

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

Signály a systémy

2021/2022

Projekt – filtrace signálu

Obsah

Úloha 1.	3
Úloha 2.	3
Úloha 3.	4
Úloha 4.	5
Úloha 5.	6
Úloha 6.	6
Úloha 7.	7
Úloha 8.	9
Úloha 9.	9
Úloha 10.	10

Úloha 1.

Dĺžka signálu vo vzorkách: 49562

Dĺžka signálu v sekundách [s]: 3.097625

Minimálna hodnota signálu: -0.088714599609375

Maximálna hodnota signálu: 0.071563720703125

Vykreslenie grafu:

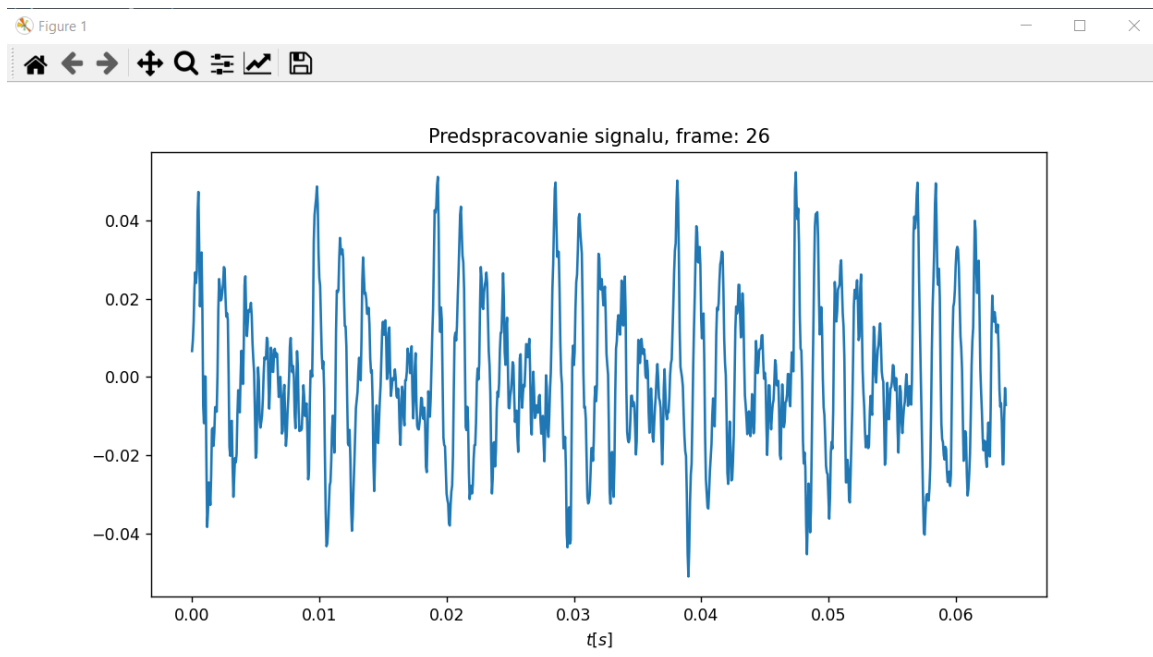


Obr. 1.1. – načítanie pôvodného signálu

Poznámka: V prvej úlohe som vstupný signál načítal pomocou knižnice „soundfile“ a funkcie *sf.read()*, ktorá načítaný signál automaticky normalizovala ale neustrednila.

Úloha 2.

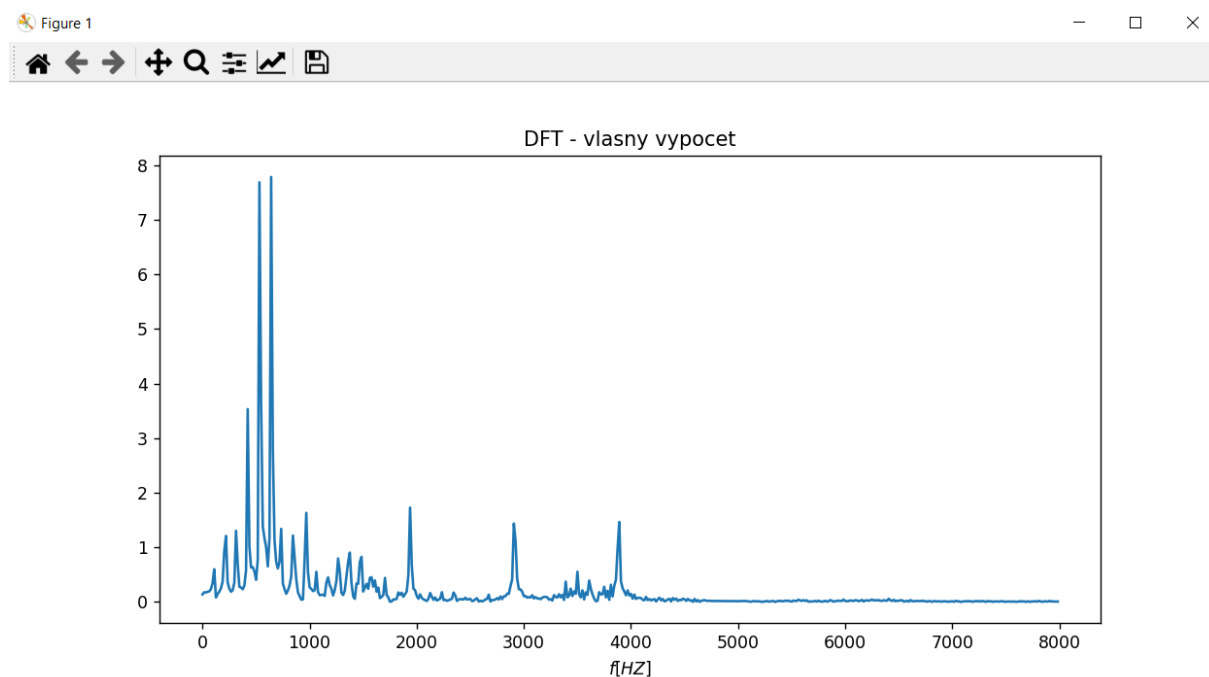
Pri druhej úlohe som signál načítal nanovo kvôli potrebám normalizácie a ustrednenia. Signál som načítal pomocou funkcie *wavfile.read()* z knižnice *scipy.io*, normalizoval a ustrednil na základe zadania úlohy. Pri normalizovaní som použil hodnotu 2^{15} , ktorú mala v pomôckach pani Katka Žmolíková. Tuto hodnotu v pomôcke používala pre normalizáciu signálu.



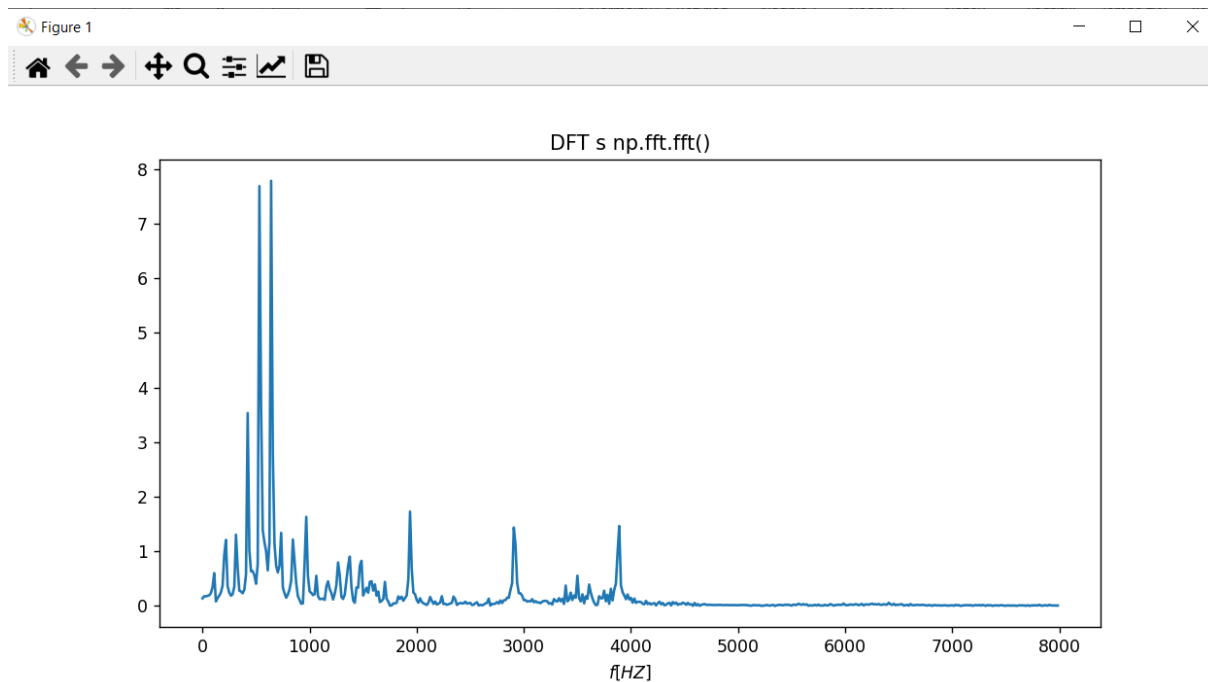
Obr. 2.1. – Vyplotenie zvoleného znelého rámca

Úloha 3.

DFT som implementoval na môj zvolený rámec, teda na rámec číslo 26. Pre aplikáciu DFT som si vytvoril funkcie *dft(data)*, ktorá brala ako parameter premennú “data”, v ktorej sa nachádzal načítaný signal. Následne som si do premennej *selectedFrame* uložil zvolený frame číslo 26 a vykonal DFT nad týmto framom. Výsledok som uložil do premennej *DftArray*. Túto premennú funkcia vrátila.



Obr. 3.1. – DFT vlastným výpočtom pre zvolený rámec



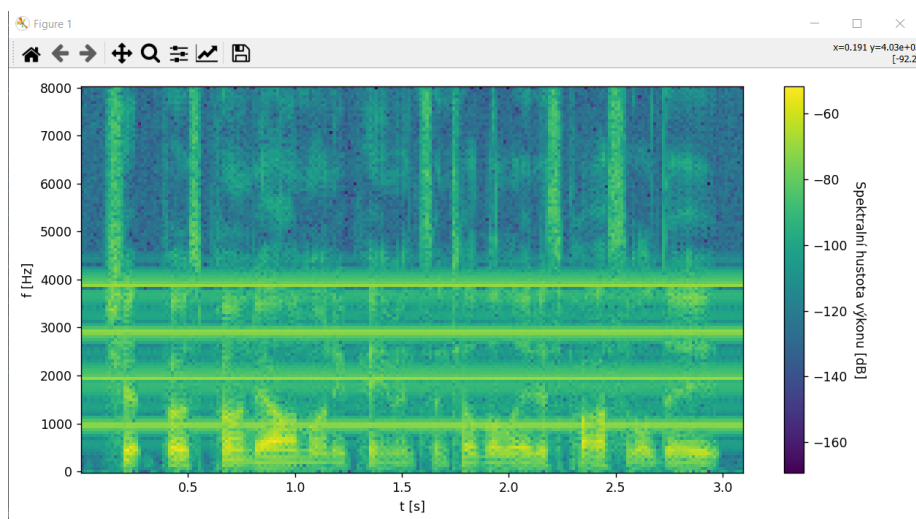
Obr. 3.2. – DFT vytvorené na základe vstavanej funkcie `np.fft.fft()`

Pri porovaní grafov, ktoré som vygeneroval je vidieť, že moja vlastná DFT prebehla úspešne, pretože sa na prvý pohľad skoro 100% zhoduje v DFT, ktoré vytvorila vstavaná funkcia `np.fft.fft()`.

Úloha 4.

Pri tvorbe spektrogramu som sa inšpiroval nápodovou od pani Žmolikovej a spektrogram vytvoril na základe jej nápovedy. Link na použitú nápovedu:

https://nbviewer.org/github/zmolikova/ISS_project_study_phase/blob/master/Zvuk_spektra_filtrace.ipynb



Obr. 4.1. – Zobrazenie spektrogramu celého signálu

Úloha 5.

Na základe spektrogramu u vybraného rámca číslo 26 je vidieť, že sa rušivé elementy (cosínusovky) nachádzajú na frekvenciách $f_1 = 972$, $f_2 = 1924$, $f_3 = 2910$, $f_4 = 3885$. Odčítanie som realizoval na vyplotenom grafe DFT, ktorý mi sám ukázal frekvenciu, na ktorej sa nachádzali rušivé cosínusovky. Na základe týchto hodnôt môžeme vidieť, že tieto rušivé cosínusovky sú harmonicky vzťahované a teda že f_2 , f_3 , f_4 sú k -násobok f_1 .

Úloha 6.

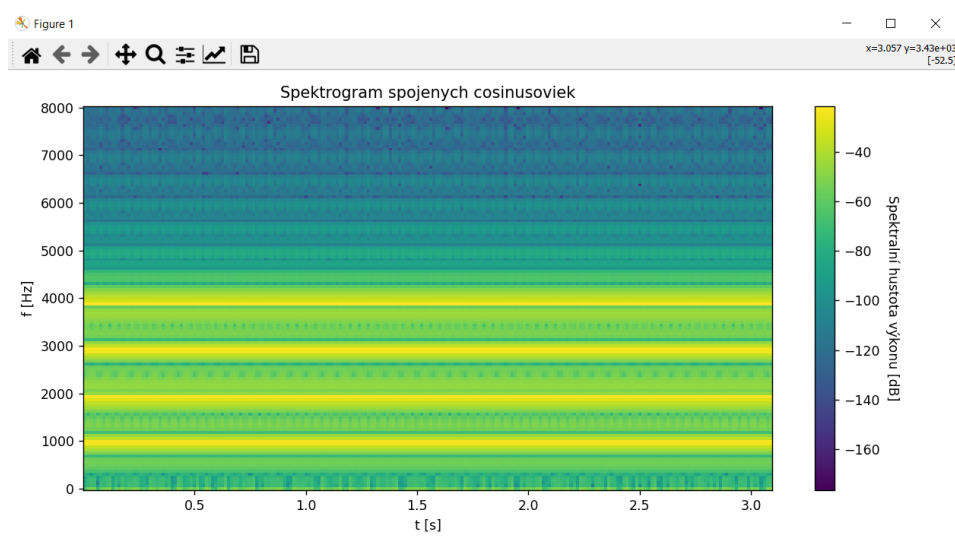
Cosínusovky som vygeneroval pomocou funkcie `cos()` z knižnice `numpy`. Následne som si vytvoril pole, v ktorom bol uložený čas, ktorý sa používal pri tvorbe cosínusoviek. Pri tvorbe cosínusoviek som použil hodnoty frekvencií tie, ktoré som nameral z grafov a uviedol aj v protokole v úlohe 5. Na záver som cosínusovky spojil do jedného signálu a z neho pomocou funkcie `write` z knižnice `scipy.io` vygeneroval požadovaný `4cos.wav` súbor.

```
f1 = 972
f2 = 1924
f3 = 2910
f4 = 3885

N = data.size
time = np.arange(N)/fs
y1 = np.cos(2*np.pi*f1 * time)
y2 = np.cos(2*np.pi*f2 * time)
y3 = np.cos(2*np.pi*f3 * time)
y4 = np.cos(2*np.pi*f4 * time)
Y = y1+y2+y3+y4
wavfile.write("4cos.wav", fs, Y.astype(np.float32))
```

Obr. 6.1. – Zdrojový kód pre tvorbu rušivých cosínusoviek

Zobrazenie spektrogramu spojených cosínusoviek:



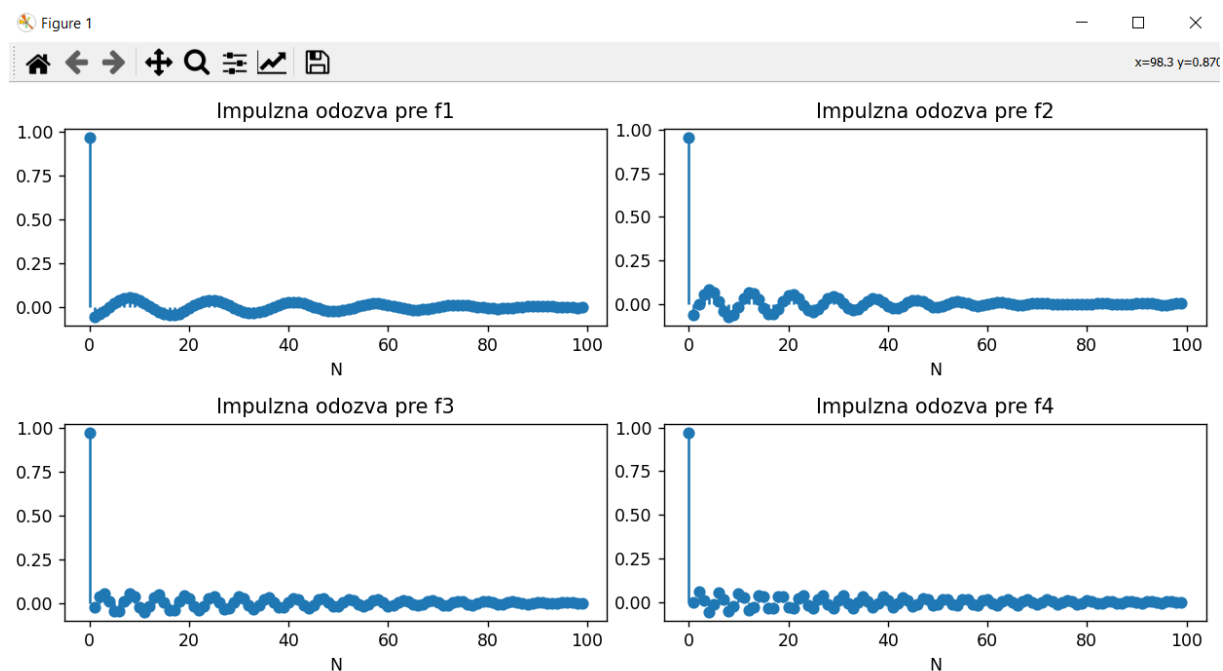
Obr. 6.2. – Spektrogram spojených rušivých cosínusoviek

Pri porovnaní grafov pôvodného spektrogramu a spektrogramu spojených cosinusoviek je vidieť koreláciu medzi danými frekvenciami. Táto korelácia nám ukazuje, že sme správne určili rušivé cosínusovky.

Úloha 7.

Pri tvorbe číslicového filteru som si zvolil tretí spôsob a využil funkciu *scipy.signal.butter*, do ktorej som ako parametre složil potrebné hodnoty. Vytvoril som týmto spôsobom 4 filtre a teda 4 koreficienty filterov a 4 impulzné odozvy, jedna impulzná odozva pre jeden číslicový filter.

Zobrazenie impulzných odoziev:



Obr. 7.1. – Zobrazenie impulzných odoziev pre vytvorené číslicové filtre

Impulzné odozvy som pri plotení obmedzil na 100 vzorkov, pretože sa následne ich limita blížila k 0 a nebolo nutné ich do grafu zahrnúť. Bližšie informácie o priebehu impulznej odozvy nájdete pri spustení programu a vyplotení potrebného grafu.

Koeficienty 1. filteru:

```
B1 = [ 0.96968306, -7.19967045, 23.92468457, -46.40509761, 57.42121559,
      -46.40509761, 23.92468457, -7.19967045, 0.96968306]
A1 = [ 1, -7.3676241, 24.29453924, -46.76053222, 57.41712999,
      -46.04625104, 23.55799639, -7.03512877, 0.94028524]
```

Koeficienty 2. filtru:

$B2 = [0.95550095, -8.34660231, 36.11225767, -100.70464511, 200.24150034,$
 $-297.87987356, 339.26851941, -297.87987356, 200.24150034, -100.70464511,$
 $36.11225767, -8.34660231, 0.95550095]$

$A2 = [1, -8.66904654, 37.22281174, -103.0146623, 203.28308239,$
 $-300.1157975, 339.23048733, -295.59645228, 197.20683088, -98.43059447,$
 $35.03084292, -8.03568886, 0.91298207]$

Koeficienty 3. filtru

$B3 = [0.96968306, -3.22027134, 7.88912026, -11.880532, 14.299598,$
 $-11.880532, 7.88912026, -3.22027134, 0.96968306]$

$A3 = [1, -3.29539372, 8.0108682, -11.97119889, 14.29804551,$
 $-11.78833901, 7.76800568, -3.14667508, 0.94028524]$

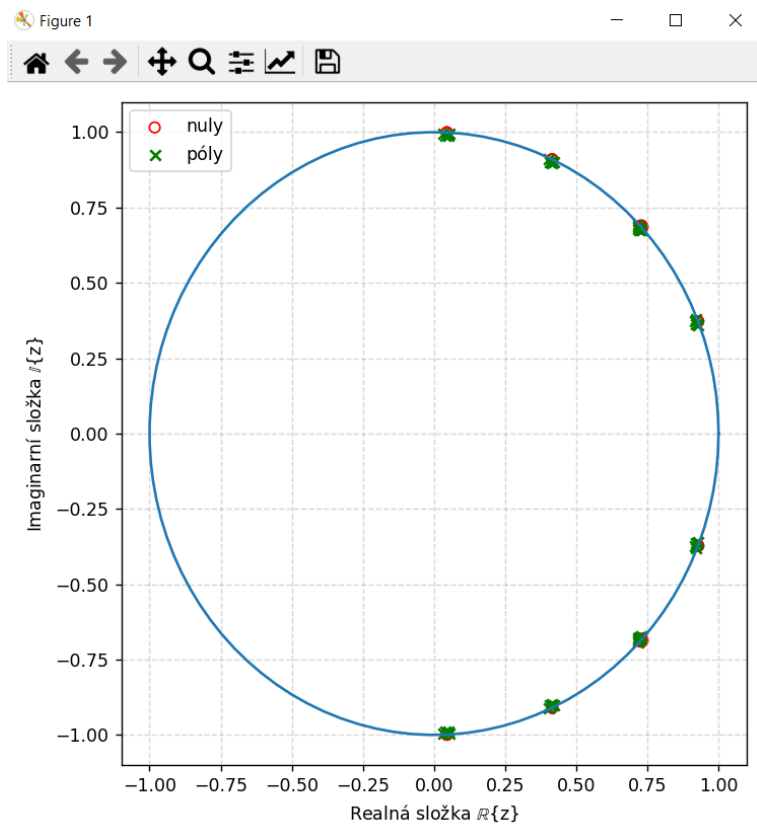
Koeficienty 4. filtru

$B4 = [0.96968306, -0.35023539, 3.92616973, -1.0535618, 5.91303779, -1.0535618,$
 $3.92616973, -0.35023539, 0.96968306]$

$A4 = [1, -0.35840567, 3.9866018, -1.06158658, 5.91211133, -1.04537104,$
 $3.865745, -0.34223109, 0.94028524]$

Úloha 8.

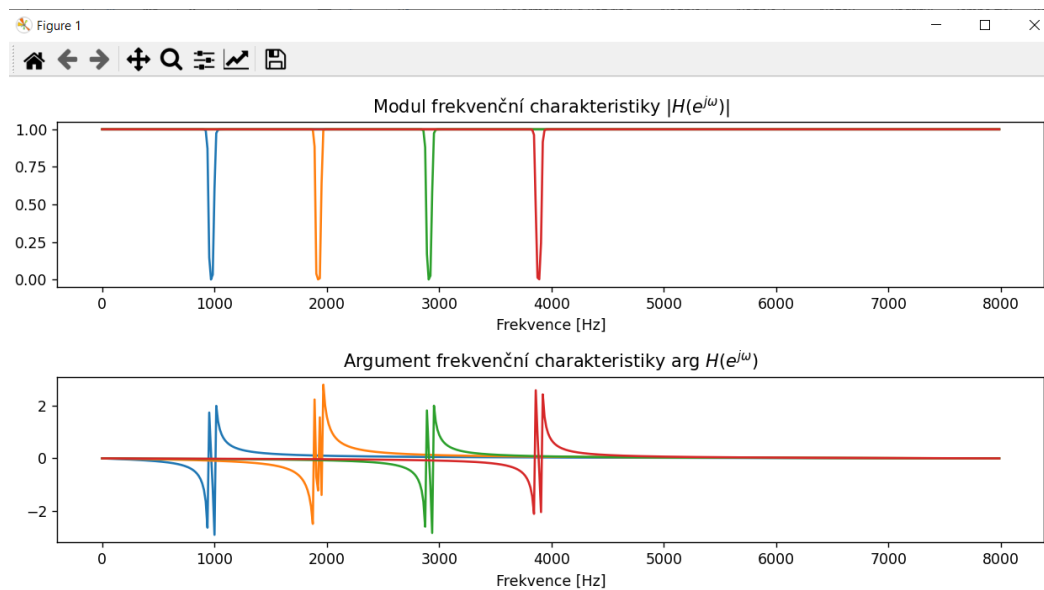
Nuly a póly som vypočítal pomocou funkcie `scipy.signal.tf2zpk(b1, a1)`. Následne som sa na základe týchto hodnôt vyplotil nasledujúci graf jednotkovej kružnice.



Obr. 8.1. – Vyplotenie núl a pólov číslicových filtrov

Úloha 9.

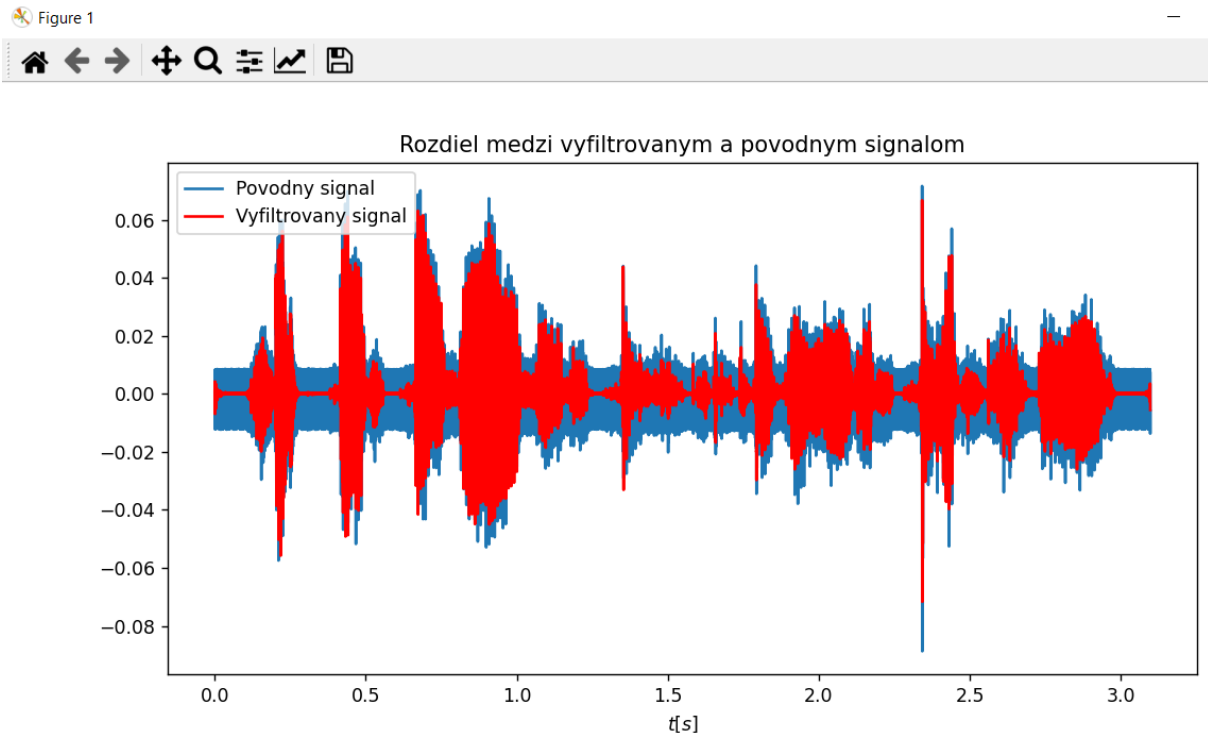
Graf frekvenčnej charakteristiky:



Obr. 9.1. – Frekvenčná charakteristika filtrov

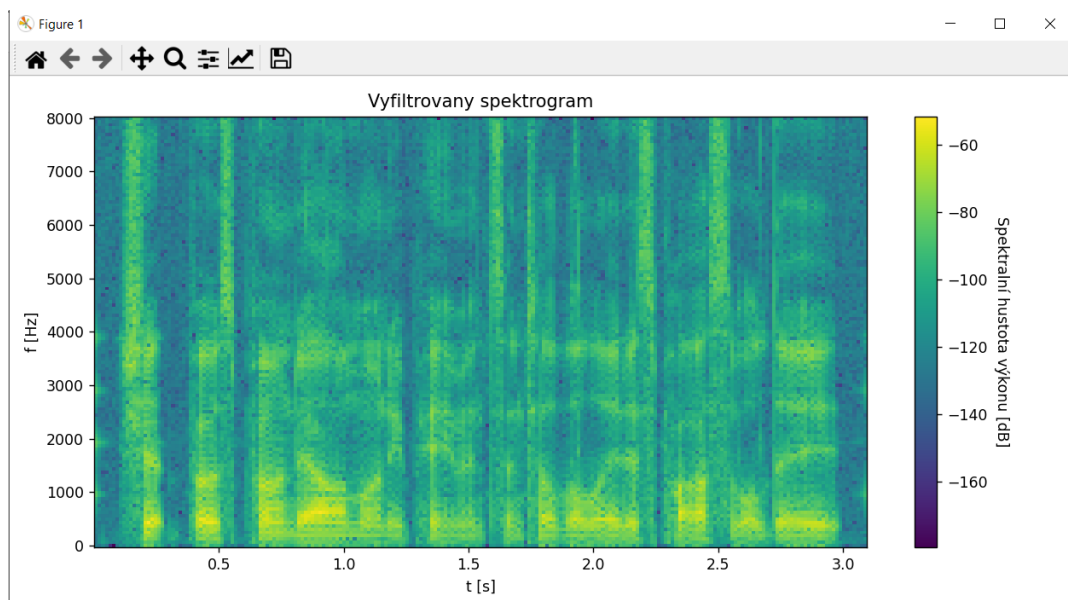
Úloha 10.

Zobrazenie pôvodného signálu a vyfiltrovaného signálu:



Obr. 10.1. – Porovnanie pôvodného signálu a vyfiltrovaného signálu

Graf spektrogramu vyfiltrovaného signálu:



Obr. 10.2. – Spektrogram vyfiltrovaného signálu

Posluchom vyfiltrovaného signálu dokážem povedať, že sa mi signál podarilo vyčistiť správne. To isté môžem tvrdiť na základe spektrogramu vyfiltrovaného signálu a grafu porovnania pôvodného a vyfiltrovaného signálu. Môžem skonštatovať, že projekt sa mi podarilo úspešne splniť.