

**VSG/ISS**

**ZS 2021/22**

Tomáš Benda, 200604

Semestrální projekt

## Postup řešení

Program je psaný v jazyce Python (Python 3.9.6). V projektu jsou využity knihovny Numpy, Scipy a Soundfile a Pathlib.

### 4.1

K načtení zvukového souboru byla použita knihovna Soundfile. Výstupem funkce `sf.read()` je pole hodnot signálu a frekvence  $F_s$ .

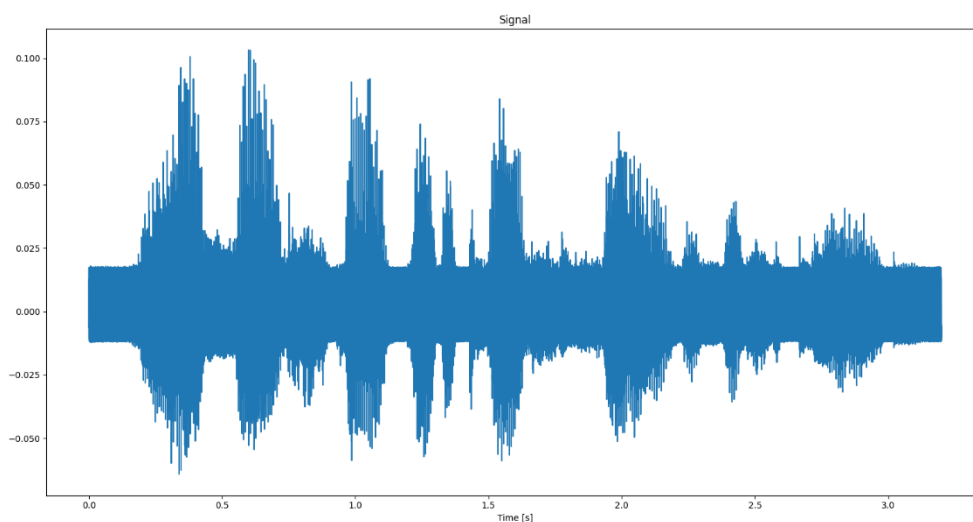
Délka signálu ve vzorcích je pak:  $len\_samples = 51200$  vzorků

Délka signálu v sekundách je:  $len\_seconds = \frac{délka\ pole}{F_s} = 3.2\ s$

Maximální hodnota = 0.1032

Minimální hodnota = -0.0641

Na následujícím grafu je zobrazený signál v celém plném časovém rozsahu 3.2 s.



Obr. 1: Graf hodnot signálu v čase

### 4.2

Odečtení střední hodnoty a normalizace do dynamického rozsahu -1 a 1 byla provedena následující částí kódu:

```
signal = signal-signal.mean()
abs_val = max(abs(signal))
signal = signal/abs_val
```

Podle zadání byla připravena matice pro rámce, kde každý rámeček měl být uložen ve sloupci.

```

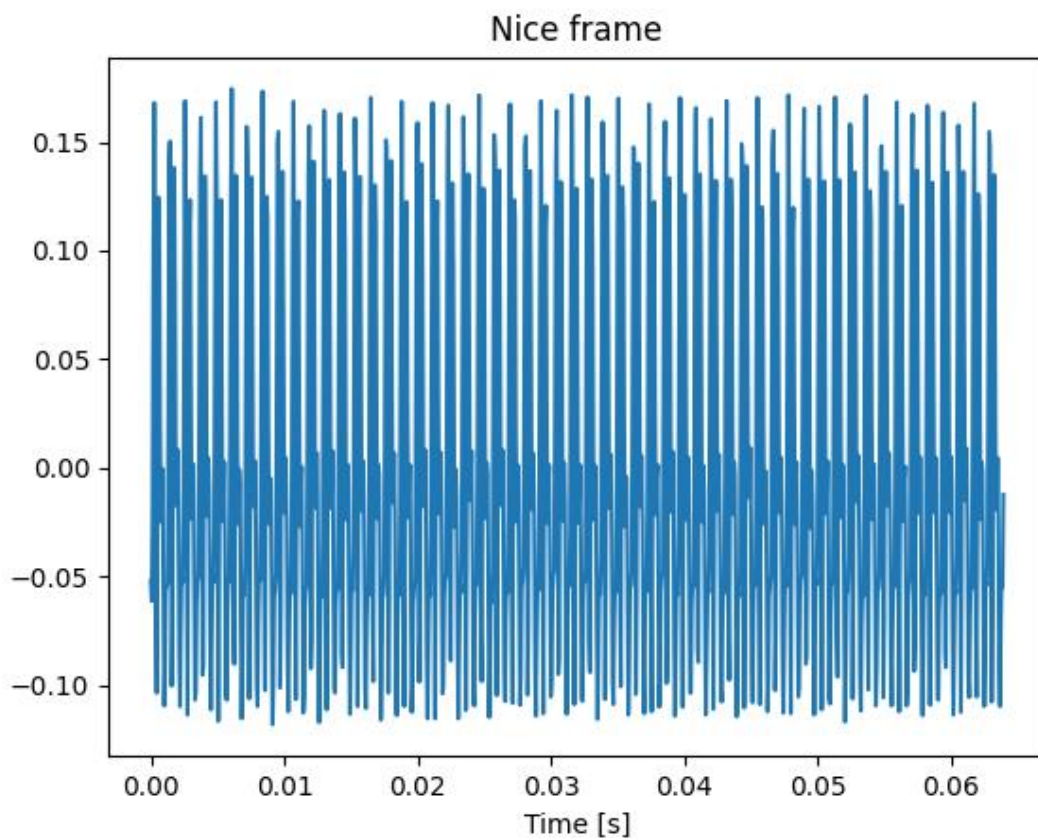
samples = 1024
overlap = 512
cols = int(round(len(signal)/overlap))

signal_div = np.zeros((samples,cols))
for i in range(cols-1):
    signal_div[:,i] = signal[overlap*i:overlap*i+samples]
last_frame = signal[overlap*(cols-1):overlap*(cols-1)+samples]
remaining = len(last_frame)
signal_div[0:remaining,cols-1] = last_frame

```

Proměnná `signal_div` je pak matice s 1024 řádky a 100 sloupci.

Pro zobrazení byl vybrán první rámec, který se zdál být “pěkný”.

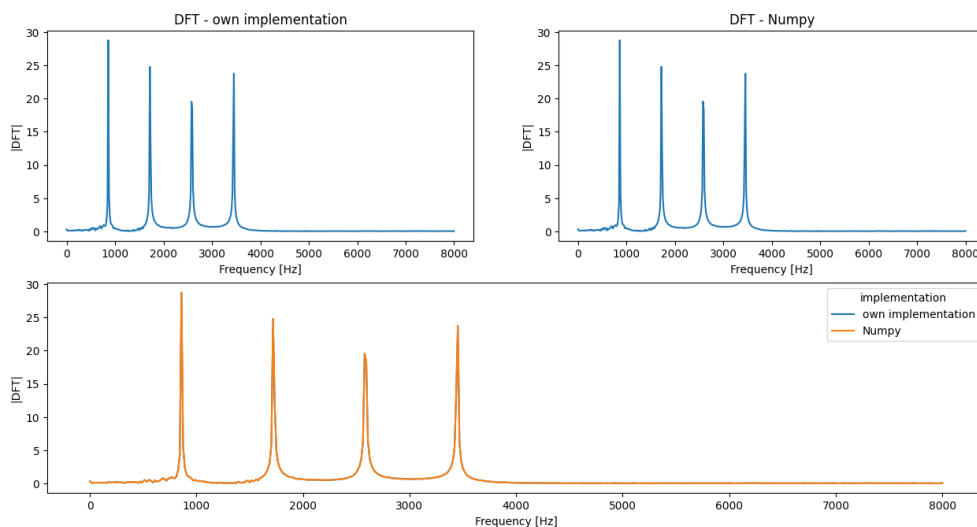


Obr. 2: “Graf “pěkného” rámce

#### 4.3

Výsledkem tohoto úkolu byla funkce pro výpočet DFT a její srovnání s knihovní funkcí, v tomto případě s FFT z knihovny Numpy (`np.fft.fft()`). Funkce byla napsána pomocí zprvu pomocí `for` cyklů,

následně za pomoci internetové nápovědy napsána tak, aby byla “vektorová”. Zobrazený graf ukazuje, že obě funkce vrátí téměř stejné výsledky. Třetí graf na Obr. 3 ukazuje výsledek na jednom grafu – výsledné hodnoty se téměř úplně překrývají.



Obr. 3: Grafy zobrazující podobnost výsledků knihovní a implementované funkce pro DFT

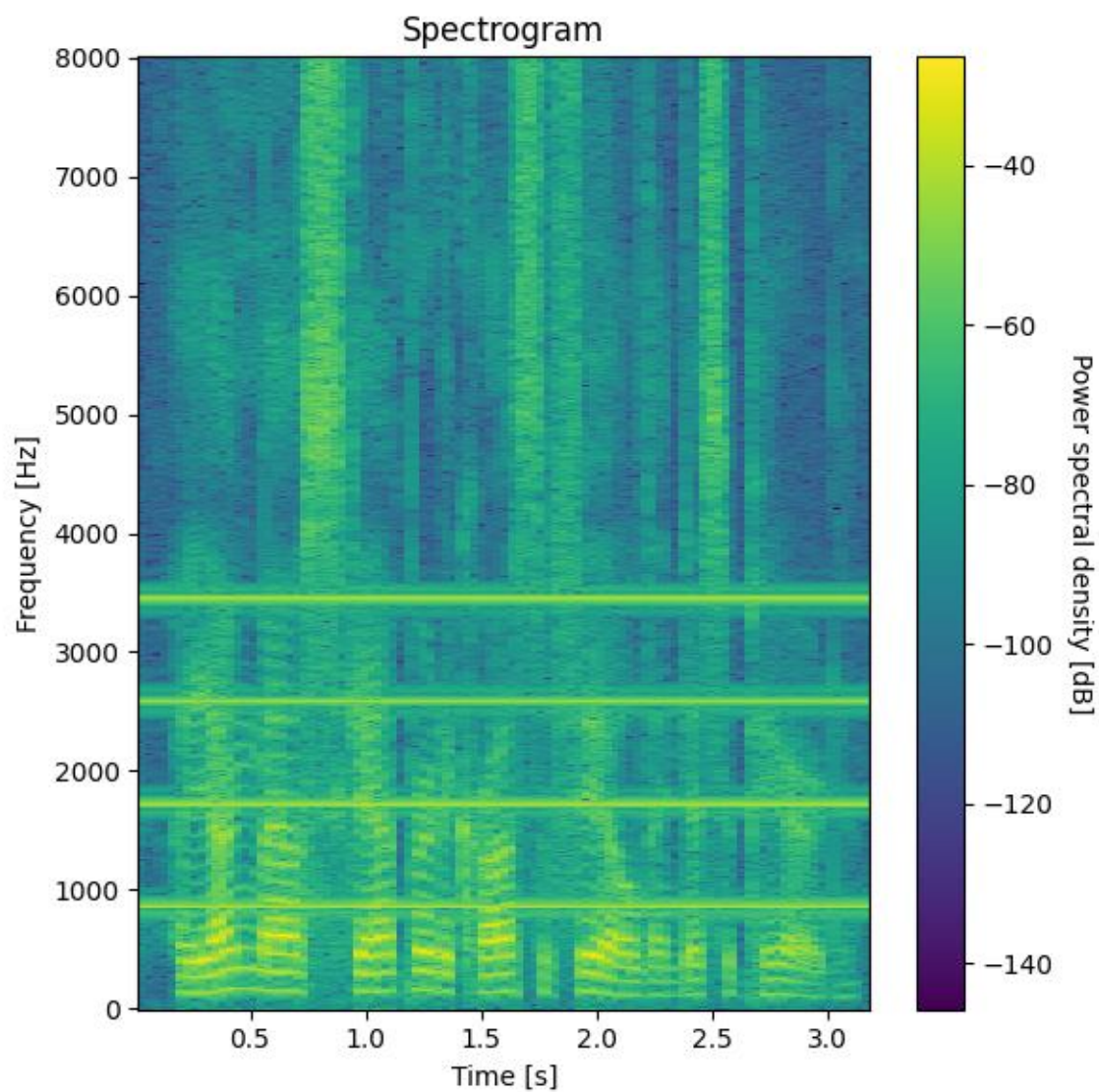
#### 4.4

Pro výpočet logaritmického výkonového spektrogramu byla využita funkce z knihovny Scipy. Jako parametr `nperseg` (počet hodnot na segment) byla zadána hodnota 1024 a jako parametr `noverlap` (překrytí) 512. Hodnoty byly dále upraveny dle zadání.

```
f, t, sgr = spectrogram(x=signal, fs=fs, nperseg=samples, noverlap=overlap)
```

```
sgr_log = 10 * np.log10(sgr)
```

Následně byl zobrazen spektrogram s jasně viditelnými rušivými frekvencemi.



Obr. 4: Spektrogram signálu

4.5

Ze spektrogramu byly “ručně” odečteny rušivé frekvence:

$f_1 = 860 \text{ Hz}$

$f_2 = 1720 \text{ Hz}$

$f_3 = 2580 \text{ Hz}$

$f_4 = 3440 \text{ Hz}$ .

Rušivé kosinusovky jsou harmonicky vztažené.

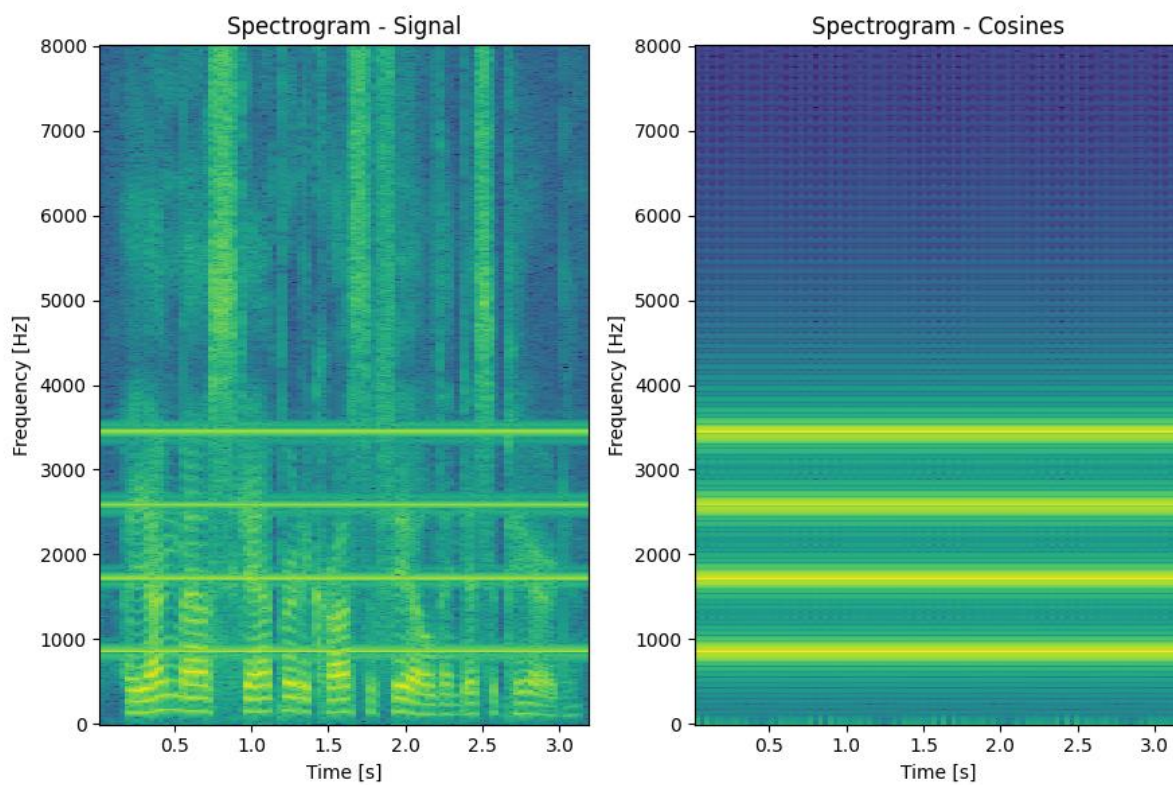
$f_2 = 2 * f_1$

$f_3 = 3 * f_1$

$f_4 = 4 * f_1$

## 4.6

Díky nalezeným frekvencím byl vygenerován signal se směsí 4 kosinusovek. Následující obrázek 5 ukazuje srovnání spektrogramů zvukového signálu vstupního a signálu vygenerovaného.



Obr. 5: Srovnání spektrogramů vstupního signálu a vygenerovaného signálu směsí 4 kosinusovek

Dle poslechu se rovněž zdá, že jsou frekvence určeny správně.

Vygenerovaný signál 4 kosinusovek byl následně uložen do souboru.

## 4.7

Pomocí funkce `scipy.signal.butter()` byly vytvořeny čtyři band-stop filtry pro potlačení nežádoucích frekvencí rušivých kosinusovek.

Koeficienty filtrů:

Filtr 1:

$b = [0.9384318, -8.85592526, 38.12129562, -98.51745701, 169.21293121, -201.79853572, 169.21293121, -98.51745701, 38.12129562, -8.85592526, 0.9384318]$

$a = [1, -9.31701822, 39.5970585, -101.03415583, 171.33905629, -201.75160308, 167.03872864, -96.026231, 36.68981959, -8.41629214, 0.88065425]$

Filtr 2:

$b = [0.9384318, -7.32521914, 27.56386276, -65.00730545, 105.87118467, -124.06663627, 105.87118467, -65.00730545, 27.56386276, -7.32521914, 0.9384318]$

$a = [1, -7.70661429, 28.63064495, -66.66671929, 107.19877961, -124.03447172, 104.5083027, -63.36230555, 26.52857695, -6.96157458, 0.88065425]$

Filtr 3:

$b = [0.9384318, -4.96694146, 15.20779046, -30.99921422, 46.8228768, -53.31188399, 46.8228768, -30.99921422, 15.20779046, -4.96694146, 0.9384318]$

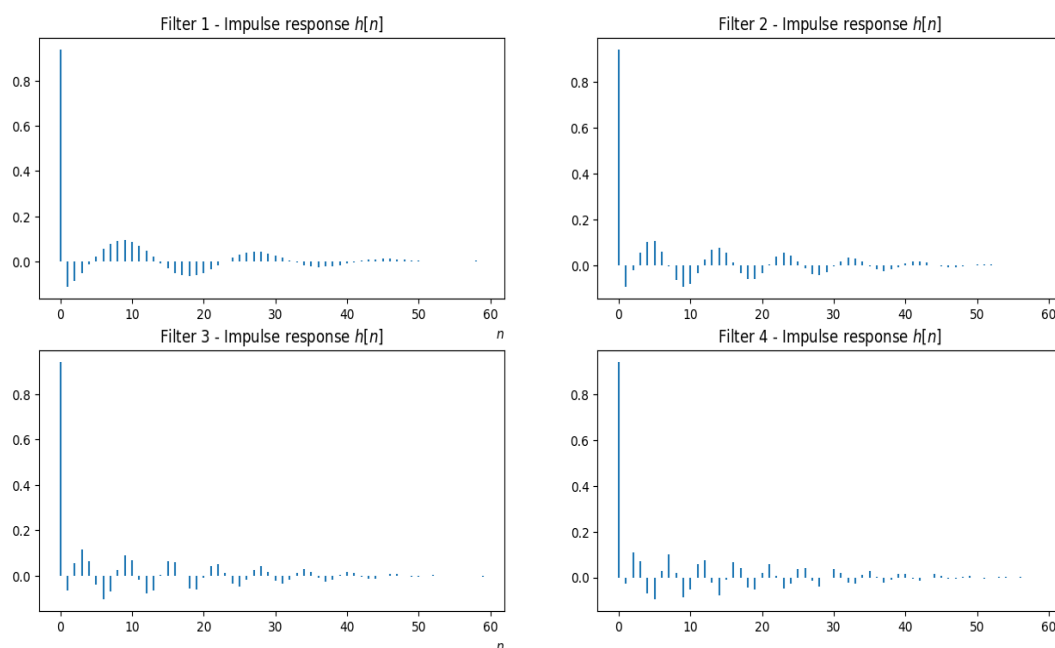
$a = [1, -5.22555043, 15.79591511, -31.78928026, 47.4076206, -53.29535654, 46.21781551, -30.21363964, 14.63619267, -4.72036846, 0.88065425]$

Filtr 4:

$b = [0.9384318, -2.04752023, 6.47911426, -8.96985564, 14.91531915, -13.85951925, 14.91531915, -8.96985564, 6.47911426, -2.04752023, 0.9384318]$

$a = [1, -2.1541265, 6.72910177, -9.19795468, 15.09992058, -13.85424433, 14.72097513, -8.74206994, 6.23507869, -1.94587555, 0.88065425]$

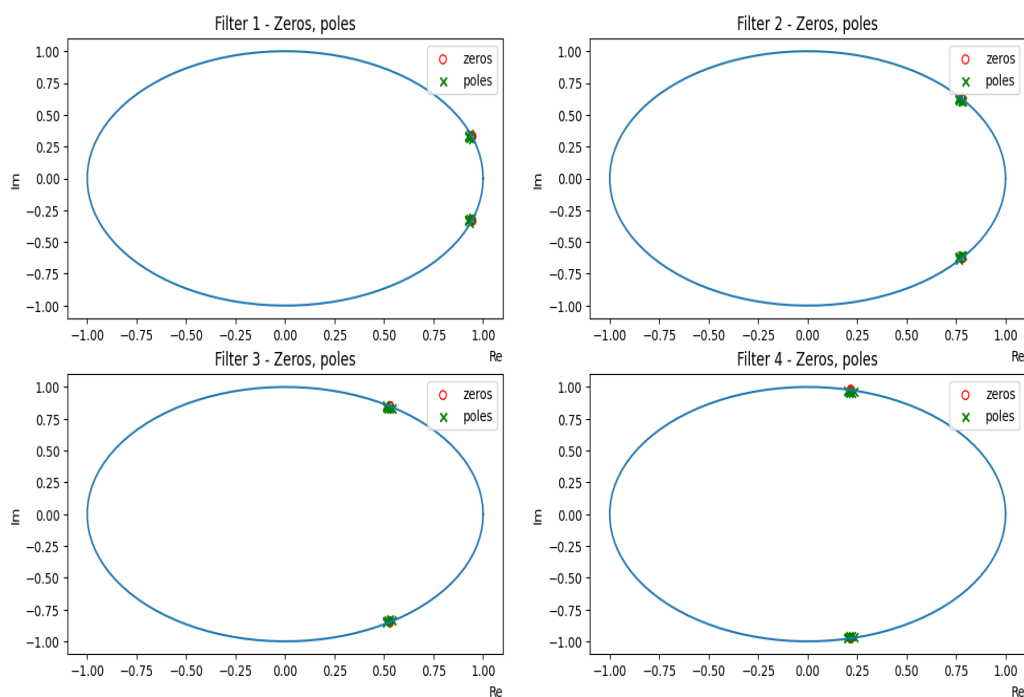
Následující obrázek ukazuje impulsní odezvu filtrů:



Obr. 6: impulsní odezvy filtrů

#### 4.8

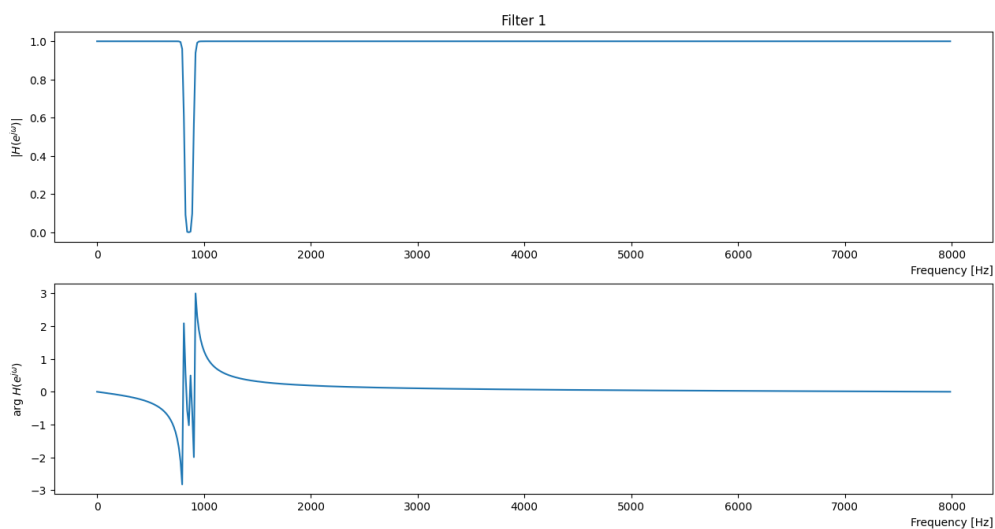
Póly a nuly byly určeny pomocí funkce `scipy.signal.tf2zpk()` a následně zobrazeny v komplexní rovině se zobrazenou jednotkovou kružnicí pro ukázání stability filtru – všechny póly jsou uvnitř jednotkové kružnice a tudíž jsou filtry stabilní.



Obr. 7: Zobrazení pólů a nul v komplexní rovině pro všechny filtry

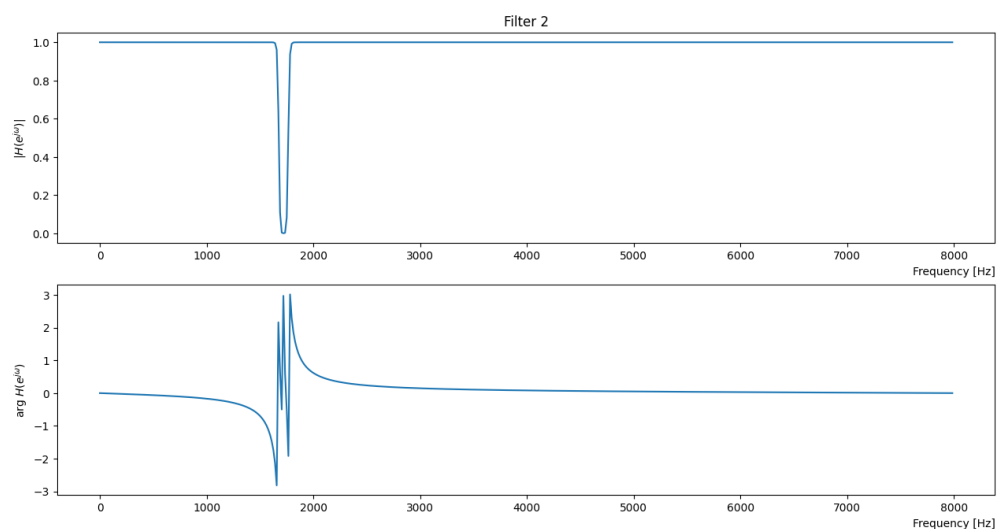
#### 4.9

Frekvenční charakteristika byla spočítána funkcí `scipy.signal.freqz()`. Grafy jsou zobrazeny níže.

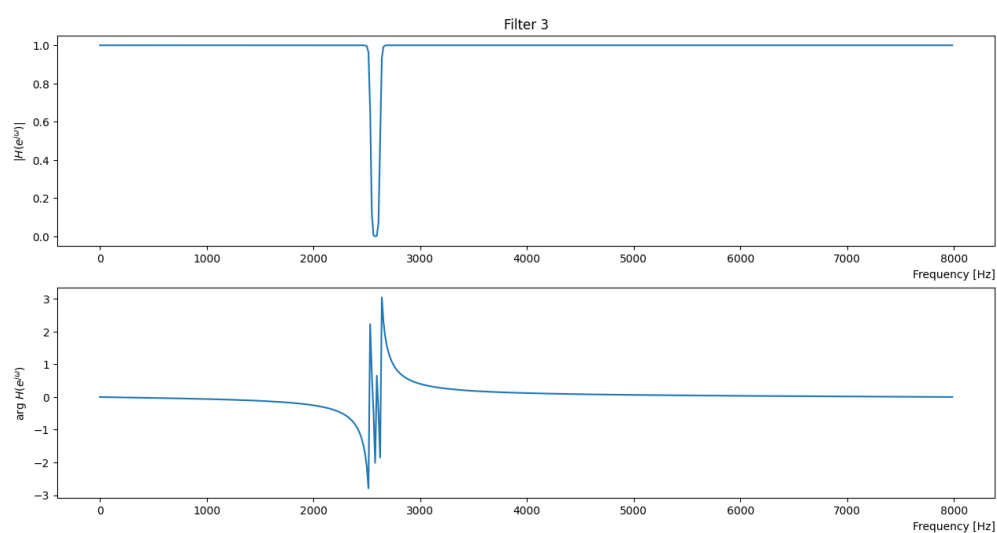


Obr. 8: Frekvenční charakteristika filtru 1

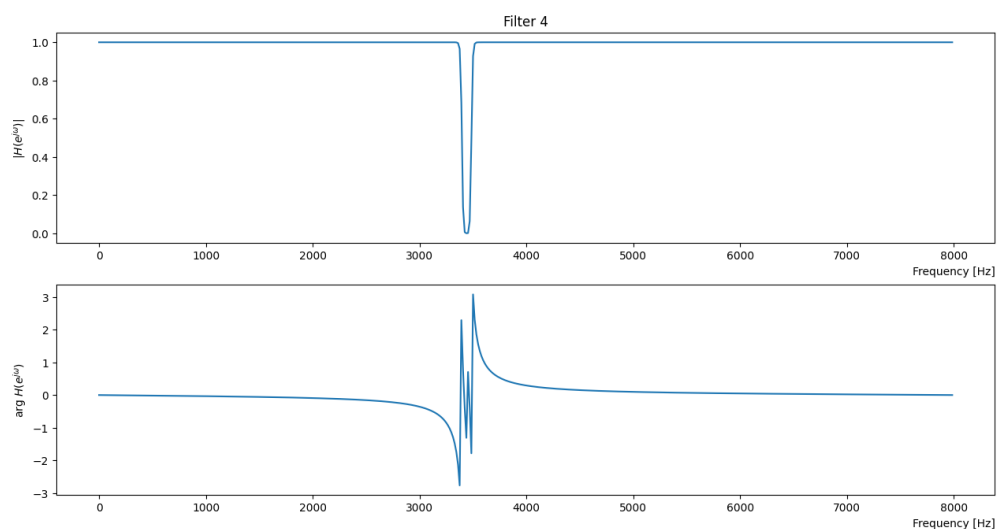




Obr. 9: Frekvenční charakteristika filtru 2



Obr. 10: Frekvenční charakteristika filtru 3



Obr. 11: Frekvenční charakteristika filtru 4

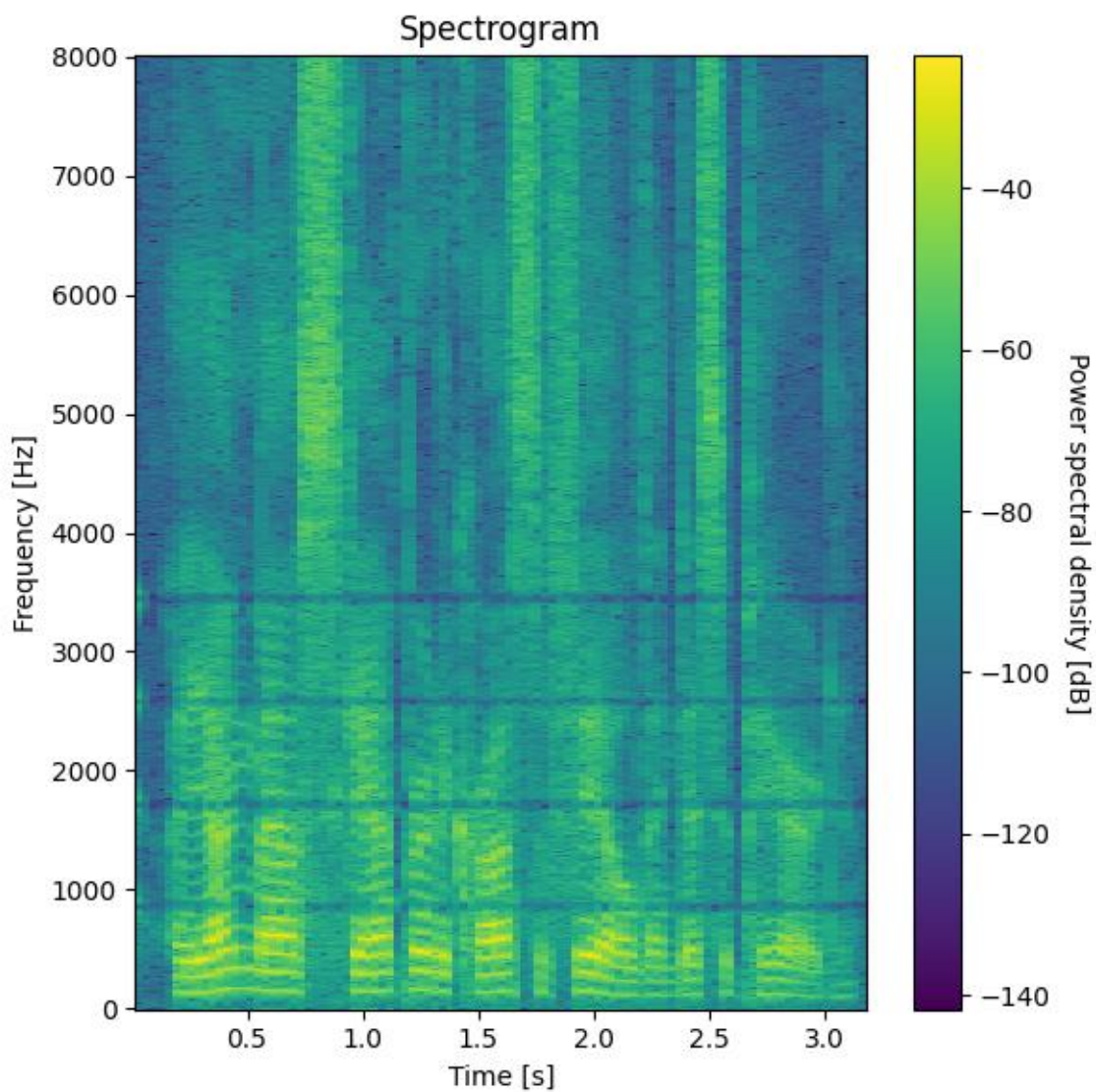
Z grafů je patrné, že filtry opravdu potlačují frekvence v okolí rušivých frekvencí.

#### 4.10

Signál byl vyfiltrován pomocí 4 navržených filtrů typu pásmová zádrž. Následně byl ještě normalizován na dynamický rozsah -1 a 1 pomocí dělení maximální absolutní hodnotou signálu. Signál byl uložen do souboru. Dle poslechu souboru došlo k vyfiltrování rušivých kosinusovek na zjištěných frekvencích. Při poslechu signálu je slyšet pouze mluvené slovo původního signálu bez rušení díky vyfiltrování frekvencí.

## Závěr

V rámci projektu byly navrženy 4 filtry pásmová zádrž, které dle poslechu a grafů, jež byly prezentovány, vyfiltrovaly nežádoucí kosinusovky. Při poslechu výsledného signálu již není zvuk 4 vložených harmonicky vztažených kosinusovek.



Obr. 12: Výsledný spektrogram vyfiltrovaného signálu

Na Obr. 12 je vidět spektrogram vyfiltrovaného signálu. Při porovnání s Obr. 4 je vidět, že rušivé kosinusovky byly vyfiltrovány. Frekvenční charakteristiky filtrů pak ukazují, že opravdu filtry působí na frekvencích kosinusovek zjištěných při analýze signálu.