**Лабораторная работа №1**

1. path\_to\_wave **=** "/content/audio\_voice.wav"

2.**import** numpy **as** np

**from** IPython.display **import** Audio

**from** scipy.io **import** wavfile

**import** matplotlib.pyplot **as** plt

**from** itertools **import** zip\_longest

3. frame\_rate, sound\_data **=** wavfile**.**read(path\_to\_wave)

time **=** np**.**arange(0,len(sound\_data))**/**frame\_rate

sound\_data **=** sound\_data**.**T[0]

print('Sample rate:',frame\_rate,'Hz')

print(f'Total time: {len(sound\_data)**/**frame\_rate}s')

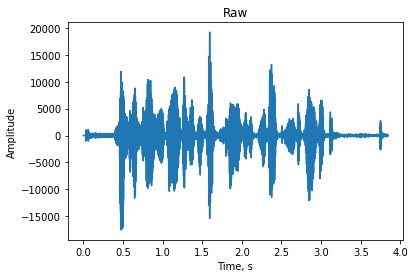
plt**.**plot(time, sound\_data)

plt**.**title("Raw")

plt**.**xlabel("Time, s")

plt**.**ylabel("Amplitude")

plt**.**show()



4. Fd **=** frame\_rate

Fs **=** frame\_rate **/** 100

m **=** 1

wn **=** 2 **\*** Fs**/** Fd *#*

5. **from** scipy **import** signal

**def** filter\_low\_pass(FD, FS, m, data):

wn **=** 2 **\*** FS**/** FD

b, a **=** signal**.**butter(m, wn, 'lowpass')

**return** signal**.**filtfilt(b, a, data)

filtedData **=** filter\_low\_pass(Fd, Fs, m, sound\_data)

fig, ax **=** plt**.**subplots()

plt**.**title("Low filted data")

plt**.**xlabel("Time, s")

plt**.**ylabel("Amplitude")

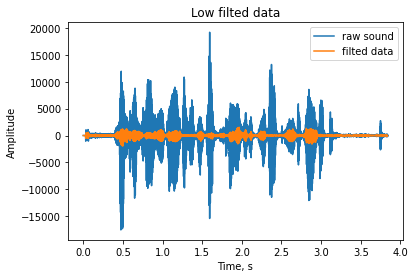
ax**.**plot(time, sound\_data, label **=** 'raw sound')

ax**.**plot(time, filtedData, label **=** 'filted data')

ax**.**legend()

plt**.**show()

Audio(filtedData,rate**=**frame\_rate)



## Фильтр высоких частот.

Фильтр высоких частот - фильтр, пропускающий высокие частоты входного сигнала, при этом подавляя частоты сигнала ниже частоты среза.



6. Fs **=** frame\_rate **/** 10

7. **def** filter\_high\_pass(FD, FS, m, data):

wn **=** 2 **\*** FS**/** FD

b, a **=** signal**.**butter(m, wn, 'highpass')

**return** signal**.**filtfilt(b, a, data)

filtedData **=** filter\_high\_pass(Fd, Fs, m, sound\_data)

fig, ax **=** plt**.**subplots()

plt**.**title("High filted data")

plt**.**xlabel("Time, s")

plt**.**ylabel("Amplitude")

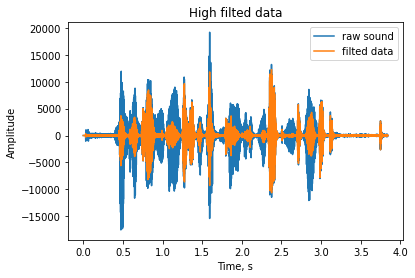
ax**.**plot(time, sound\_data, label **=** 'raw sound')

ax**.**plot(time, filtedData, label **=** 'filted data')

ax**.**legend()

plt**.**show()

Audio(filtedData,rate**=**frame\_rate)



## Distortion

12. TH **=** 10000

MAX **=** 30000

13. distortion\_data **=** []

**for** i in sound\_data:

**if** abs(i) **<=** TH:

value **=** i

**else**:

value **=** MAX **\*** (**-**1 **if** i **<** 0 **else** 1)

distortion\_data**.**append(value)

fig, ax **=** plt**.**subplots()

plt**.**title("Distortion")

plt**.**xlabel("Time, s")

plt**.**ylabel("Amplitude")

ax**.**plot(time, distortion\_data, label **=** 'distortion data')

ax**.**plot(time, sound\_data, label **=** 'raw sound')

ax**.**legend()

plt**.**show()

Audio(np**.**array(distortion\_data, dtype**=**np**.**int16),rate**=**frame\_rate)

