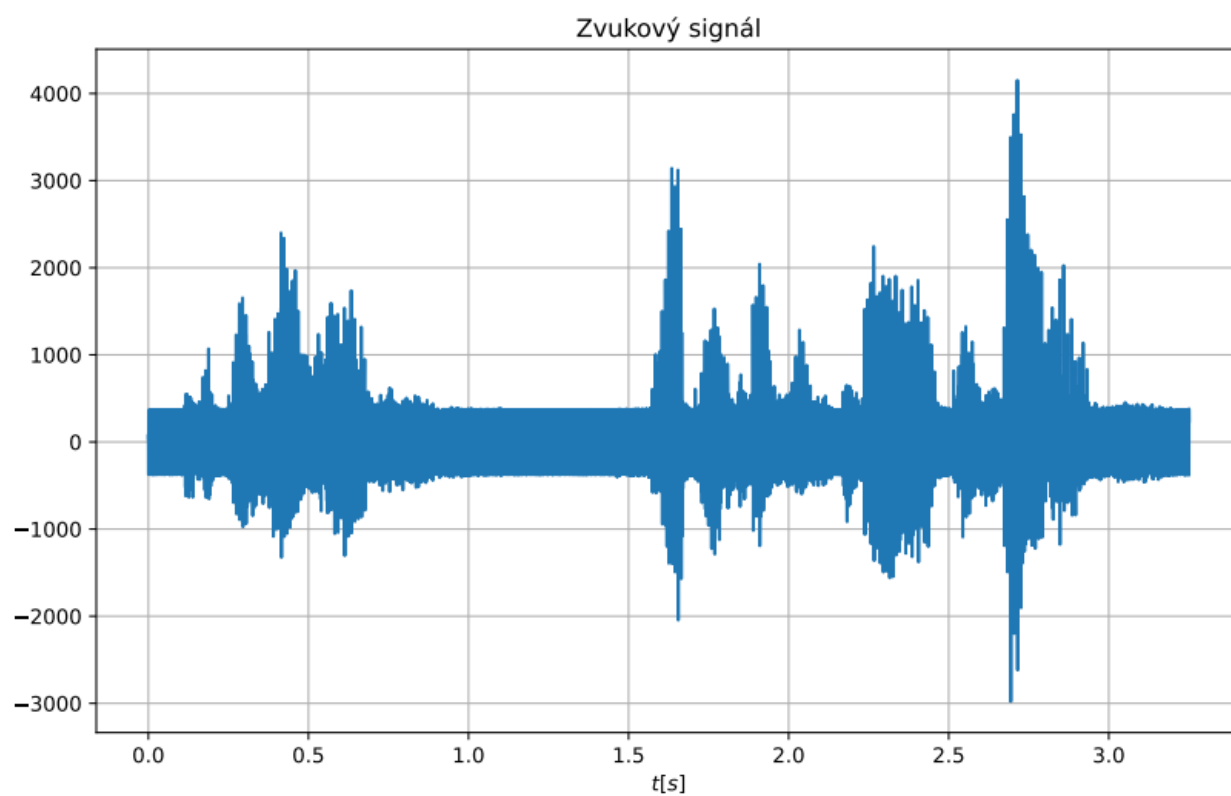


## Základy 4.1

Délka ve vzorcích je 52020  
Délka v sekundách je 3.25125  
Max hodnota je 4152  
Min hodnota je -2979

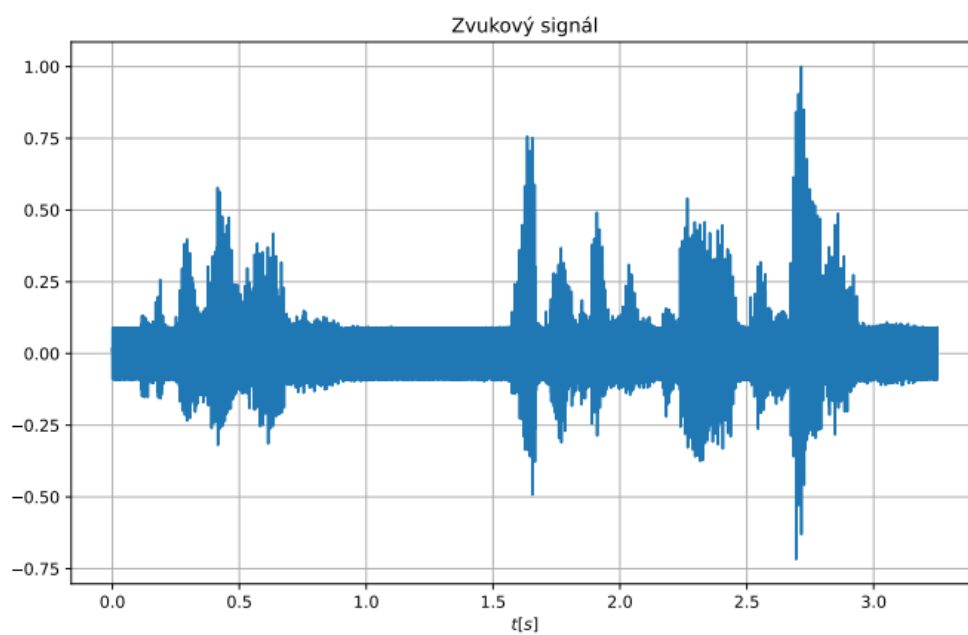


## Předzpracování a rámce 4.2

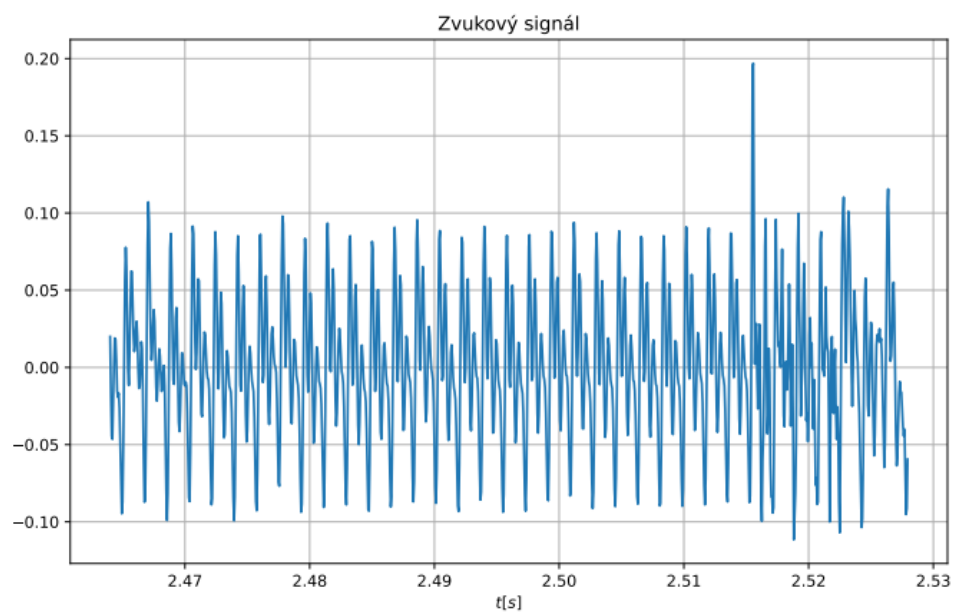
*Ustředěný signál:*

```
#Zadání 2
```

```
file_mean = np.mean(file)
file = file - file_mean
file_abs = np.abs(file)
file = file / file_abs.max()
```



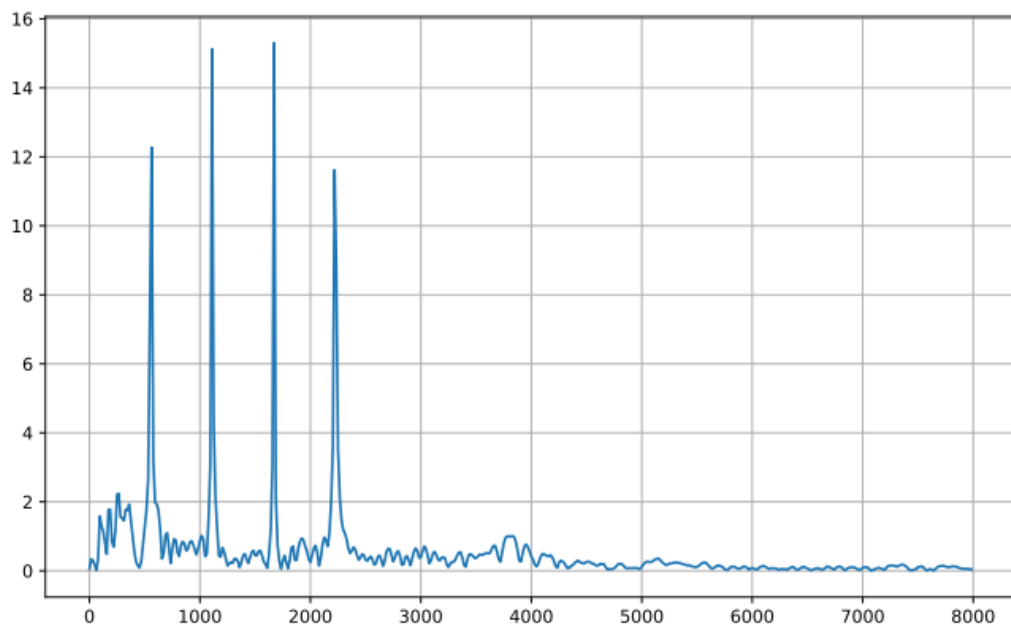
*Rámec:*



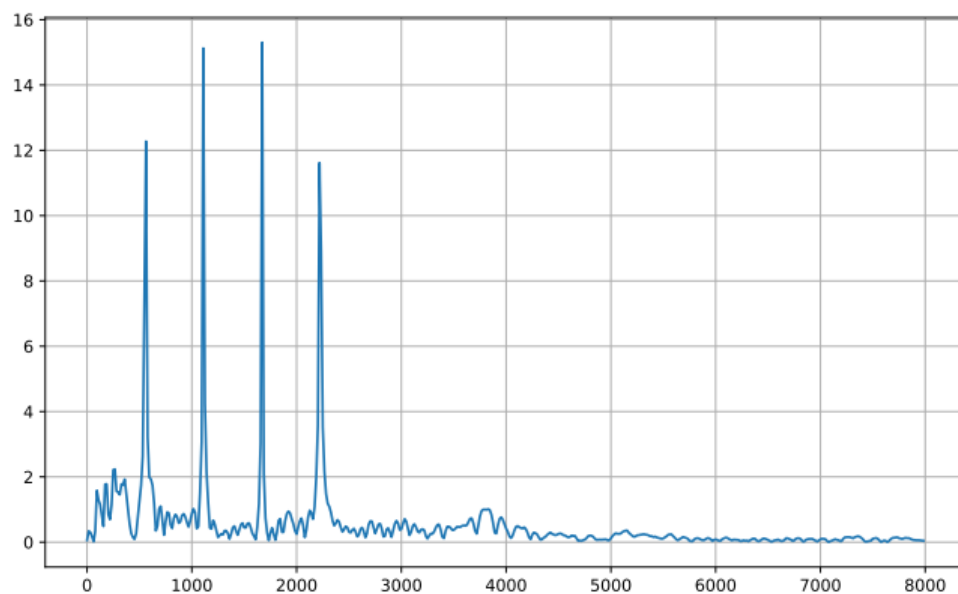
### DFT 4.3

```
#Zadani 3

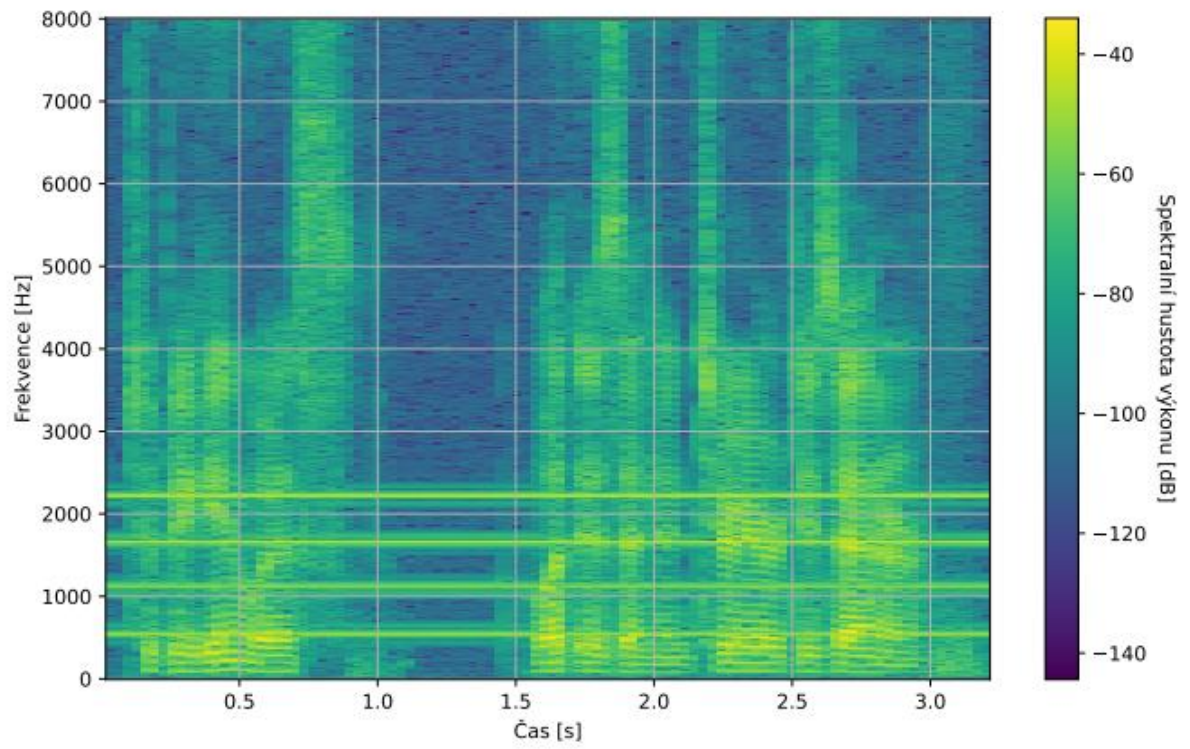
ramce_size = ramce[77].size
data = np.matrix(ramce[77]).transpose()
j1, j2 = np.meshgrid(np.arange(ramce_size), np.arange(ramce_size))
exp = np.exp(-2 * 1j * np.pi / ramce_size)
matr = np.power(exp, j1 * j2)
kfs = matr.dot(data)
file_dtf = np.squeeze(np.asarray(kfs))
```



Kontrolni dft pomoci np.fft.fft



**Spektrogram 4.4**



Udělal jsem to stejně jako bylo v notebooku Zvuk\_spektra\_filtrace.ipynb.

#### Určení rušivých frekvencí 4.5

```
#Zadani_5
dft = np.fft.fft(ramce[1])
peaks, _ = signal.find_peaks(np.abs(dft[:512]), height=1)
peaks = 8000*peaks/512
print("Peaks:", peaks)
```

Frekvence: 562.5,1109.375,1671.875,2218.75

Udělal jsem to tak, že ve vybraném rámci pomoci metody `find_peaks`, která vyhledává localní maximy, vyhledal nutné frekvence.

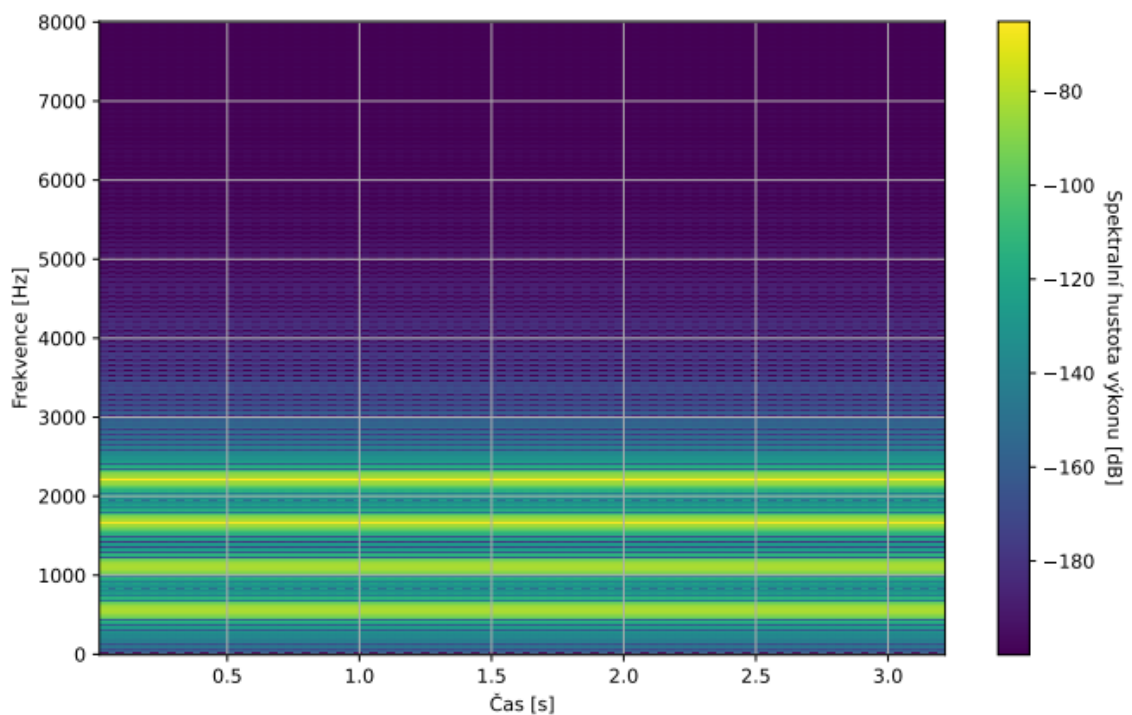
## Generování signálu 4.6

```
t_cos = np.arange(0, delka_sec, 1/fs)
i = 1
cosmix = 0.001*delka_sec*np.cos(2*np.pi*peaks[0]*t_cos)
while i < 4:
    cosmix = cosmix + 0.001*delka_sec*np.cos(2*np.pi*peaks[i]*t_cos)
    i = i + 1

wavfile.write("4cos.wav", fs, cosmix)
fs4, f4cos = wavfile.read('4cos.wav')

spectrogram(f4cos, 2, fs4)
```

Tady pomocí cyklu `while` a metody „`cos`“ jsem udělal 4 cosinusovky a sečetl jejich a udělal spektrogram s toho.



Ze spektrogramu vychází že Frekvence jsem určil správně