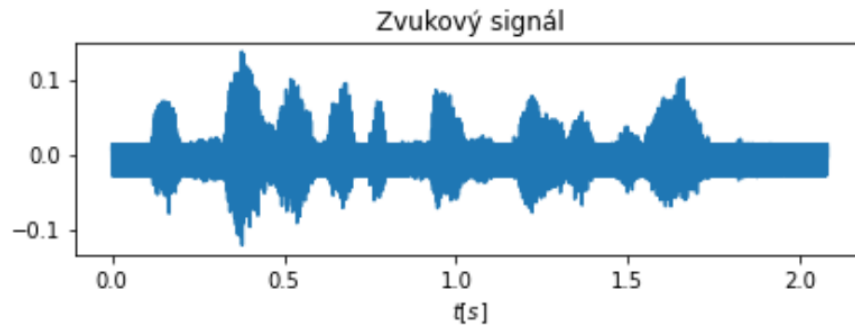


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

ISS/VSG Projekt 2021/22

Úloha 4.1

- dĺžka signálu v sekundách = 2.08 sekúnd
- dĺžka signálu vo vzorkoch = 33280 vzorkov
- minimálna hodnota = -0.1213
- maximálna hodnota = 0.1380



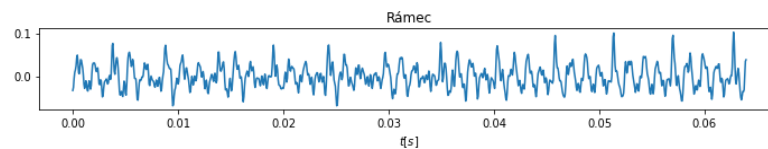
Úloha 4.2

- signál mám už normalizovaný vďaka načítaniu pomocou `soundfile.read()`
- delenie na rámce je realizované pomocou jednoduchého for cyklu

```
myMatrix = np.zeros((1024,0))
y=0
z=1024

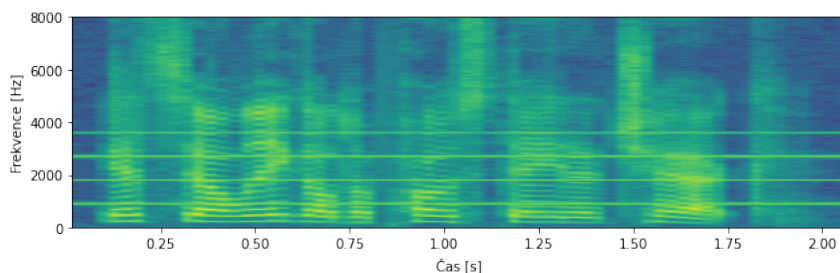
for x in range(64):
    newColon = np.array(mySignal[y:z])
    y=y+512
    z=z+512
    myMatrix = np.insert(myMatrix,x,newColon,axis=1)
```

rámec nb. 50



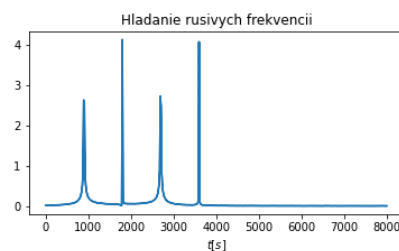
Úloha 4.4

Na vykreslenie spektrogramu používam funkciu `plt.specgram`. Môžeme na ňom vidieť štyri rušivé frekvencie.



Úloha 4.5

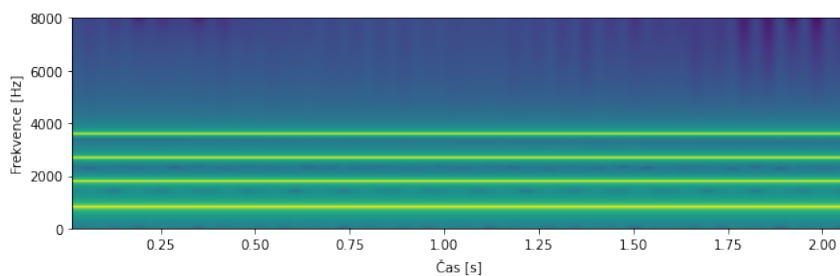
Rušivé frekvencie som určoval pomocou furierovej transformácie. Ztransformoval som prvý rámec v ktorom bolo len "pípanie", teda len rušivé frekvencie, a žiadna reč. Frekvencie som následne odčítal z grafu transformácie. použil som funkcie `numpy.fft.fft()` a `fft.fftfreq()`



f1 - 819 Hz
f2 - 1796 Hz
f3 - 2688 Hz
f4 - 3594 Hz

Úloha 4.6

Signál som vygeneroval ako súčet štyroch cosinusoviek, každá vytvorená podľa jednej z rušivých frekvencií a dĺžky pôvodného signálu. Na spektrograme je vidieť že sa skutočne jedná o rušivé frekvencie, ktoré boli viditeľné na spektrograme pôvodného signálu.



Úloha 4.7

Filter som navrhoval podľa bodu 3, teda návrh štyroch pásmových zádrží. Použil som na to funkcie `buttord` a `butter`. Vyrobil som štyri filtre a následne som nimi postupne prefilteroval pôvodný signál pomocou funkcie `lfilter`.

```

nyc=8000 #16000/2

fr1=891
left,right=signal.buttord([(fr1-90)/nyc,(fr1+90)/nyc],[(fr1-30)/nyc,(fr1+30)/nyc],3,40)
b,a= signal.butter(left, right, btype = 'bandstop')

fr2=1782
left,right=signal.buttord([(fr2-90)/nyc,(fr2+90)/nyc],[(fr2-30)/nyc,(fr2+30)/nyc],3,40)
d,c= signal.butter(left, right, btype = 'bandstop')

fr3=2673
left,right=signal.buttord([(fr3-90)/nyc,(fr3+90)/nyc],[(fr3-30)/nyc,(fr3+30)/nyc],3,40)
f,e= signal.butter(left, right, btype = 'bandstop')

fr4=3565
left,right=signal.buttord([(fr4-90)/nyc,(fr4+90)/nyc],[(fr4-30)/nyc,(fr4+30)/nyc],3,40)
h,g= signal.butter(left, right, btype = 'bandstop')

toFilter = mySignal
signalSize = mySignal.size

filteredSig = signal.lfilter(b,a,toFilter)
filteredSig = signal.lfilter(d,c,filteredSig)
filteredSig = signal.lfilter(f,e,filteredSig)
filteredSig = signal.lfilter(h,g,filteredSig)

```

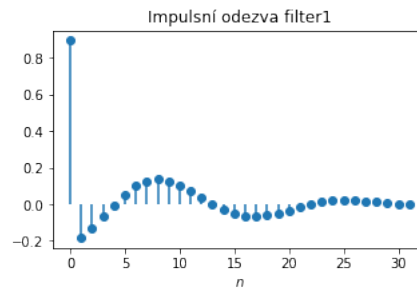
filter1

-koeficienty filtru

b: [0.89596255 -8.41735389 36.11142357 -93.10369957 159.6915466 -190.3557352 159.6915466 -93.10369957
36.11142357 -8.41735389 0.89596255]

a: [1. -9.18840599 38.55516072 -97.22999027 163.13029893 -190.22306068 156.11662019 -89.0491002 33.79303669
-7.707285 0.80274889]

-impulzná odozva



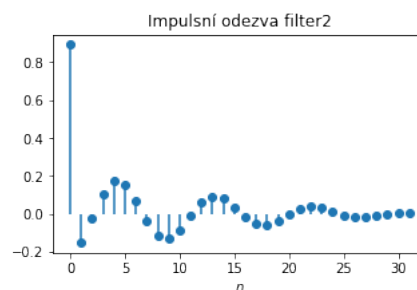
filter2

-koeficienty filtru

d: [0.89378864 -6.83777447 25.39342464 -59.36691858 96.20447715 -112.55351039 96.20447715 -59.36691858
25.39342464 -6.83777447 0.89378864]

c: [1. -7.47857378 27.14995238 -62.05272195 98.31267698 -112.46142775 93.99457931 -56.72144221 23.72731406
-6.2487308 0.79885814]

-impulzná odozva



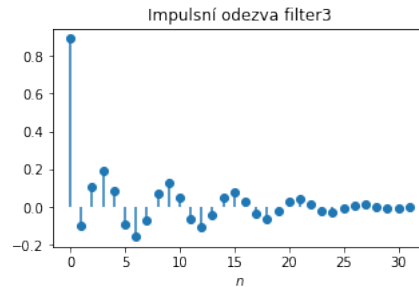
filter3

-koeficienty filtru

f: [0.89286098 -4.44540114 13.31745977 -26.59727145 39.87724324 -45.17785727 39.87724324 -26.59727145 13.31745977
-4.44540114 0.89286098]

e: [1. -4.86601858 14.24589672 -27.80795262 40.75167733 -45.13229376 38.94576696 -25.39788064 12.43458623
-4.05905687 0.79720072]

-impúlna odozva



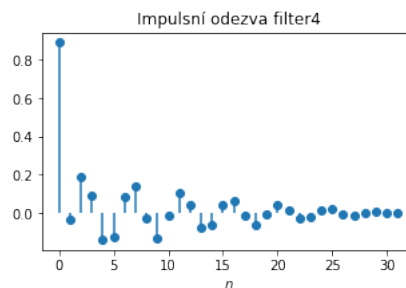
filter4

-koeficienty filtru

h: [0.89222733 -1.51686589 5.49265949 -6.41819969 12.07647006 -9.80672253 12.07647006 -6.41819969 5.49265949
-1.51686589 0.89222733]

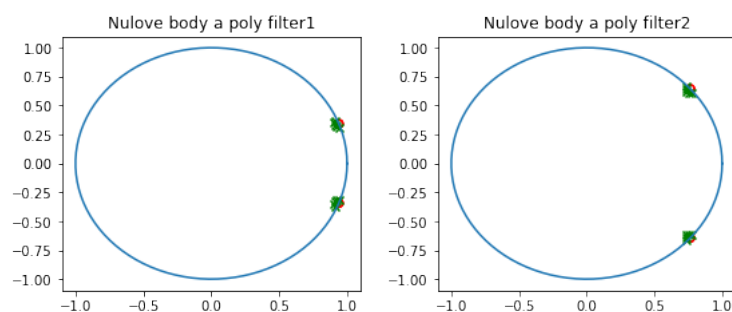
g: [1. -1.66132757 5.87615114 -6.71094787 12.33785069 -9.7944366 11.78786978 -6.12589562 5.12477254 -1.38424602
0.79606962]

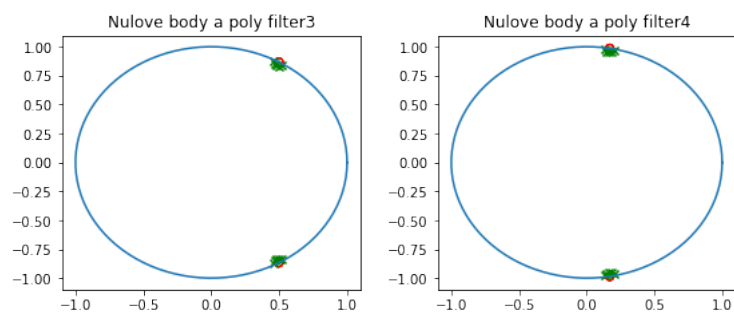
-impúlna odozva



Úloha 4.8

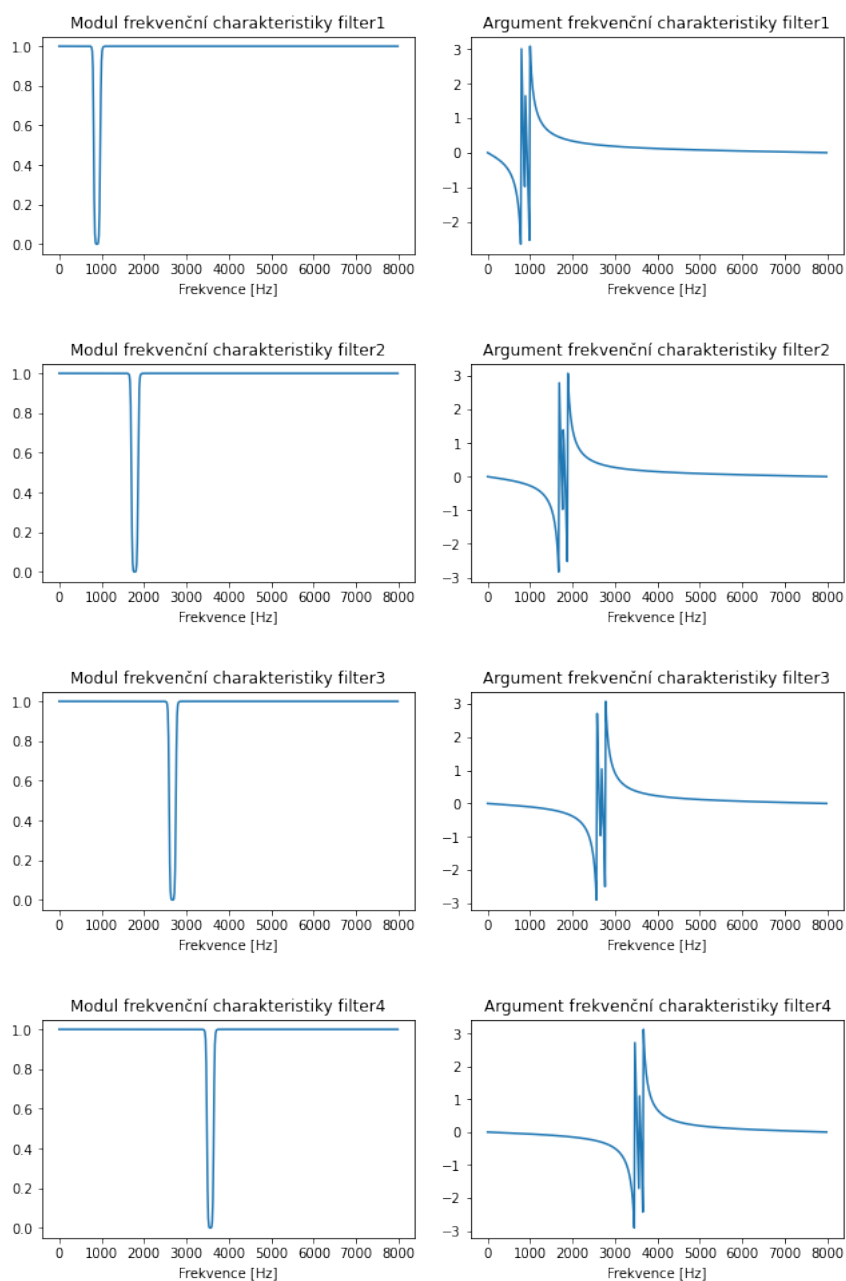
Nulové body a póly som získaval pomocou funkcie `freqz()` z knižnice `scipy` a na vykreslenie v komplexnej rovine som použil kód Kataríny Žmolíkové





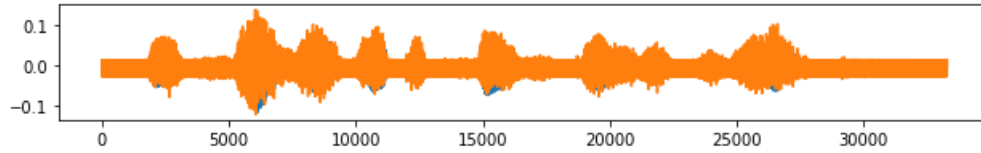
Úloha 4.9

Frekvenčnú charakteristiku som získal pomocou funkcie `freqz()` z knižnice `scipy` a pri vykresľovaní som sa taktie inšpiroval kódmi Kataríny Žmolíkové. Na grafoch môžeme vidieť že filtre filtrujú správne frekvencie.



Úloha 4.10

Moje štyri filtre úspešne odfiltrovi nežiadúce frekvencie a nový signál je čistá reč bez šumu. Na obrázku môžeme vidieť prekrytie nového filtrovaného signálu(oranžová) s pôvodným signálom (modrá)



zdroje

Cvičenia Kataríny Žmolíkové:

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