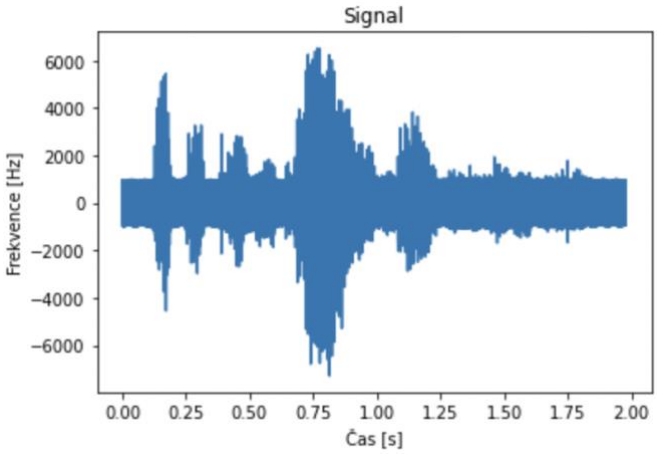
Protokol

Marko Kubrachenko (xkubra00)
Prosinec 26, 2021

Úloha č.1

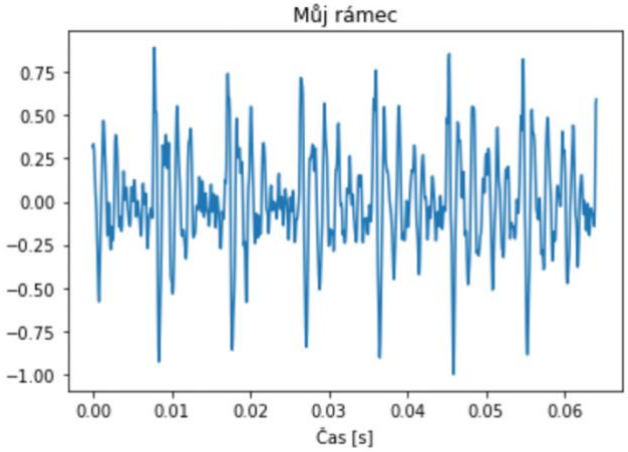
Načítal jsem signál pomocí wavfile.read, na určení maximální a minimální hodnoty jsem použil funkce max() a min(). Na zjištění délky signálu ve vzorcích jsem použil funkci size(), délku signálu v sekundách jsem spočítal dělením délky signálu ve vzorcích vzorkovací frekvencí.

Délka signálu ve vzorcích je 31642
Délka signálu v sekundách je 1.977625 s
Maximální hodnota signálu je 6521 Hz
Minimální hodnota signálu je -7289 Hz



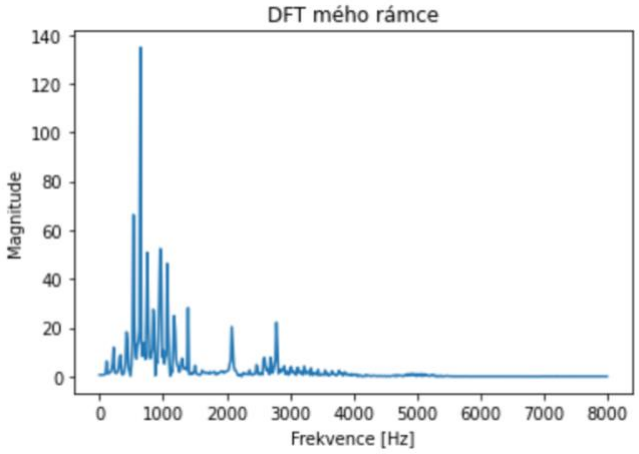
Úloha č.2

Předzpracování signálu jsem dělal podle návodu v zadání. Na ukládání rámců jsem vytvořil matici pomocí funkce np.empty, naplnění matice jsem provedl ve dvou for cyklech, kde jsem pokaždé přidával 512 k levé a pravé hranici signálu, abych dostal překrytí 512 vzorků, a postupně zapisoval vzorce.



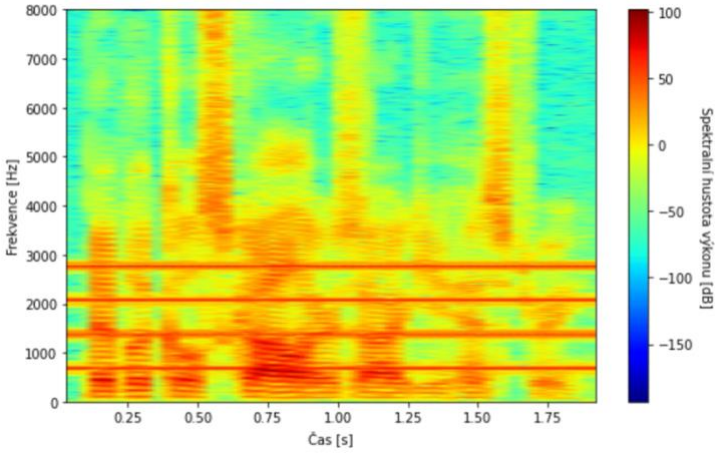
Úloha č.3

Funkci DFT jsem implementoval jako násobení matice bázi s vektorem mého rámce pomocí funkce np.dot. Funkce má dva parametry: pole vzorků signálu a počet vzorků signálu. Funkce vrací pole koeficientů DFT. Výsledek jsem dal do absolutní hodnoty a vykreslil jsem prvních 512 koeficientů.



Úloha č.4

Pro výpočet a zobrazení spektrogramu signálu (rámce o délce 1024 vzorků s překrytím 512 vzorků) jsem využil knihovní funkci spectrogram se správně nastavenými parametry: nperseg = 1024, noverlap = 512. Výsledek jsem upravil pomocí vzorce ze zadání, a přidal jsem k výsledku hodně malou hodnotu, protože funkce spectrogram může vratit i nulu, která se nebude líbit logaritmu.

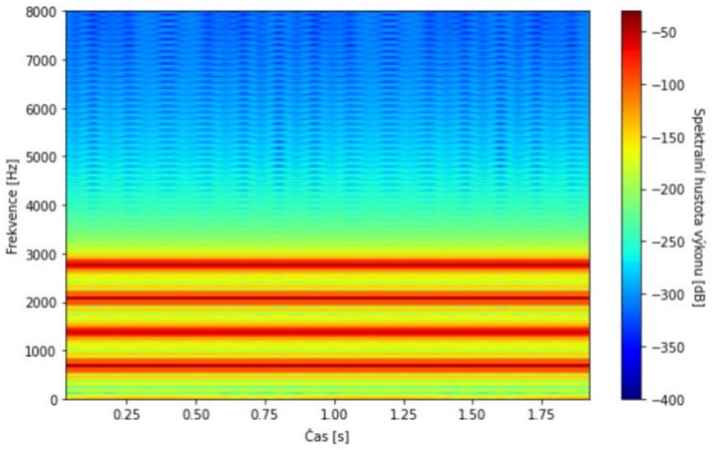


Úloha č.5

Rušivé frekvence jsem určil tak, že jsem si vzal první rámec bez řeči, spočítal jsem DFT pomocí funkce z úlohy č.3, seřadil jsem výsledek v opačném pořadí (od největšího po nejmenší) a vybral jsem z toho první 4 hodnoty: 687.5, 1390.625, 2078.125, 2781.25. Je vidět, že všechny frekvence jsou násobky té nejmenší, ale nemáme dostatečně dat, proto výsledky nejsou úplně přesné.

Úloha č.6

Signál směsi 4 cosinusovek jsem dostal sčítáním 4 cosinusovek, které jsem vygeneroval pomocí funkce np.cos, frekvenci, určených v minulé úloze, a pole časových hodnot, které jsem vytvořil tak, že jsem spočítal kolik je čas mezi vzorky v původním signál (vydělil jsem délku v sekundách délkou ve vzorcích původního signálu) a využil jsem funkci np.arange pro naplnění pole. Zápis směsí cosinusovek jsem provedl pomocí wavfile.write.

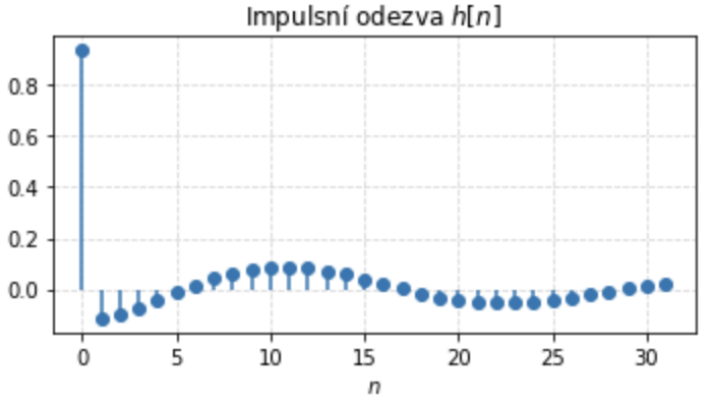


Úloha č.7

Pro implementaci filtru jsem využil způsob návrhu 4 pásmových zádrží. Stačilo na to použít funkce signal.butter a signal.buttord. Nastavil jsem závěrné pásmo na 30 Hz kolem rušivých frekvencí a přechod 50 Hz z obou stran pásma, zvlnění v propustném pásmu na 3 dB a potlačení v závěrném pásmu na -40 dB.

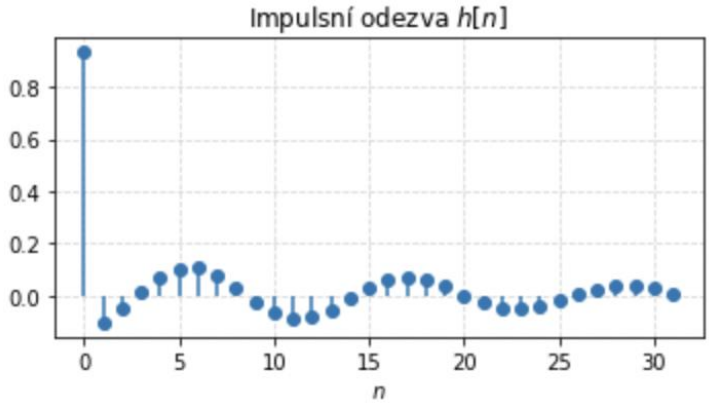
```
Coefficients A of filter 1:  [  1.          -7.58887952  25.47018087 -49.3627974   60.41137554
 -47.80388211  23.88686159  -6.89237594   0.87954196]
Coefficients B of filter 1:  [  0.93783899  -7.23317869  24.67134116 -48.5907888   60.42959967
 -48.5907888   24.67134116  -7.23317869   0.93783899]
```

Impulse response of filter 1:



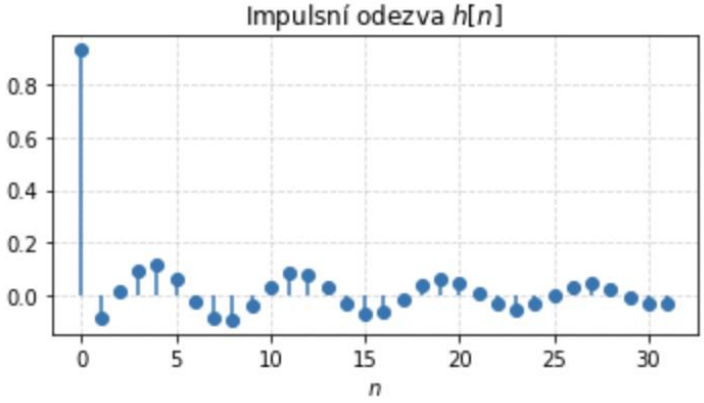
```
Coefficients A of filter 2:  [  1.          -6.72581049  20.83355029 -38.53767276  46.43680955
 -37.28940041  19.50577401  -6.09319174   0.8766009 ]
Coefficients B of filter 2:  [  0.93626967  -6.4025588   20.1637294  -37.9204789   46.45273661
 -37.9204789   20.1637294   -6.4025588    0.93626967]
```

Impulse response of filter 2:



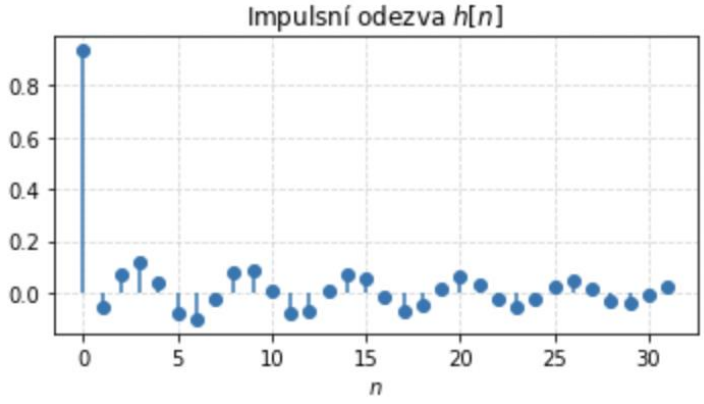

```
Coefficients A of filter 3:  [  1.          -5.38723013  14.75124552 -25.40137255  29.95365082
 -24.56987902  13.80130557  -4.87530957   0.87535436]
Coefficients B of filter 3:  [  0.93560374  -5.12559139  14.27238853 -24.99130425  29.96557174
 -24.99130425  14.27238853  -5.12559139   0.93560374]
```

Impulse response of filter 3:



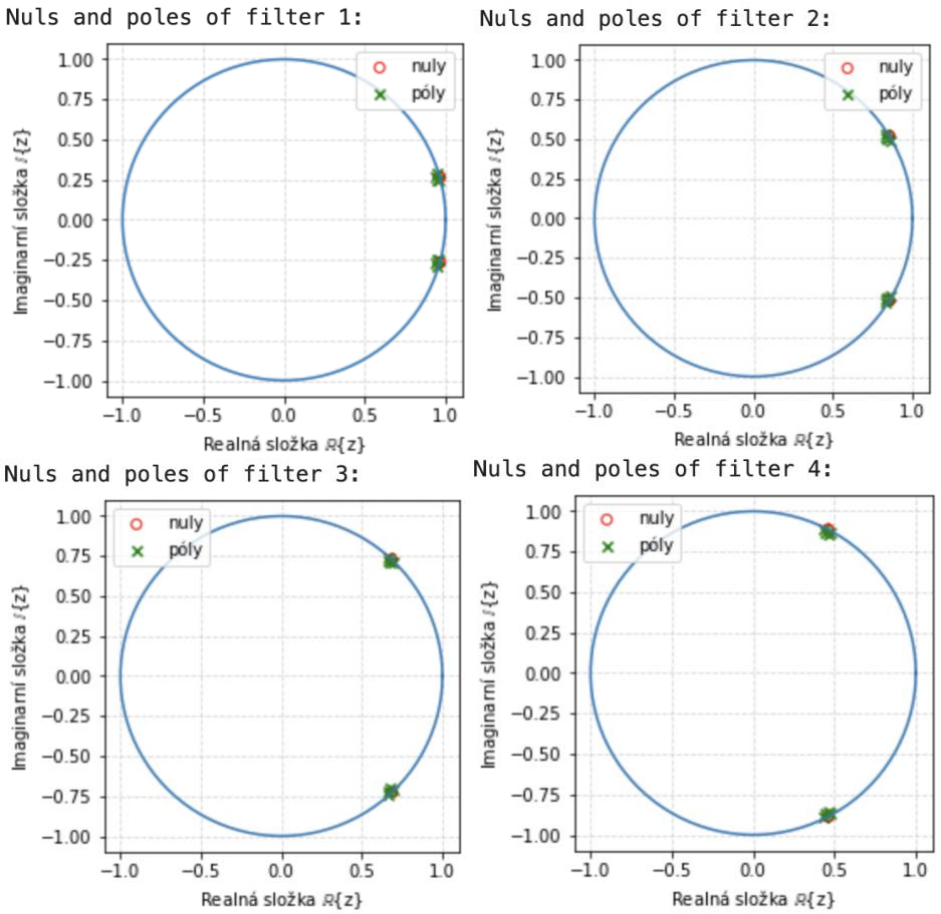

```
Coefficients A of filter 4:  [  1.          -3.63140971   8.81449531 -13.53281967  15.86526976
 -13.09633292   8.25505929  -3.29121908   0.87708898]
Coefficients B of filter 4:  [  0.93653029  -3.45759694   8.53306321 -13.31829375  15.87272632
 -13.31829375   8.53306321  -3.45759694   0.93653029]
```

Impulse response of filter 4:



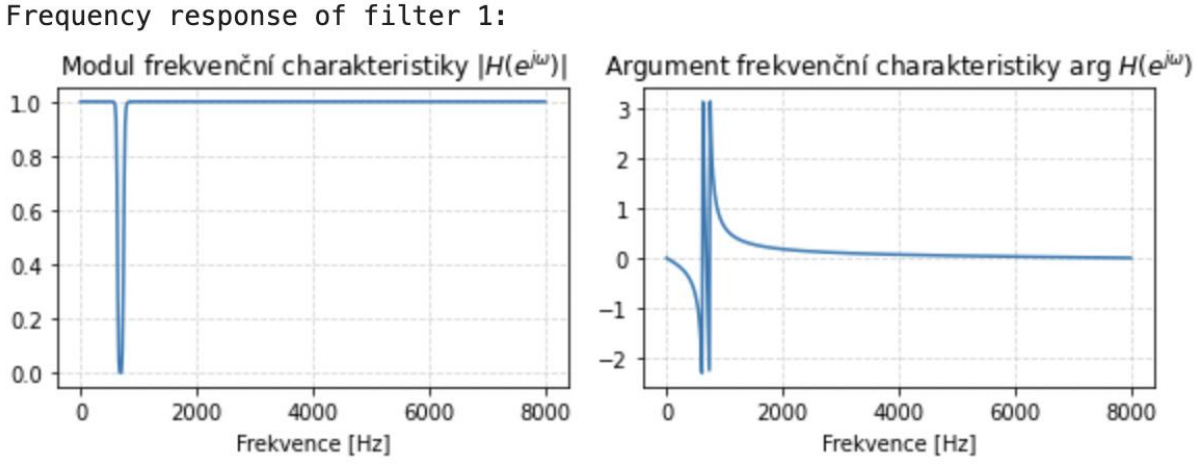
Úloha č.8

Nulové body a póly filtrů jsem dostal převodem jejich koeficientů na nulové body a póly pomocí funkce signal.tf2zpk. Všechny póly se nachází uvnitř jednotkové kružnice, znamená to, že filtry jsou stabilní.

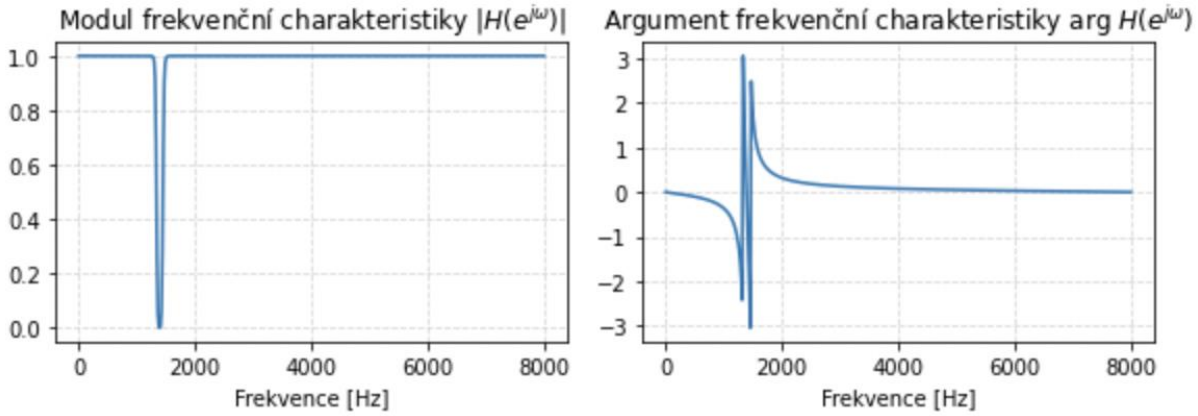


Úloha č.9

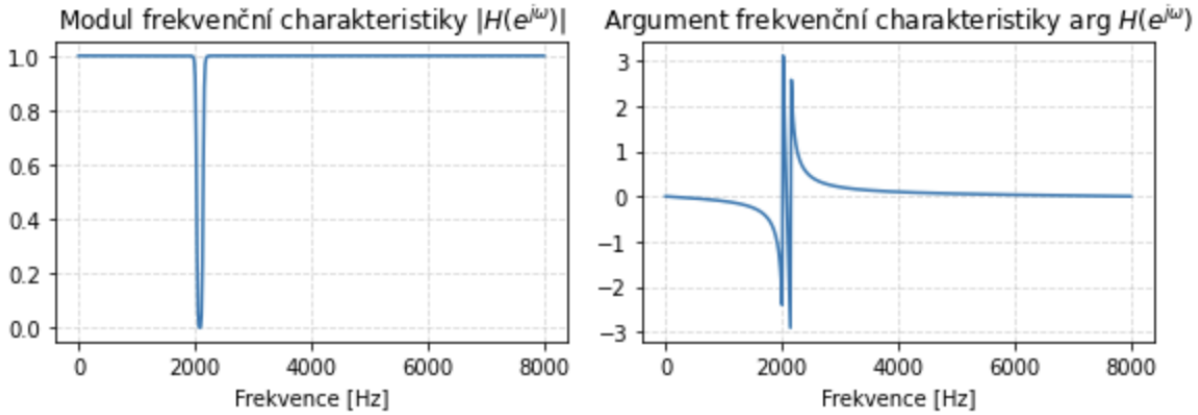
Frekvenční charakteristiku filtrů jsem vypočítal pomocí funkce signal.freqz a koeficientů filtrů. Je vidět, že navřené filtry opravdu dělají to, co chceme.



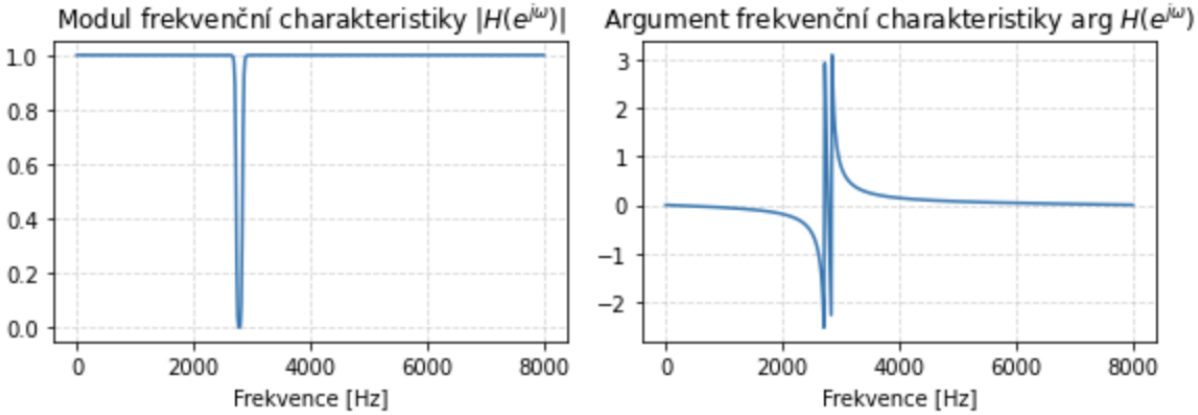
Frequency response of filter 2:



Frequency response of filter 3:



Frequency response of filter 4:



Úloha č.10

Filtrací jsem prováděl postupně s využitím funkce signal.filtfilt. První jsem odstranil nejnižší rušivou frekvenci (687.5 Hz), pak ji dvojnásobek (1390.625 Hz), trojnásobek (2078.125 Hz) a konečně poslední (2781.25 Hz). Normalizoval jsem signál stejně jak v úloze č.2 a uložil jsem výsledek pomocí funkce wavfile.write. Po poslechnutí jsem zjistil, že čištění signálu dopadlo velmi dobře.

Využité zdroje: stránky Ing. Kateřiny Žmolíkové, dokumentace numpy, scipy, matplotlib.