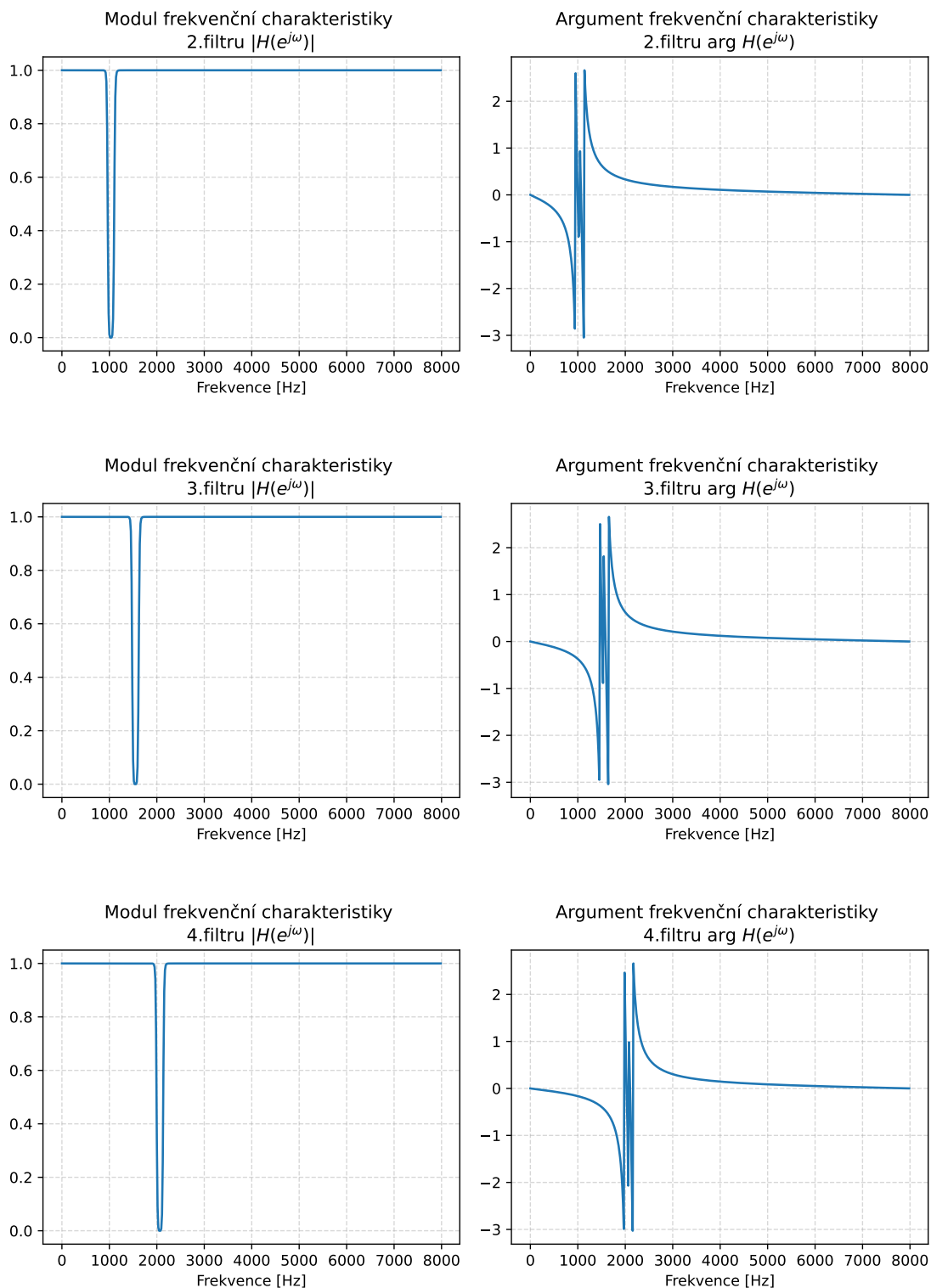


Při zvětšení jde vidět, že body na stejném místě ale vedle.

9 Frekvenční charakteristika

Zobrazil jsem frekvenční charakteristiku čtyř filtrů pomocí funkce „scipy.signal.freqz“





Vidíme z frekvenční charakteristiky ze filtr filtruje na zadaných námi frekvencích

10 Filtrace

Filtrace signálu byla provedena pomocí funkce „`scipy.signal.lfilter`“. Vstupní signál byl propuštěn postupně přes všechny čtyři filtry. Provedl jsem kontrolu, signál je v rozhraní od -1 do 1. Pro uložení signálu jsem použil

„denormalizace“ pomocí násobení na největší hodnotu 16 bytného integeru pak jsem převedl signál do 16 bitového intu. Podle zadání signál musí být uložen se šířkou 16 bit a pak signál byl uložen pomocí funkce „wavfile.write“. Poslechem bylo ověřeno, že signál je vyčištěn. Po filtraci už neslyšíme rušivé cosinusovky.