

ISS projekt 2021/22

Emma Krompaščíková (xkromp00)

4.1 Základy:

In [103...

```
import math
import numpy as np
from matplotlib import pyplot as plt
import soundfile as sf
import scipy.signal

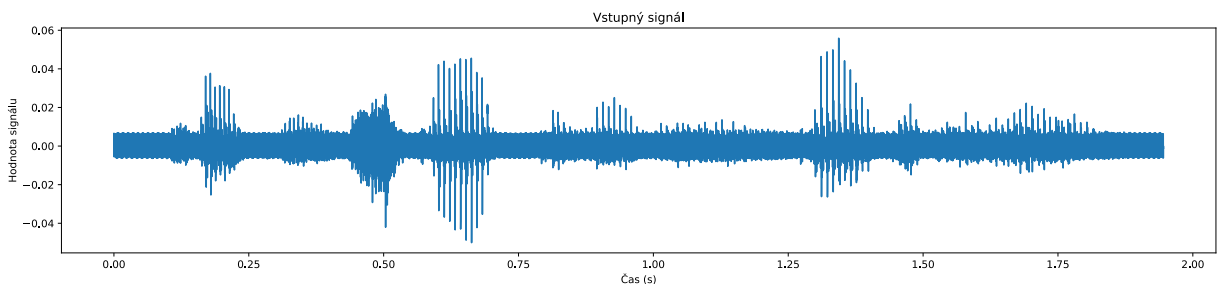
samples, sampleFrequency = sf.read('audio/xkromp00.wav')

samplesCount = len(samples)
lengthInSeconds = samplesCount / sampleFrequency

print("Dĺžka vo vzorkoch:", samplesCount)
print("Dĺžka v sekundách:", lengthInSeconds)
print("Minimálna hodnota:", min(samples))
print("Maximálna hodnota:", max(samples))

plt.figure(figsize=(20,4)) #vykreslenie grafu
plt.title("Vstupný signál")
plt.xlabel("Čas (s)")
plt.ylabel("Hodnota signálu")
plt.plot(np.arange(0, lengthInSeconds, 1/16000), samples)
plt.show()
```

Dĺžka vo vzorkoch: 31130
Dĺžka v sekundách: 1.945625
Minimálna hodnota: -0.050048828125
Maximálna hodnota: 0.055877685546875



4.2 Předzpracování a rámce:

In [104...

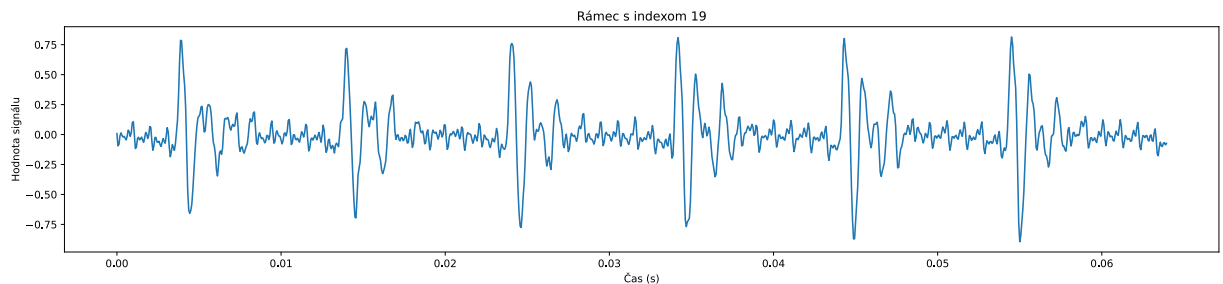
```
meanValue = np.mean(samples)
samples -= meanValue

maxAbsValue = np.abs(samples).max()
samples /= maxAbsValue

frameStart = 0
frames = []

while frameStart + 1024 < samplesCount:
    frame = samples[frameStart:frameStart + 1024] #ulozenie rámcov ako stĺpce matice
    frames.append(frame)
    frameStart += 512
```

```
plt.figure(figsize=(20,4))
plt.title("Rámec s indexom 19") #pekný rámec
plt.xlabel("Čas (s)")
plt.ylabel("Hodnota signálu")
plt.plot(np.arange(0, 1024/16000, 1/16000), frames[19])
plt.show()
```



4.3 DFT:

In [105...

```
def myDFT(signal, N):
    coeffs = []

    for k in range(N):
        base = []

        for n in range(N):
            base.append(np.exp(-2 * math.pi * 1j * n / N * k))

        coeff = np.matmul(signal, base)
        coeffs.append(coeff)

    return coeffs

def myDFTmultiple(frames, N):
    bases = []

    for k in range(0, N):
        base = []

        for n in range(0, N):
            base.append(np.exp(-2 * math.pi * 1j * n / N * k))

        bases.append(base)

    frameCoeffs = []

    for frame in frames:
        coeffs = []

        for k in range(0, N):
            coeff = np.matmul(frame, bases[k])
            coeffs.append(10 * np.log10(np.abs(coeff) ** 2))

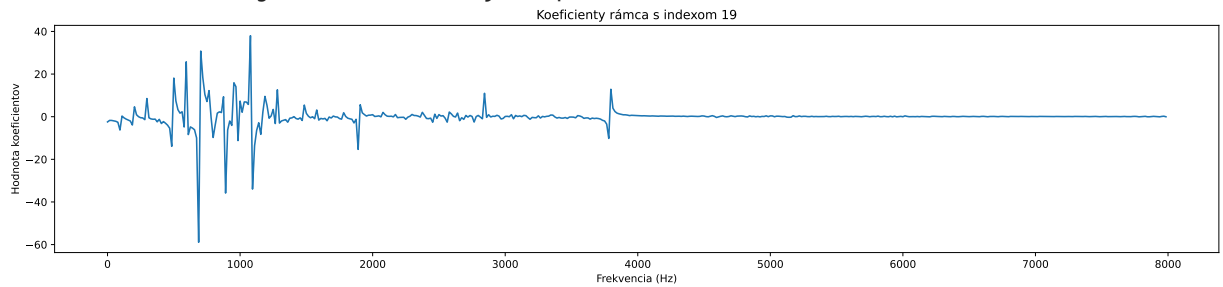
        frameCoeffs.append(coeffs[0:512])
    return frameCoeffs

coefficients = np.fft.fft(frames[19], 1024)
myCoefficients = myDFT(frames[19], 1024)

print("Porovnanie: Zhodujú sa koeficienty s np.fft.fft?", np.allclose(coefficients,
plt.figure(figsize=(20,4))
```

```
plt.title("Koeficienty rámcu s indexom 19")
plt.xlabel("Frekvencia (Hz)")
plt.ylabel("Hodnota koeficientov")
plt.plot(np.arange(0, sampleFrequency / 2, 1/1024 * 16000), myCoefficients[0:512])
plt.show()
```

Porovnanie: Zhodujú sa koeficienty s np.fft.fft? True

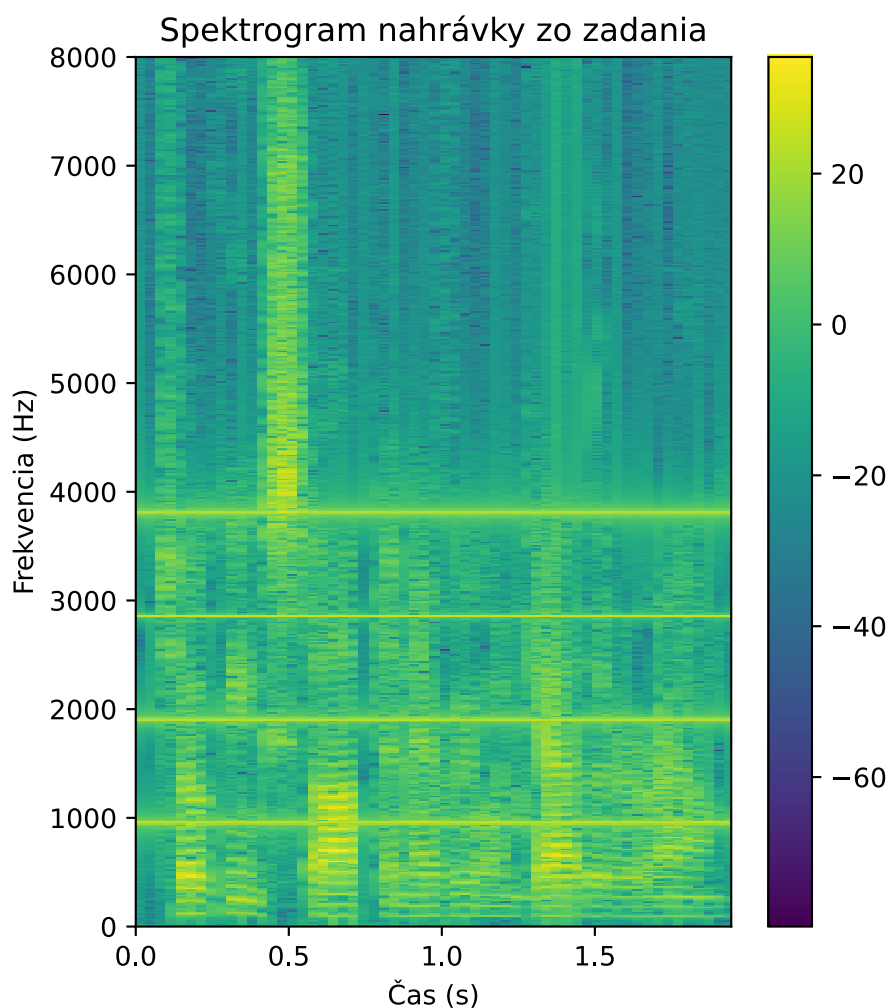


4.4 Spektrogram:

In [106...

```
coeffs = np.swapaxes(myDFTmultiple(frames, 1024), 0, 1)

plt.figure(figsize=(5,6))
plt.title("Spektrogram nahrávky zo zadania")
plt.xlabel("Čas (s)")
plt.ylabel("Frekvencia (Hz)")
spectrogram = plt.imshow(coeffs, aspect='auto', interpolation='none', origin='lower')
plt.colorbar(spectrogram)
plt.show()
```



4.5 Určení rušivých frekvencí:

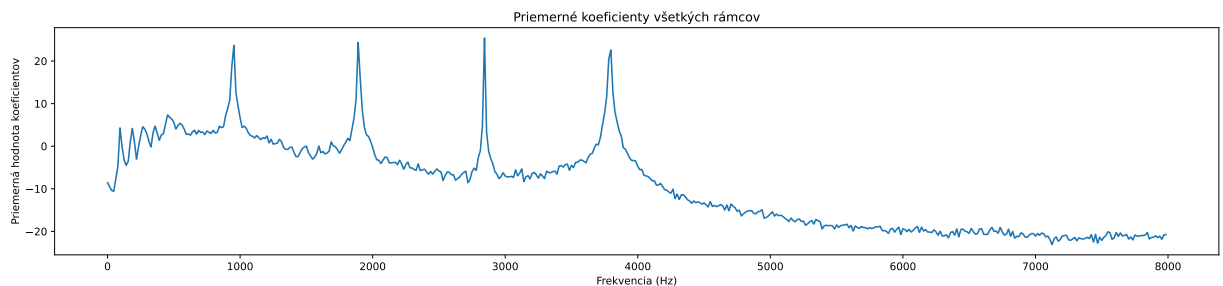
In [107...

```
averageFrame = np.mean(coeffs, 1)

plt.figure(figsize=(20,4))
plt.title("Priemerné koeficienty všetkých rámcov")
plt.xlabel("Frekvencia (Hz)")
plt.ylabel("Priemerná hodnota koeficientov")
plt.plot(np.arange(0, sampleFrequency / 2, 1/1024 * 16000), averageFrame)
plt.show()

peaks = scipy.signal.find_peaks(averageFrame, height=20)

for peak in peaks[0]:
    print("Maximum na frekvencii", peak / 1024 * sampleFrequency)
```



Maximum na frekvencii 953.125
Maximum na frekvencii 1890.625
Maximum na frekvencii 2843.75
Maximum na frekvencii 3796.875

Frekvencie sú harmonicky vzťahované.

Druhá frekvencia (1890.625) je približne dvojnásobkom prvej (953.125).

Tretia frekvencia (2843,75) je jej trojnásobkom a posledná frekvencia (3796,875) je jej štvornásobkom.

4.6 Generování signálu:

In [108...

```
newSignal = []

for sampleIndex in range(samplesCount):
    newSample = 0
    newSample += 0.01 * math.cos(2 * math.pi * 950 * sampleIndex / sampleFrequency)
    newSample += 0.01 * math.cos(2 * math.pi * 950 * 2 * sampleIndex / sampleFrequency)
    newSample += 0.01 * math.cos(2 * math.pi * 950 * 3 * sampleIndex / sampleFrequency)
    newSample += 0.01 * math.cos(2 * math.pi * 950 * 4 * sampleIndex / sampleFrequency)

    newSignal.append(newSample)

sf.write("audio/4cos.wav", newSignal, sampleFrequency)
```

In [109...

```
meanValue = np.mean(newSignal)
newSignal -= meanValue

maxAbsValue = np.abs(newSignal).max()
newSignal /= maxAbsValue

frameStart = 0

frames = []
```

```

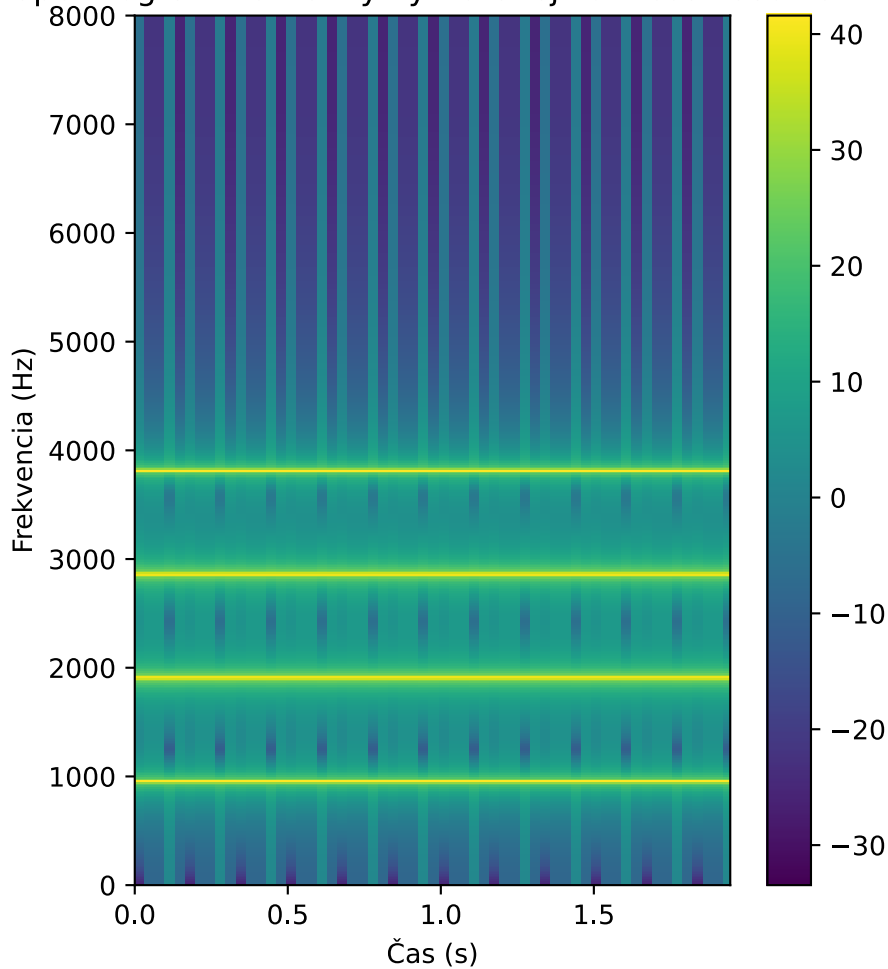
while frameStart + 1024 < samplesCount:
    frame = newSignal[frameStart:frameStart + 1024]
    frames.append(frame)
    frameStart += 512

coeffs = np.swapaxes(myDFTmultiple(frames, 1024), 0, 1)

plt.figure(figsize=(5,6))
plt.title("Spektrogram nahrávky vytvorenej zo 4 cosinusoviek")
plt.xlabel("Čas (s)")
plt.ylabel("Frekvencia (Hz)")
spectrogram = plt.imshow(coeffs, aspect='auto', interpolation='none', origin='lower')
plt.colorbar(spectrogram)
plt.show()

```

Spektrogram nahrávky vytvorenej zo 4 cosinusoviek



Porovnaním spektrogramu a posluchom som potvrdila, že frekvencie sú určené správne.