Vysoké učení technické v Brně
Fakulta informačních technologií

ISS

ISS/VSG Projekt 2021/22

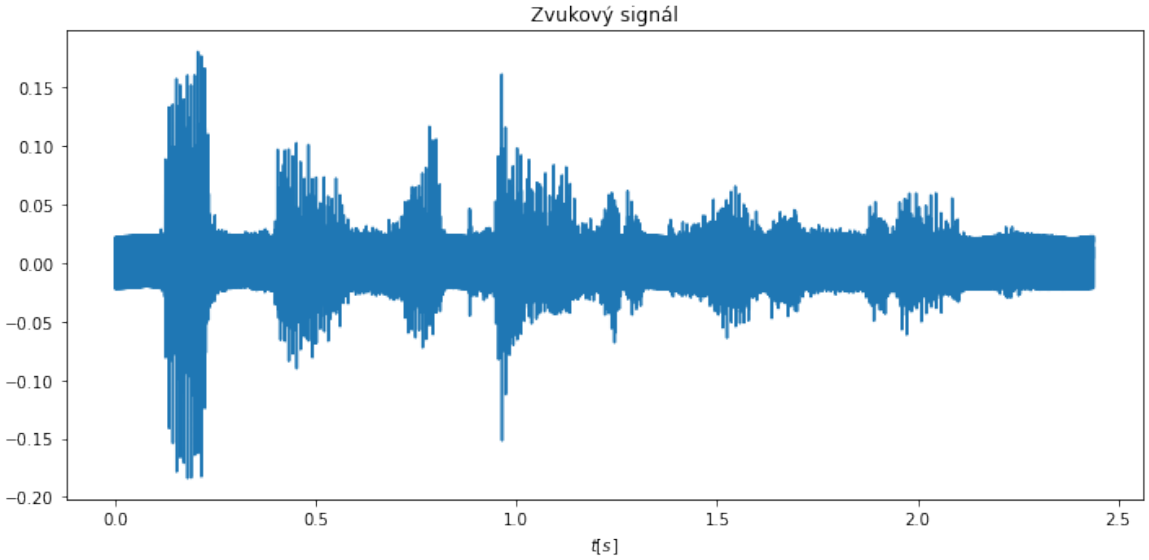
7. ledna 2022 Jan Řezníček(xrezni29)

1 Stardantní zadání

1.1 Základy

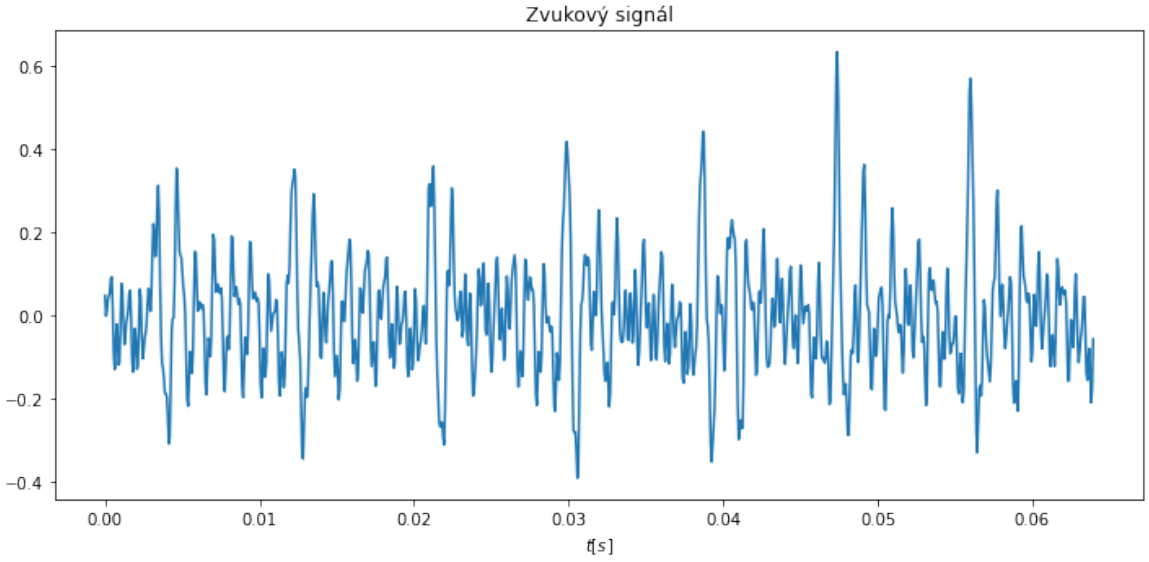
Pomocí funkce read z knihovny soundfile jsme načetli signál z xrezn29.wav. Poté jsme singál zobrazili pomocí funkce plot z knihovny matplotlib.pyplot. Na ose Y vidíme hodnotu jednotlivých vzorků a na ose X vidíme čas.

Maximální hodnota je **0.1806640625**
Minimální hodnota je **-0.183929443359375**
Délka ve vzorcích je **39015**
Délka v sekundách je **2.438375**



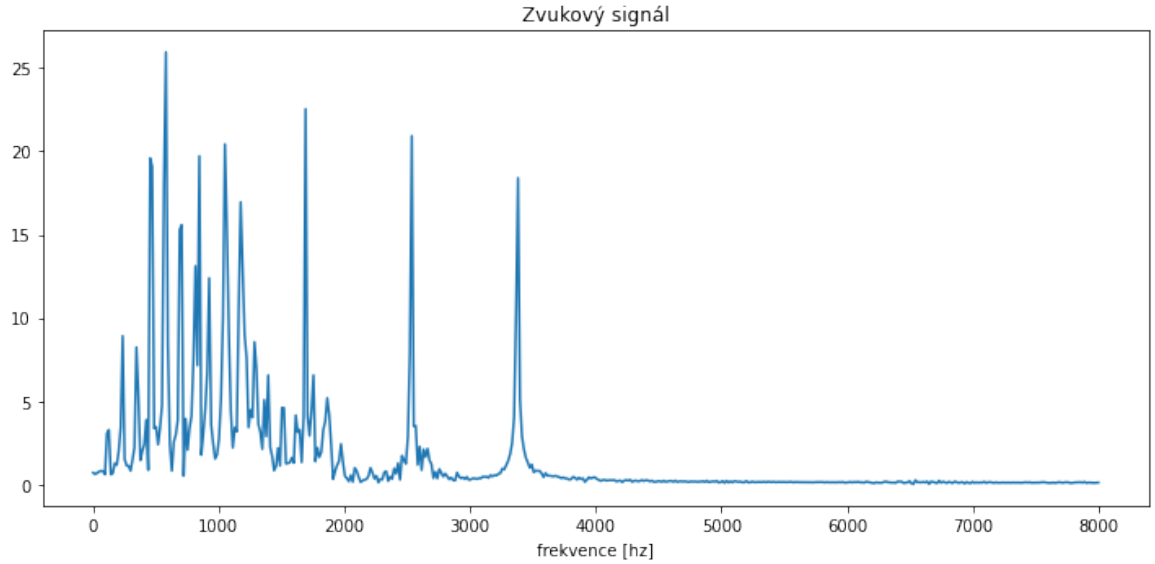
1.2 Předpracování rámce

Nejprve si pomocí funkce mean vypočítáme střední hodnotu. Pak pomocí funkce max a abs vrátíme nejvyšší hodnotu. Poté od signálu odečteme střední hodnotu a vydělíme ho maximální hodnoutou. Pak signál rozdělíme na rámce s velikostí 1024 vzorků a překrytím 512 vzorků. Nakonec rámec zobrazíme pomocí funkce plot.



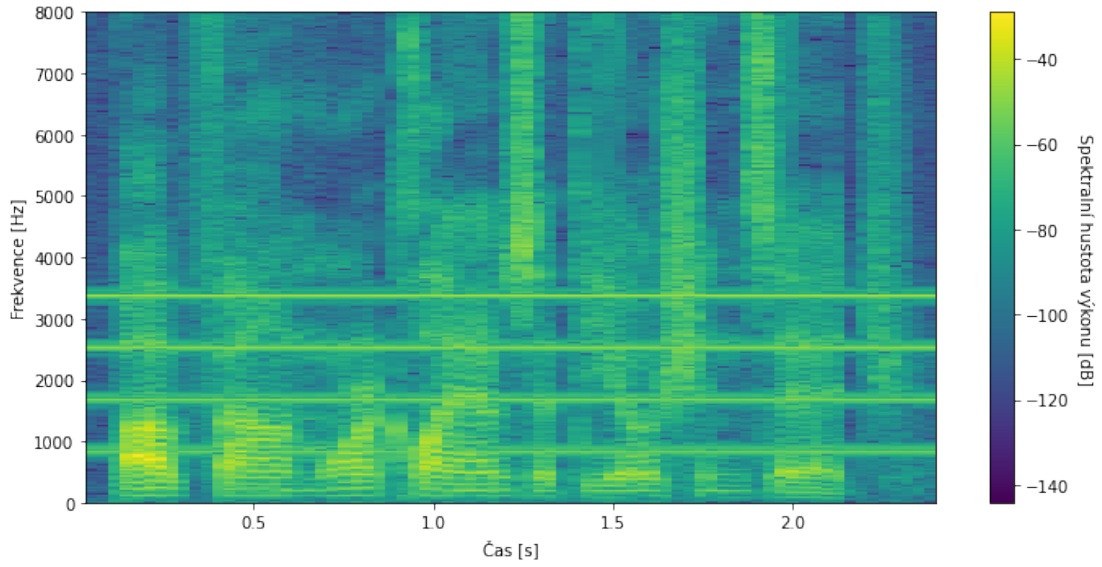
1.3 DFT

Diskrétní furierova transformace je implementována jako rekurzivní násobení matic pouze pro 1024 vzorků. První si celý rámec rozložíme na sudé a liché čísla, díky vlastnosti symetrie víme že budeme počítat jen pro polovinu vzorků sudých a lichých. Skončí poté co bude velikost vstupnu menší rovna 1.



1.4 Spectogram

Z signálu a frekvence dostaneme spectogram pomocí funkce spectogram z knihovny scipy.signal další parametry co jsme nastavili je překrytí 512 a délka rámce 1024 následně spectogram zobrazíme.

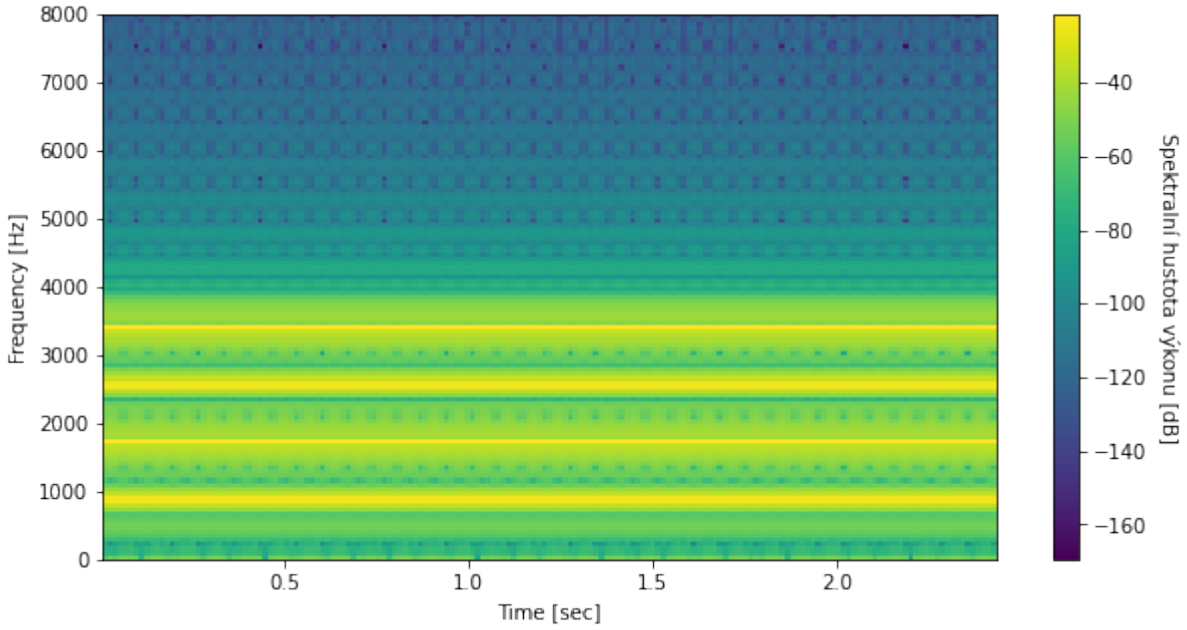


1.5 Určení rušivých frekvencí

Rušivé frekvence jsem určil tak že jsem dal celý signál jen po vzorek který je mocninou čísla 2 do DFT zobrazil absolutní hodnotu poté jsem zobrazil graf a následně jsem vypsal všechny indexy/(počet vzorků/frekvence) co jsou větší jak 600 jde to vidět v grafu. A to jsou naše rušivé frekvence. Frekvence které nám program vypsal jsou **842.28515625 1684.5703125 2526.85546875 3369.140625,** zde jasně vydíme, že jsou frekvence harminicky vztažné(zhruba) každá je násobek té nejnižší.

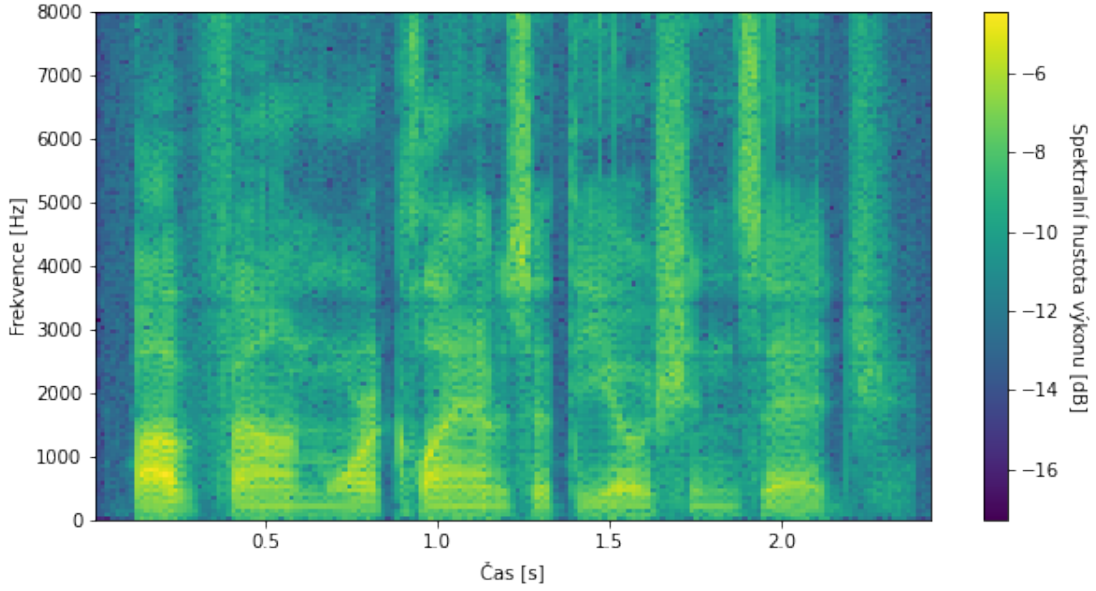
1.6 Generování signálu

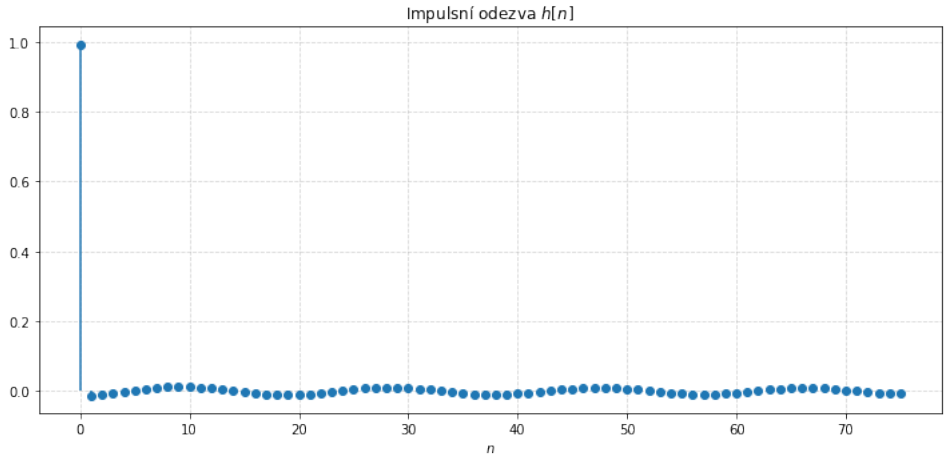
První si vypočítáme čas jak dlouhý signál vlastně bude, tak že každý hodnotu každého vzorku vydělíme frekvencí a uložíme do numpy array. Následně vygenerume cosiny jako cos(2*pi*rušivé frekvence*čas) zase ukládáme do numpy array takže cos i pi je typu numpy array jelikož jsou závislé na čase. Tohle uděláme pro všechny 4. rušivé frekvence pak je dáme dohromady. A zobrazíme na spectogramu. Nakonec signál zapíšeme jako int(16) tím že signál normalizujeme a vynásobíme 2^{15}.

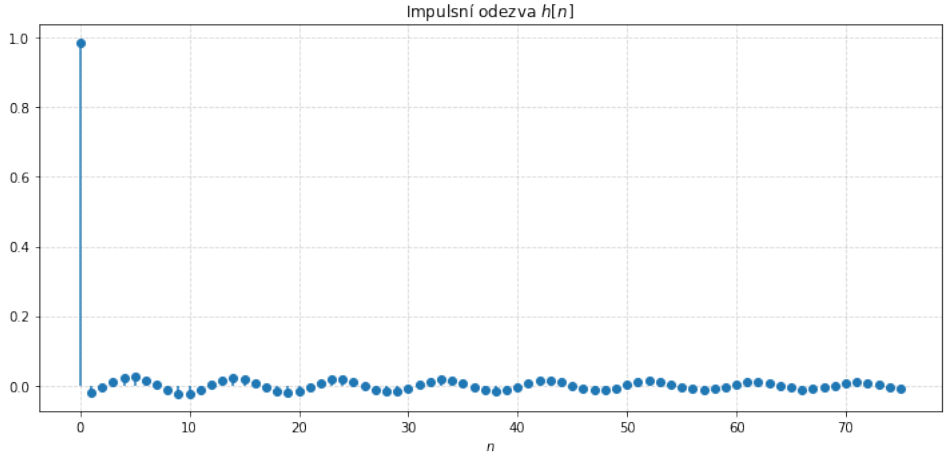


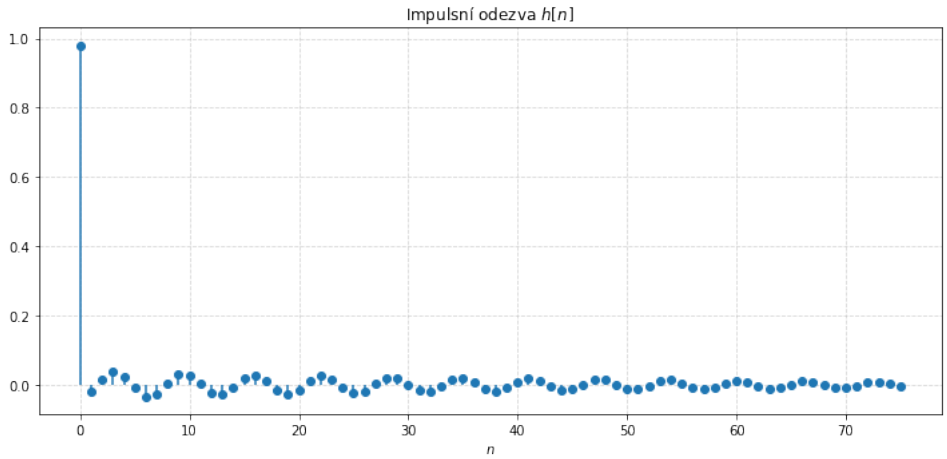
1.7 Čistící filtr

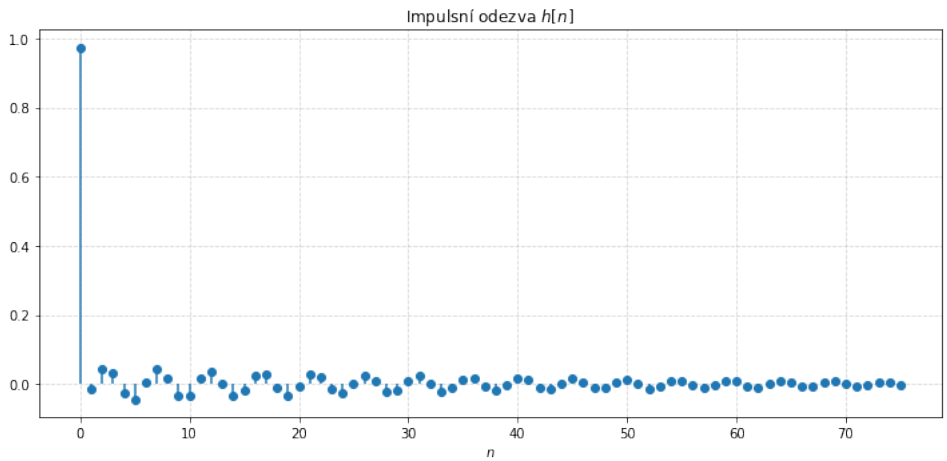
Jako filtr jsem si zvolil možnost číslo 3. návrch pásmových zádrží, který je implementovám pomocí funkce iirnotch z knihovny scipy.signal kdy vlastně pro každou rušivou frekvenci voláme tuhle funkci a následně výsledek uložíme pomococí funkce filtfilt z knihovny spicy.signal, jelikož při poslechu nahrávky je jasně slyšet pípnutí úplně na záčatku a na konci tak jsem vzal prvních a posledních 1000 vzorů a pokud je hodnota vzorku v absolutní hodnotě větší než 0.0001 tak se přepíše na 0 samozřejmně by to nešlo aplikovat na celý signál, jelikož by to přepsalo i zvuk který chceme slyšet. Poté uděláme impuzní odezvy pro každou rušivou frekvenci zvášť které jsou vidět na další straně od f1 po f4. Nakonec zobrazíme spectogram vyfiltrovaného singálu kde jasně vidíme že zmizely rušivé prvky.




Impulsní odezva $h[n]$
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
n

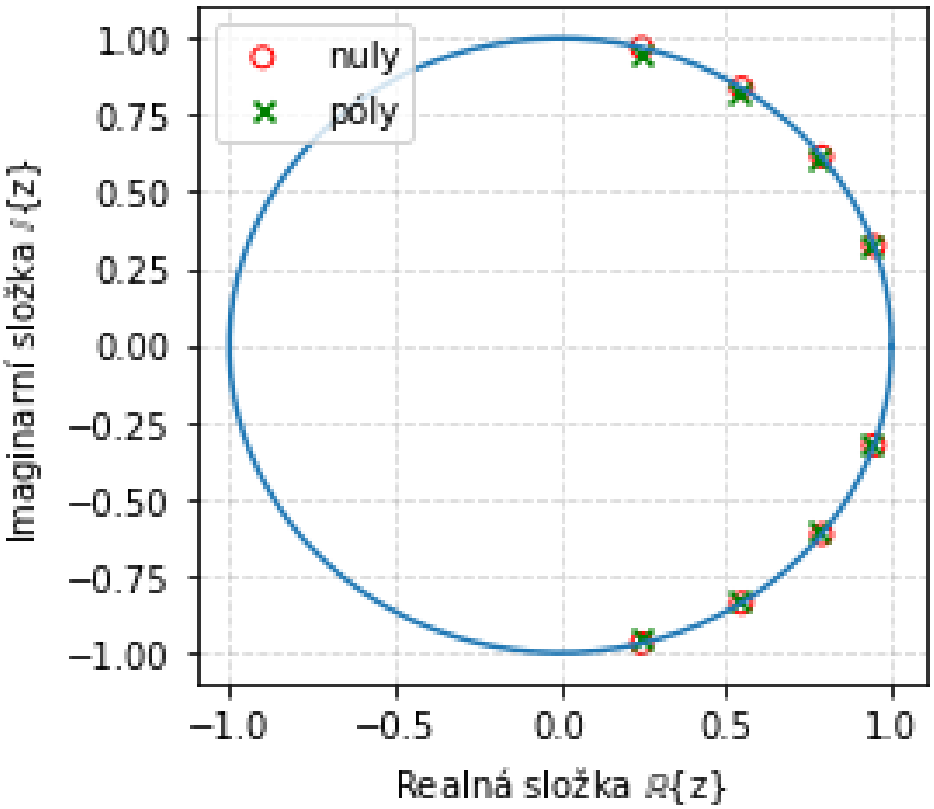

Impulsní odezva $h[n]$
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
n


Impulsní odezva $h[n]$
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
n


Impulsní odezva $h[n]$
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
70
n

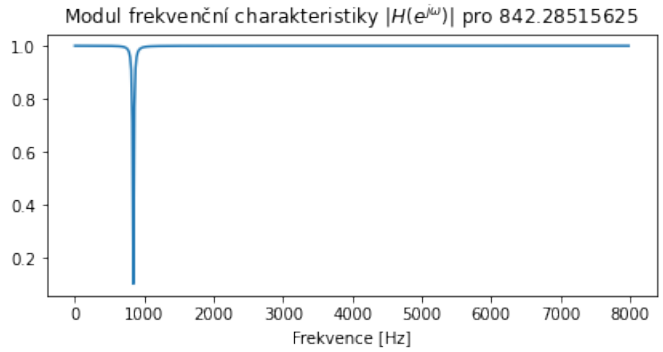
1.8 Nulové body a póly

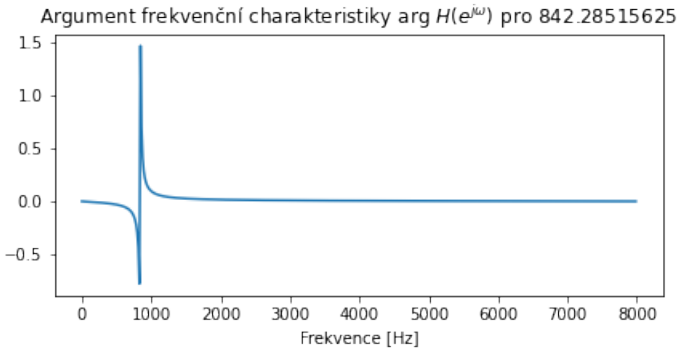
Pomocí funkce tf2zpk z knihovny spicy.signal. Zjistíme jaké máme nulové body a póly funkci voláme pro každou rušivou frekvenci zvlášť. Poté si vygenerujeme jednotkovou kružnici a na kružnici zobrazíme pomocí funkce scatter jednotlivé nulové body a póly.

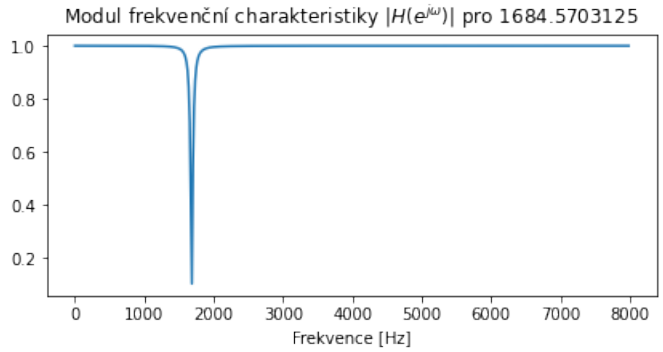


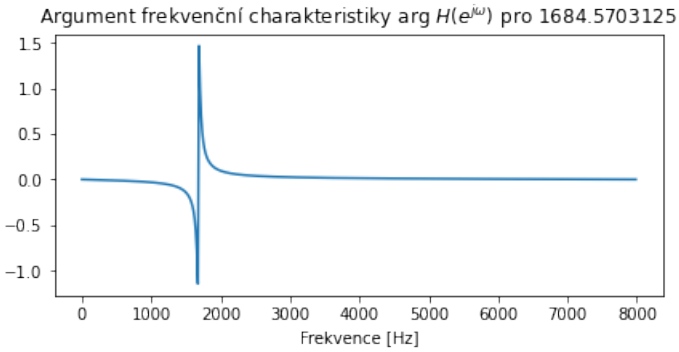
1.9 Frekvenční charakteristika

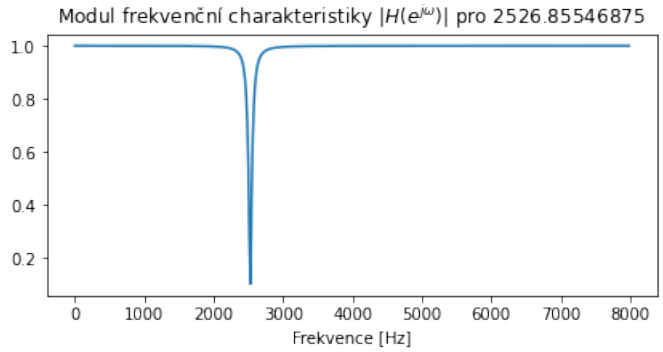
Pomocí funkce freqz si poukládáme jednotlivé filtry a pak je zobrazíme pro jednotlivé rušivé frekvence zvlášť.

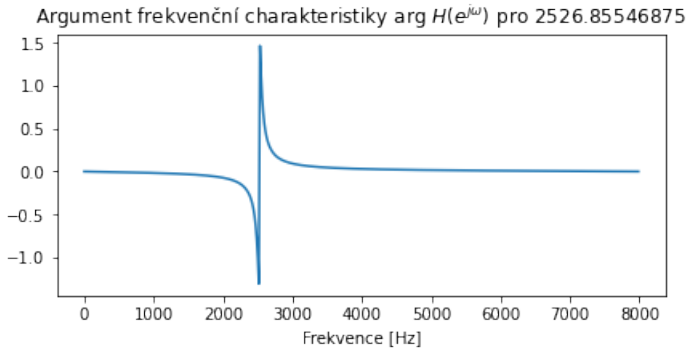


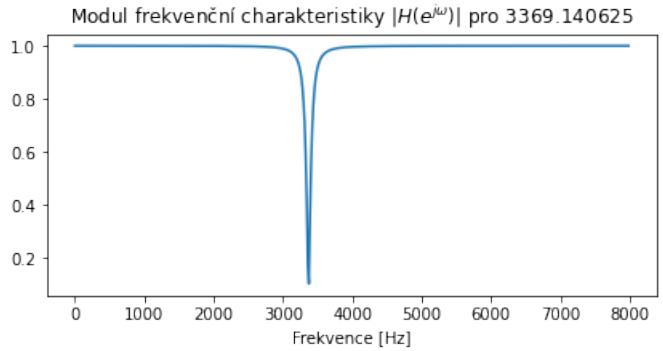


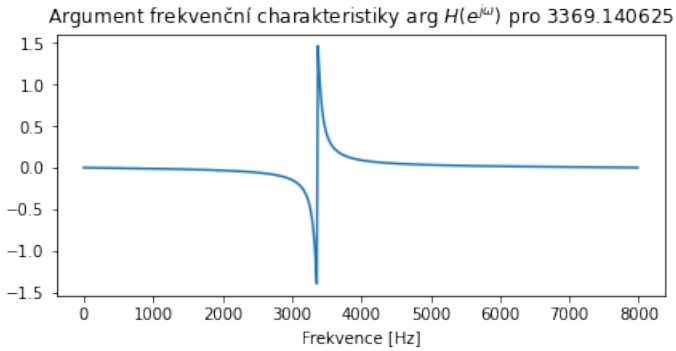












1.10 Filtrace

Singál zase vydělíme maximální absoltutní hodnotou, abychom ho dostali do dynamického rozsahu -1 až 1 poté ho zapíšeme do souboru jako int(16) ještě ho musíme převést do správného rozsahu takže ho vynásobíme maximální hodnotou int rozsahu. Po přehrátí nahrávky jasně vydíme, že zmizely rušivé prvky.