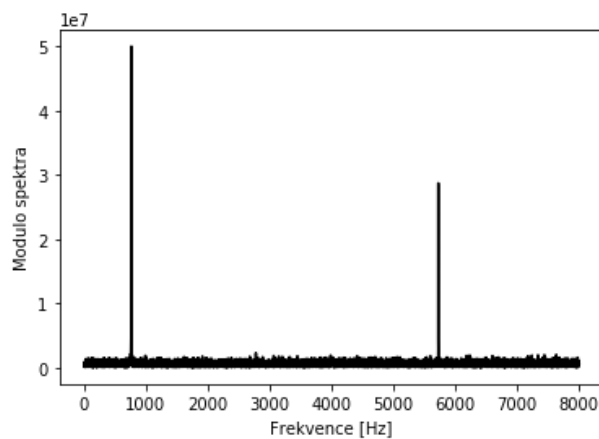


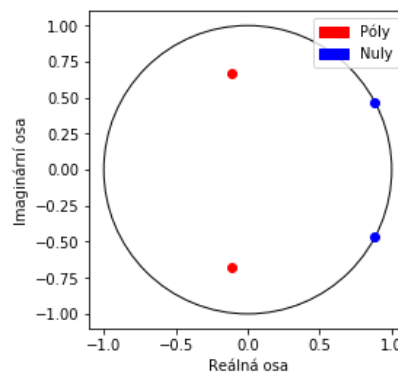
- 1) Vzorkovací frekvence: 16 000  
Délka ve vzorcích: 16 000  
Délka v sekundách: 1 s

```
from pylab import*
from scipy.io import wavfile
# Nacteni souboru
sampFreq, snd = wavfile.read('xendry02.wav')
# Vzorkovací frekvence
sampFreq
# Delka ve vzorcích
len(snd)
# Delka v sekundách
len(snd)/ sampFreq
```

- 2) Je potřeba vypočítat DFT pomocí funkce `fft`, vygenerovat si pole které představuje osu x a poté zobrazit v grafu absolutní hodnoty modulu spektra pomocí funkce `plot` v knihovně `matplotlib`. Je potřeba zobrazovat absolutní hodnoty, jinak by došlo k zahazení imaginární části.

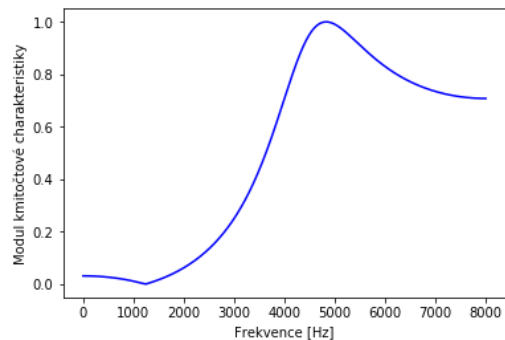


- 3) 766 Hz  
Je potřeba hledat maximum pomocí absolutních hodnot komplexních čísel, které vrátila funkce `fft`.
- 4) Nuly a póly přenosové funkce se dají získat pomocí funkce `tf2zpk`, která je v knihovně `scipy.signal`.



Filtr je stabilní, jelikož póly mají absolutní vzdálenost menší než 1.

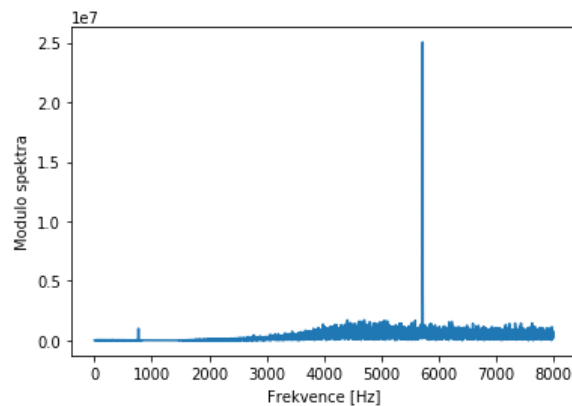
- 5) Kmitočtová charakteristika se dá získat pomocí funkce `freqz`, kde se jako argumenty zadají pole, ve kterých jsou uloženy koeficienty filtru a frekvenci pro kterou chceme získat kmitočtovou charakteristiku.



S

Jedná se o horní propust, jelikož vidíme že nízké frekvence filtr nepropouští, ale vysoké ano.

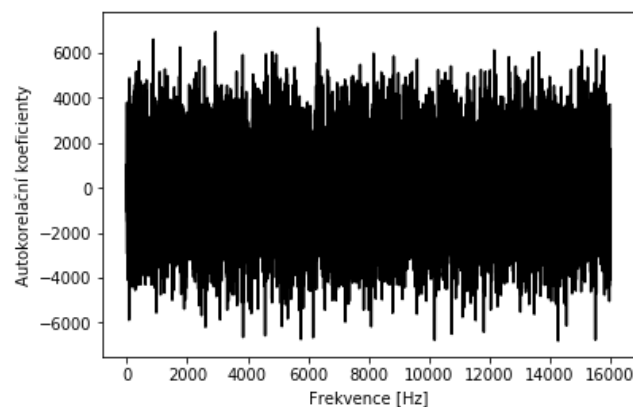
- 6) Filtrovat signál je možno pomocí funkce `lfiter` z knihovny `scipy.signal`. Dále se z výsledku vypočítá diskretní furierova transformace pomocí funkce `fft`. Modul spektra se v grafu zobrazuje jako absolutní hodnota, jinak by došlo k zahazení imaginární části.



- 7) 5726 Hz

Postup je stejný jako v úloze 3, pomocí absolutních hodnot.

- 9) Implementoval jsem vzoreček pomocí 2 cyklů, výsledek jsem postupně ukládal do pole, které jsem následně zobrazil pomocí `matplotlib.pyplot`. Bylo potřeba kontrolovat hranice signálu, aby se nešlo přes hranice signálu.



- 10) -2932.9836065573772

Položka na desátém indexu v poli, které bylo vygenerováno v minulém příkladu.