



Signály a systémy  
Projekt 2021/2022

Tomáš Valent (xvalen27)

## 1. Úloha

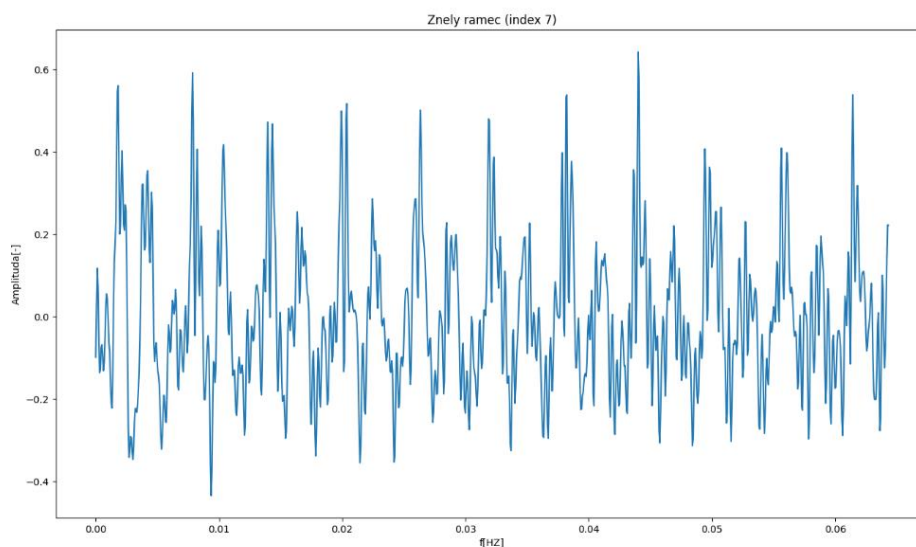
Vstupný signál	Dĺžka signálu [s]	Počet vzorkov
xvalen27.wav	2.1888125	35021

Krajné hodnoty

Minimum	Maximum
-0.84121413325554	1.0

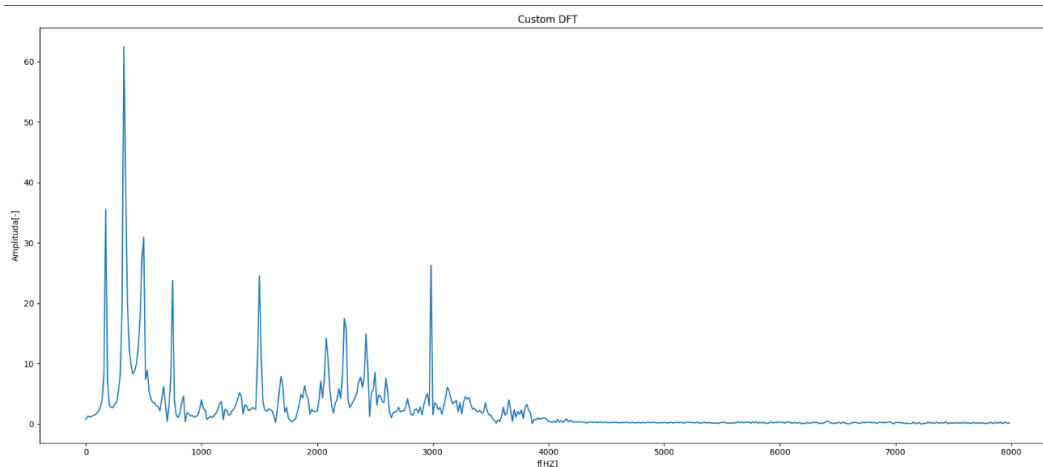
## 2. Úloha

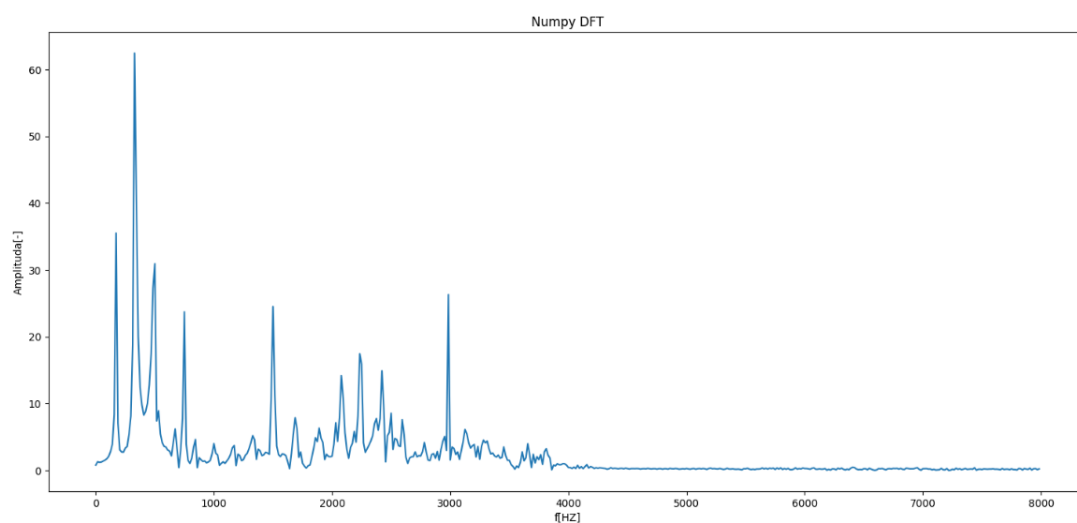
Ustrednenie prebehlo pomocou odčítania strednej hodnoty „`np.mean()`“ od signálu a normalizácia pomocou delenia signálu maximom absolútnej hodnoty „`max(abs())`“. Signál som ďalej rozdelil na 68 rámcov po 1024 vzorkoch s prekrytím 512 vzorkov. Ďalej som používal znelý rámec na indexe 7 teda 8. od začiatku.



## 3. Úloha

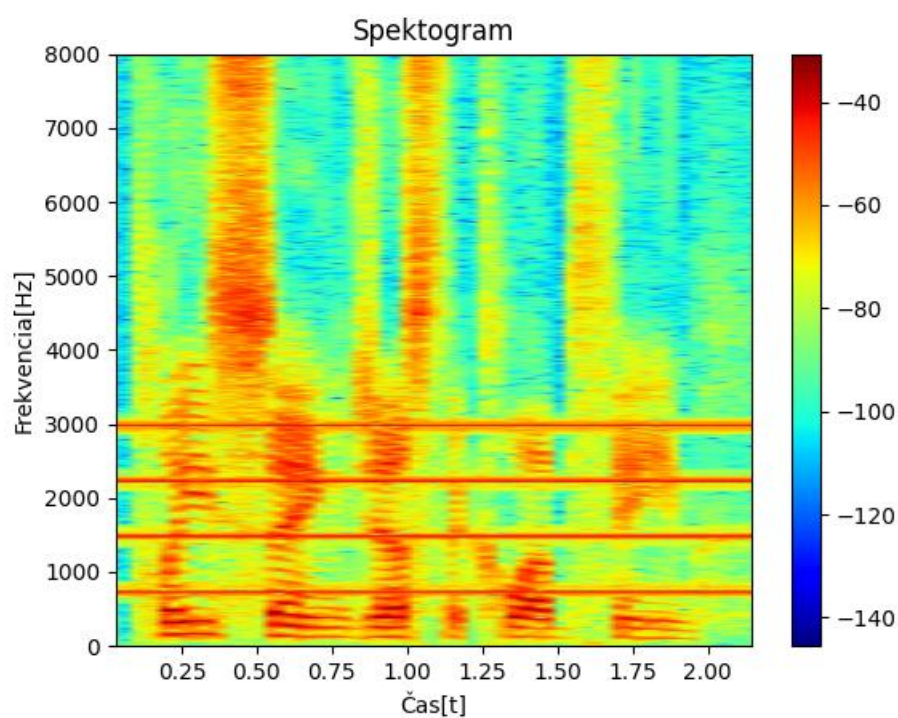
Porovnanie DFT pomocou mojej funkcie násobením matíc a DFT z NumPy knižnice na danom 8. rámci z predchádzajúcej úlohy:





#### 4. Úloha

Spektrogram:



#### 5. Úloha

Rušivé frekvencie možno ľahko vyčítať zo spektrogramu presne:

f1	750 Hz
f2	1500 Hz
f3	2250 Hz
f4	3000 Hz

## 6. Úloha

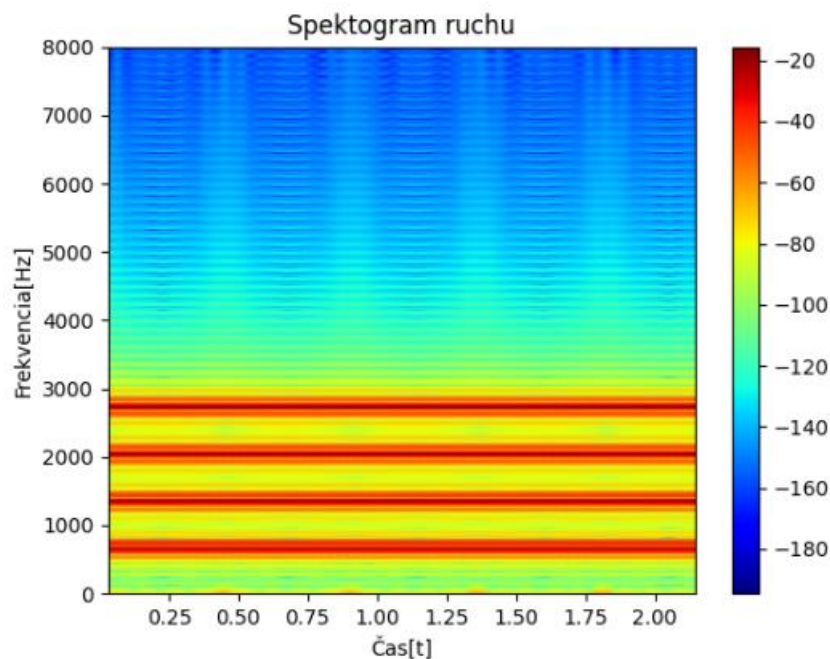
Naprav som si vygeneroval časové vzorky pre všetky 4 frekvencie pomocou funkcie „np.linspace()“, ktorá síce vyžaduje parametre typu int, ale po pretypovaní stále a vytvorení cosínusoviek nasledujúcim spôsobom som vytvoril signál s rovnakom vzorkovacou frekvenciou, dĺžkou a počtom vzorkov ako pôvodný signál.

```
out_cos1 = np.cos(2 * np.pi * f1 * time)
out_cos2 = np.cos(2 * np.pi * f2 * time)
out_cos3 = np.cos(2 * np.pi * f3 * time)
out_cos4 = np.cos(2 * np.pi * f4 * time)

output_total = out_cos1 + out_cos2 + out_cos3 + out_cos4
wavio.write("./audio/4cos.wav", output_total, Fs, sampwidth=3)
```

Premenná time je výstup funkcie np.linspace()

Spektrogram 4cos.wav:



### Zdroje:

<https://www.geeksforgeeks.org/how-to-get-the-duration-of-audio-in-python/>

<https://numpy.org/doc/stable/reference/generated/numpy.mean.html>

<https://numpy.org/doc/stable/reference/routines.fft.html>

<https://numpy.org/doc/stable/reference/generated/numpy.fft.fftfreq.html>

<https://matplotlib.org/stable/tutorials/index>

<https://jxxcarlson.medium.com/creating-audio-files-with-python-55cba61bfe73>