Fakultet elektrotehnike, Tuzla 2024.

Obrada Digitalnih Signala

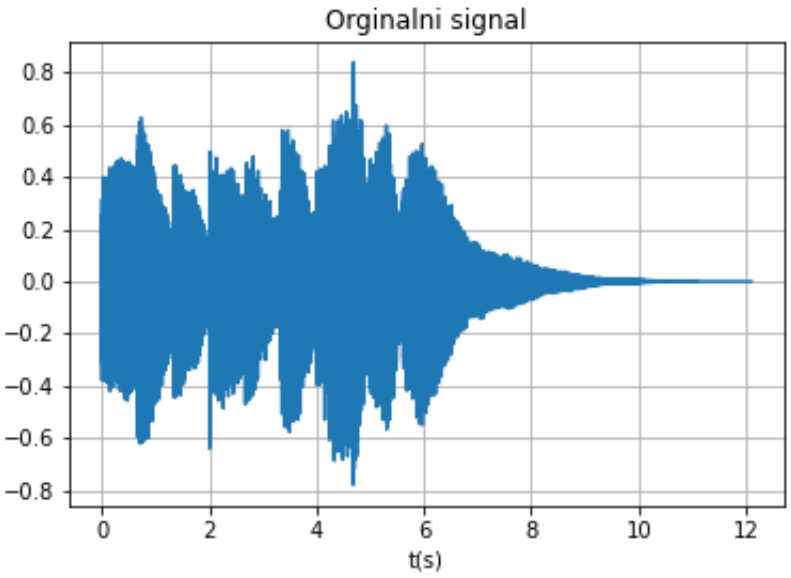
Zadaća 1

Mahir Suljić

Orginalni signal

Iz fajla *bells.wav* se pomoću `soundfile` Python biblioteke učitaje audio signal u varijablu `audio_data` sa frekvencijom uzorkovanja `samplerate`. Kako je navedeno u zadatku, frekvencija uzorkovanja se mijenja na `11025`. Varijabla `size` je broj uzoraka audio signala, varijabla `duration` je dužina signala, a varijabla `time` je vremenska osa. Ispod se plotuje signal i sprema u fajl *original.png*, koji je prikazan ispod.

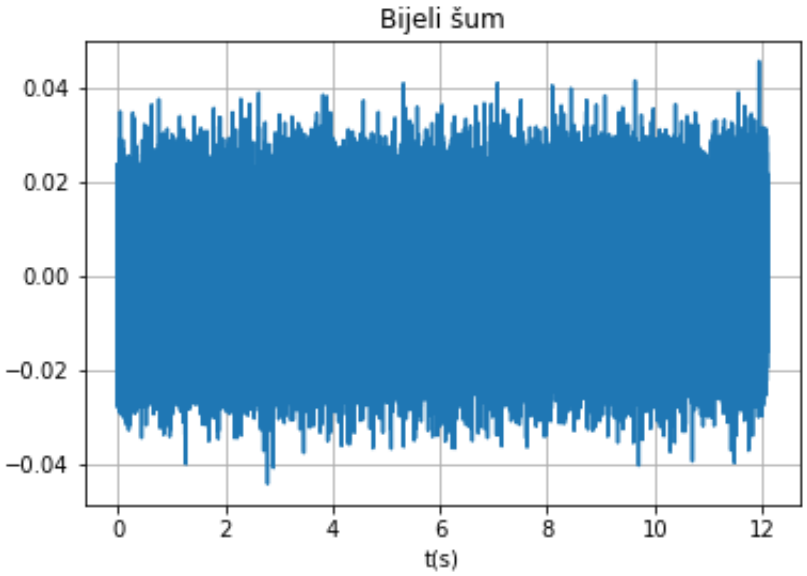
```
import matplotlib.pyplot as plt
import sounddevice as sd
import soundfile as sf
import numpy as np
import scipy as sp
from scipy import signal

file_path = "bells.wav"

audio_data, samplerate = sf.read(file_path)

audio_data = audio_data [:, 1]
samplerate = 11025

size = np.size(audio_data)
duration = size / samplerate
time = np.arange(0, duration, 1/samplerate)

plt.title("Orginalni signal")
plt.xlabel("t(s)")
plt.grid()
plt.plot(time, audio_data)
plt.savefig("original.png")
plt.show()
```



Generisanje šuma

Dalje je potrebno generisati šum. To se može uraditi pomoću paketa `numpy`. Generiše se normalna distribucija, centrirana oko vrijednosti `0`, sa maksimalnom amplitudom `0.01` i `size` uzoraka. Šum je spremljen u varijablu `noise`. Šum se generiše nasumično, a ispod koda je primjer generisanog šuma.

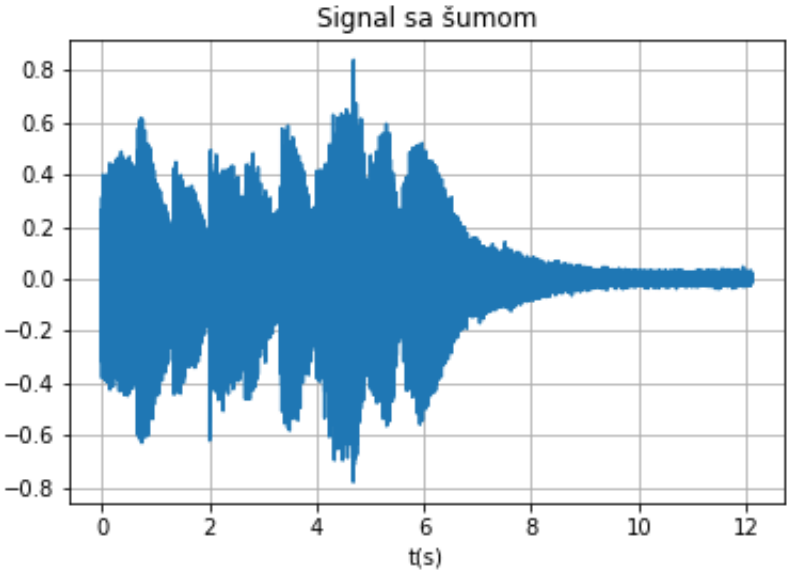
```
noise = np.random.normal(0, 0.01, size)

plt.title("Bijeli sum")
plt.xlabel("t(s)")
plt.grid()
plt.plot(time, noise)
plt.savefig("noise.png")
plt.show()
```



Dodavanje šuma na signal

Generisani šum je potrebno sabrati sa audio signalom. Rezultujući signal je spremljen u varijablu `noisy_audio`. Taj signal je prikazan na slici ispod koda.

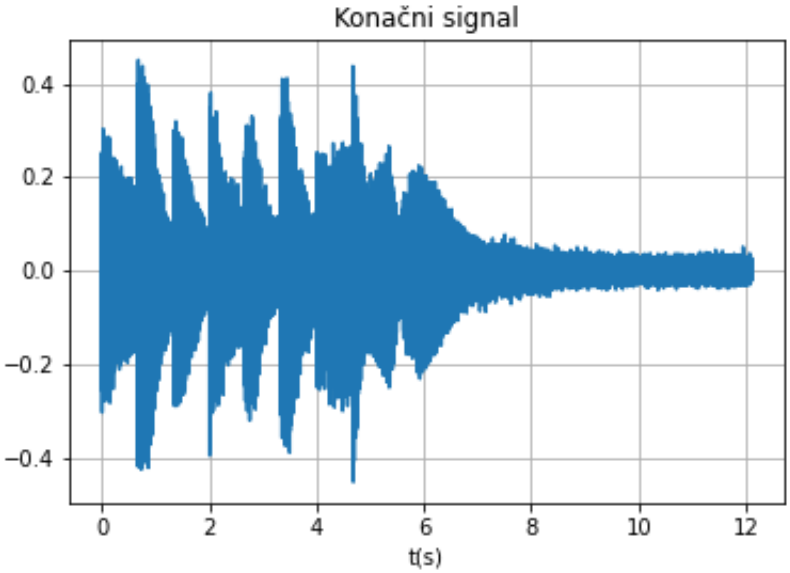
```
noisy_audio = audio_data + noise

plt.title("Signal sa sumom")
plt.xlabel("t(s)")
plt.grid()
plt.plot(time, noisy_audio)
plt.savefig("noisy_audio.png")
plt.show()
```



Konačni signal

Na kraju je potrebno izvršiti konvoluciju signala sa šumom sa signalom $h[n]$ datom u zadatku, a definisan je u kodu kao `hn`. Za konvoluciju se koristi funkcija `convolve` iz `signal` iz pakata `scipy`. Kao treći argument u funkciji `convolve` se koristi `"same"` kako bi rezultujući signal imao istu veličinu kao najveći od signala kojim se vrši konvolucija. Rezultujući signal je spremljen u varijablu `convolution`. Na kraju se dobijeni audio signal reprodukuje putem funkcije `play` iz paketa `sounddevice`.

```
hn = [-0.015, 0.058, -0.350, 1.000, -0.350, 0.058, -0.005]

convolution = signal.convolve(noisy_audio, hn, "same")

plt.title("Konacni signal")
plt.xlabel("t(s)")
plt.grid()
plt.plot(time, convolution)
plt.savefig("final.png")
plt.show()

print("Playing audio...")
sd.play(convolution, samplerate)
sd.wait()
```



Zaključak

Konačni signal je očito izobličen i ima manju amplitudu nego orginalni signal. Kada se posluša konačni signal jasno se čuje dodani šum. Ukoliko se doda šum veće amplitude, tada on počinje nadjačavati orginalni signal, te se u konačnom signalu orginali singal teže raspoznaje.