

Lab05

April 20, 2021

1 Sygnały akustyczne

1.1 Laboratorium 05 - Projektowanie filtrów IIR (NOI)

Karol Działowski

Celem laboratorium było zaprojektowanie i porównanie dwóch różnych filtrów NOI o dowolnie wybranych parametrach za pomocą gotowych funkcji na podstawie materiałów wykładowych.

```
[106]: import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from scipy.fft import fft, fftshift

import warnings
warnings.filterwarnings('ignore')
```

```
[107]: def plot_charakterystyka(w, h, fs):
    fig, axes = plt.subplots(1, 2, figsize=(15, 6))
    x = w * fs / (2 * np.pi)
    y = 20 * np.log10(np.maximum(abs(h), 1e-5))
    axes[0].plot(x, y)
    axes[0].set_xlim([0, 4000])
    axes[0].set_xlabel("Częstotliwość [Hz]")
    axes[0].set_ylabel("Amplituda [dB]")
    axes[0].title.set_text("Charakterystyka częstotliwościowa")

    y = np.arctan2(np.imag(h), np.real(h))
    y = np.degrees(np.unwrap(y))
    axes[1].plot(x, y)
    axes[1].set_xlabel("Częstotliwość [Hz]")
    axes[1].title.set_text("Charakterystyka fazowa")

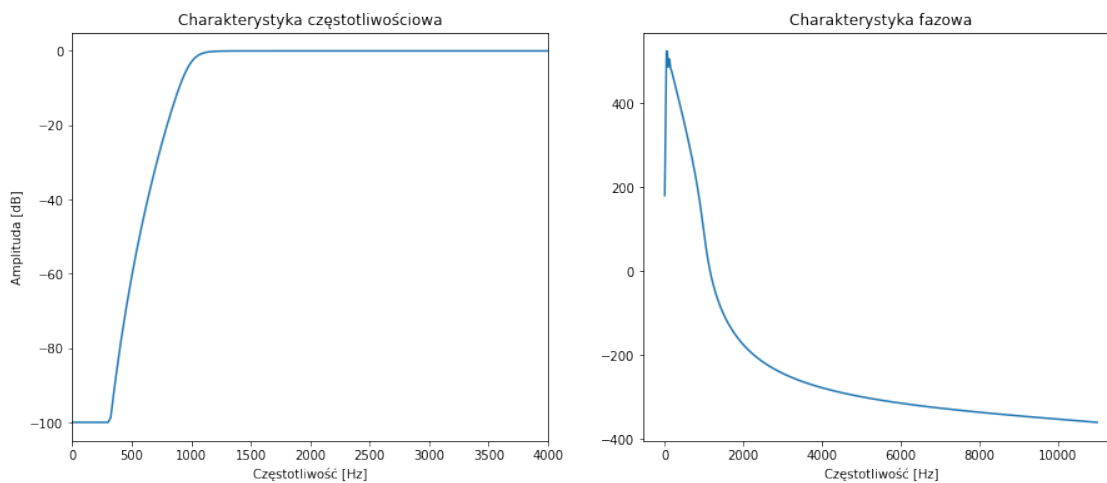
    plt.show()
```

1.2 Filtr dolno-przepustowy Butterwortha

```
[108]: fs = 22050 # czestotliwosc probkowania
n = 10 # rzad filtra
fc = 1000 # czestotliwosc odciecia

b, a = signal.iirfilter(n, fc / (fs / 2), btype='highpass', ftype='butter')
w, h = signal.freqz(b, a)
b_butter = b
```

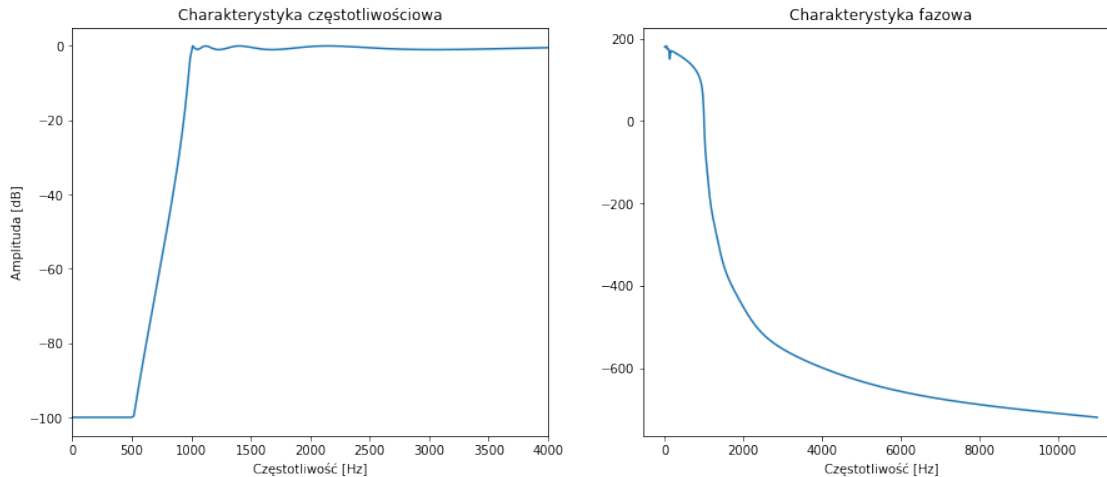
```
[109]: plot_charakterystyka(w, h, fs)
```



1.3 Filtr dolno-przepustowy Czebyszewa typu 1

```
[110]: b, a = signal.iirfilter(n, fc / (fs / 2), btype='highpass', ftype='cheby1',
    ↪rp=1)
w, h = signal.freqz(b, a)
b_cheby1 = b
```

```
[111]: plot_charakterystyka(w, h, fs)
```



1.4 Filtracja wybranego sygnału dźwiękowego

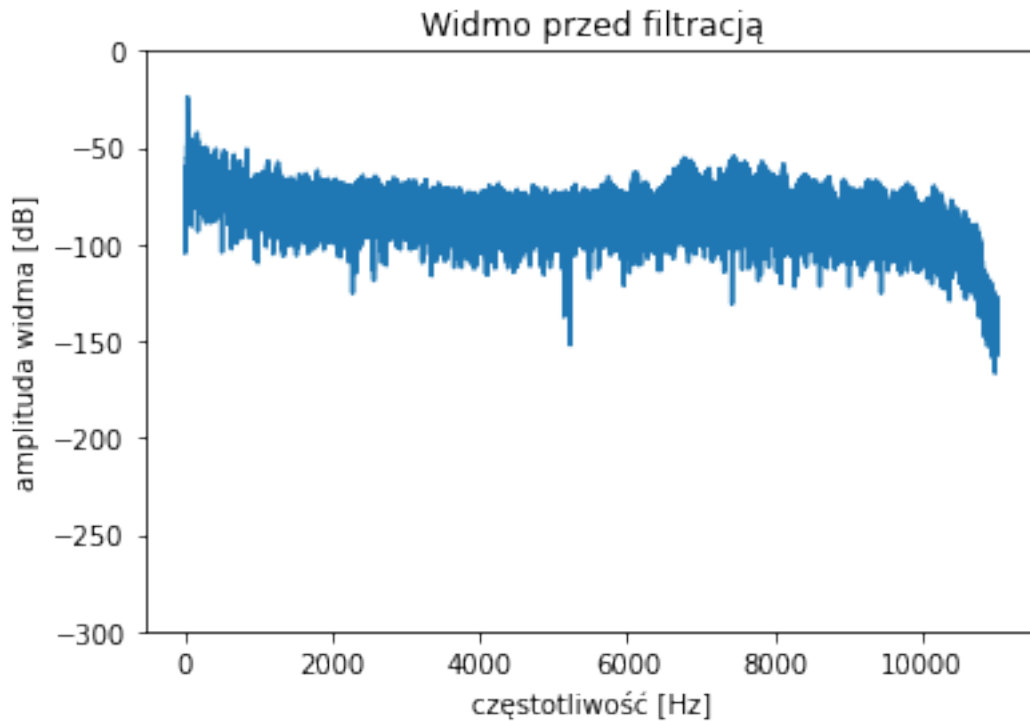
```
[112]: from scipy import signal
from scipy import fft
import numpy as np
import matplotlib.pyplot as plt
import librosa
import librosa.display
from scipy.io import wavfile
```

```
[113]: def plot_widmo(y, fs, title):
    widmo_amp = 20 * np.log10(np.abs(np.fft.rfft(y)) / (len(y)//2))
    f = np.fft.rfftfreq(len(y), 1/fs)
    plt.plot(f, widmo_amp)
    plt.xlabel('częstotliwość [Hz]')
    plt.ylabel('amplituda widma [dB]')
    plt.ylim([-300, 0])
    plt.title(title)
    plt.show()
```

1.4.1 Wczytanie wybranego pliku

```
[114]: y, sr = librosa.load("./data/dziendobry.wav", mono=True)
y = y.astype('float32')
```

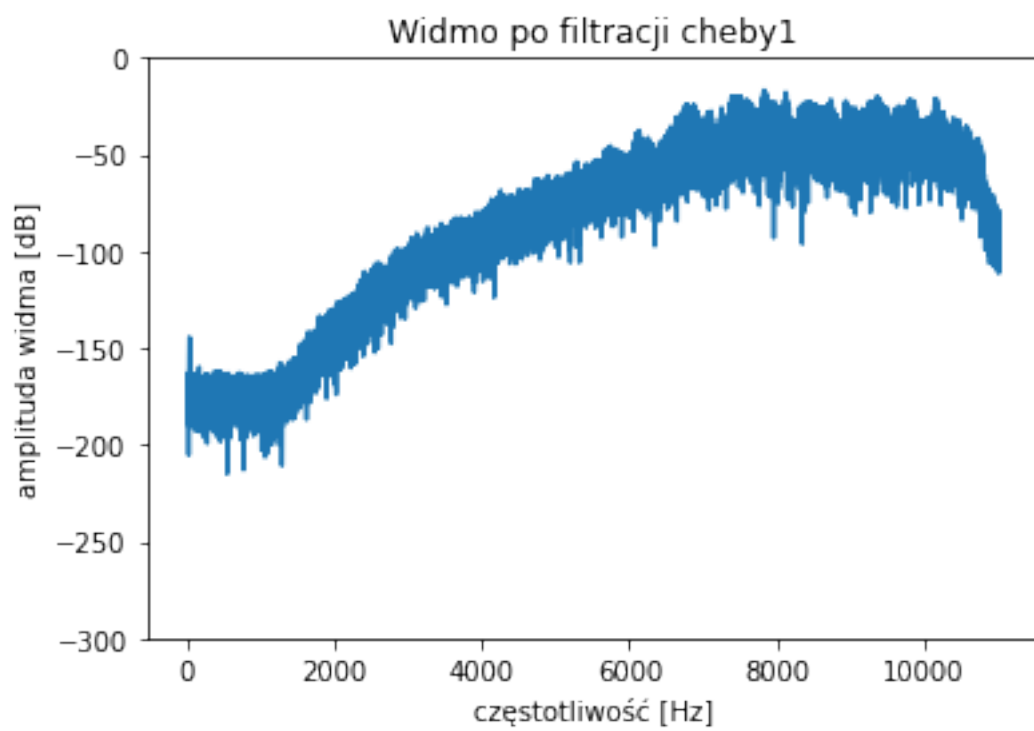
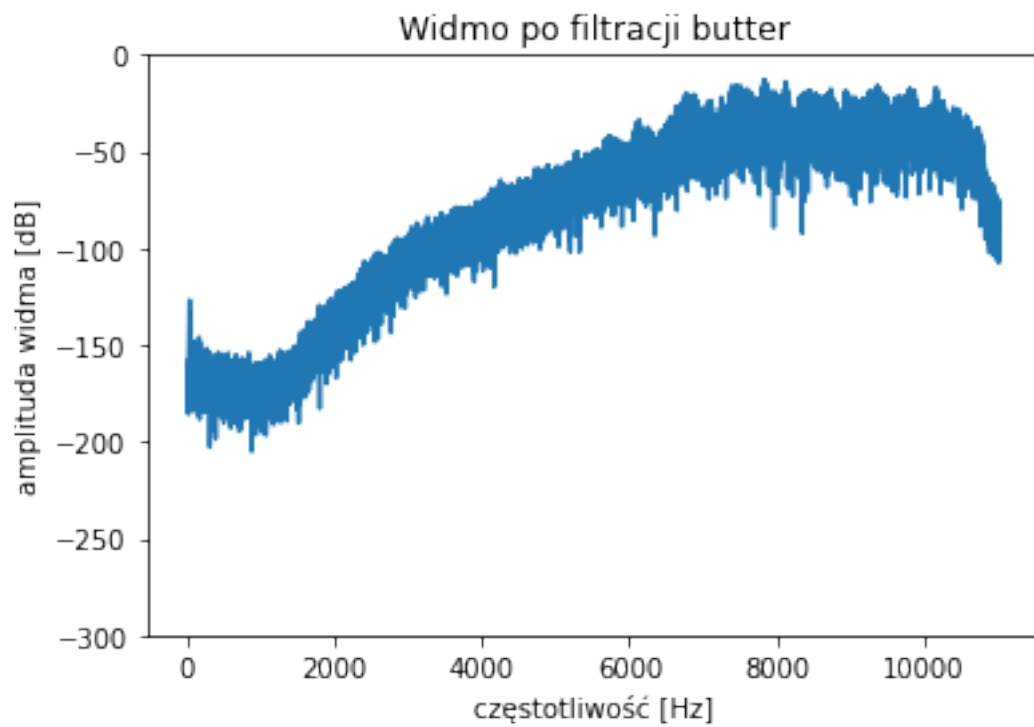
```
[115]: plot_widmo(y, sr, 'Widmo przed filtracją')
```



1.4.2 Filtracja

```
[116]: def filtering(y, b, name):
        filtered_y = signal.convolve(y.astype('float32'), b.astype('float32'))
        wavfile.write(f"./data/filtered_{name}.wav", sr, filtered_y)
        return filtered_y

        filtered_y_butter = filtering(y, b_butter, "butter")
        plot_widmo(filtered_y_butter, sr, 'Widmo po filtracji butter')
        filtered_y_cheby1 = filtering(y, b_cheby1, "cheby")
        plot_widmo(filtered_y_cheby1, sr, 'Widmo po filtracji cheby1')
```



1.5 Pasma oktawaowe

Pojęcia oktawa używa także do określenia pasma częstotliwości, którego górna i dolna granica pozostają w stosunku 2:1; oktawa stanowią trzy kolejne tercje. W tym przypadku definiuje się trzy częstotliwości charakterystyczne dla pasma oktawy: f_d - częstotliwość dolna, f_o - częstotliwość środkowa, f_g - częstotliwość górna, które pozostają w następujących zależnościach:

$$\begin{aligned} \frac{f_g}{f_d} &= \frac{2}{1} \\ f_d &= f_o \cdot \sqrt[2]{2} \\ f_g &= f_o \cdot \sqrt[2]{2} \\ f_o &= f_d \cdot \sqrt[2]{2} = f_g \cdot \sqrt[2]{2} \end{aligned}$$

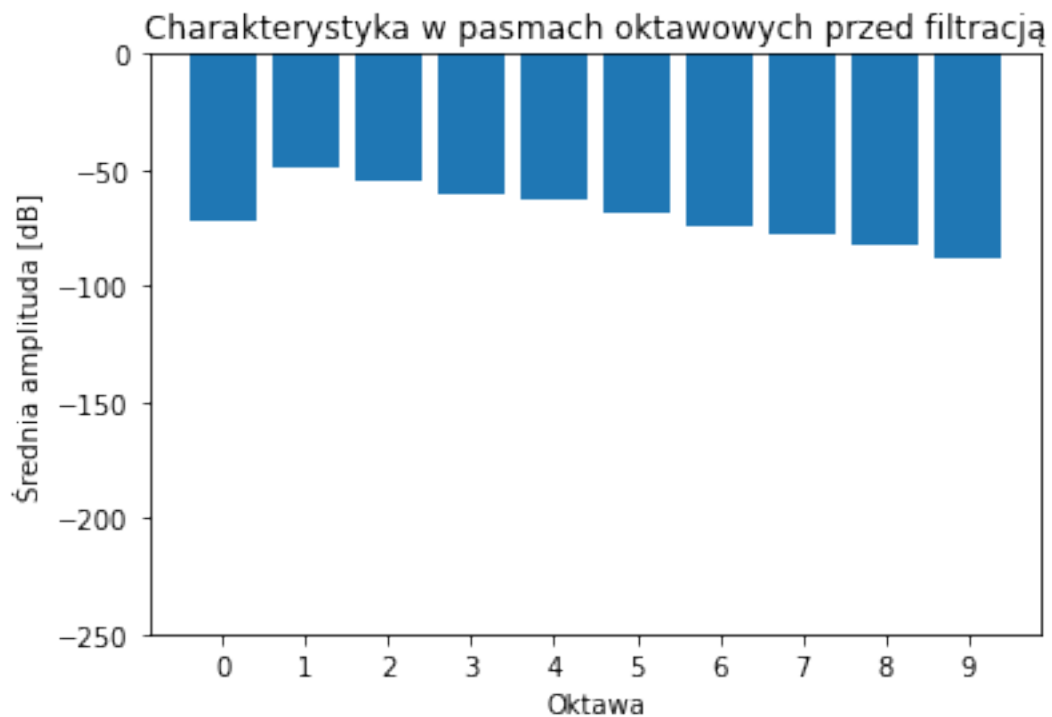
Źródło: [https://www.wikiwand.com/pl/Oktawa_\(akustyka\)](https://www.wikiwand.com/pl/Oktawa_(akustyka))

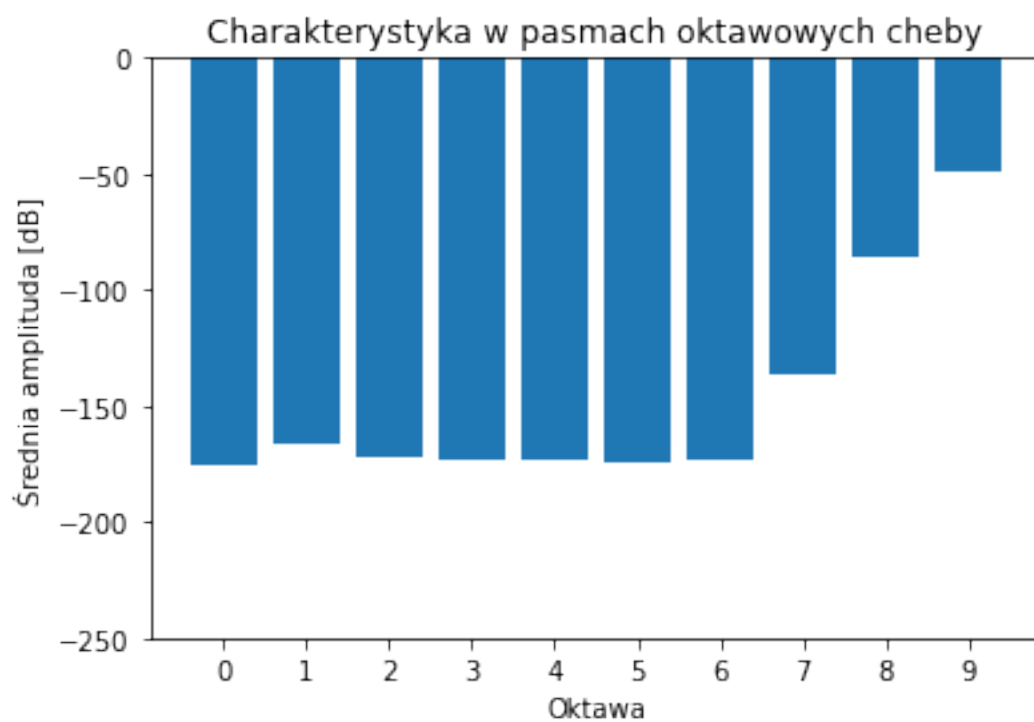
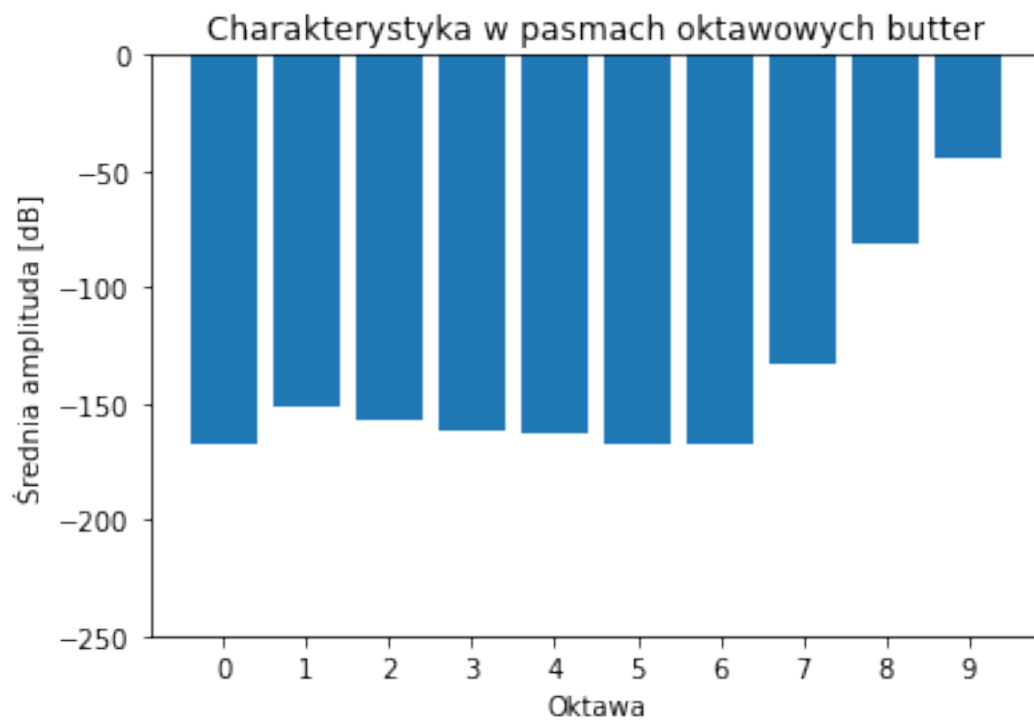
```
[117]: def plot_octaves(y, fs, title, verbose=False):
    # Widmo
    widmo_amp = 20 * np.log10(np.abs(np.fft.rfft(y)) / (len(y)//2))
    # Oś X - częstotliwości
    f = np.fft.rfftfreq(len(y), 1/fs)
    # Częstotliwości środkowe do oktav
    czestotliwosci_srodkowe = [16, 31.5, 63, 125, 250, 500, 1000, 2000, 4000,
    ↪8000]
    srednia_z_pasma = []
    for idx, f_o in enumerate(czestotliwosci_srodkowe):
        f_d = f_o * np.power(2, -1/2)
        f_g = f_o * np.power(2, 1/2)
        if verbose:
            print("Oktawa:", idx+1, "częstotliwość dolna:", f_d, "środkowa:",
    ↪f_o, "górna:", f_g)
        lower_bound = np.argmax(f>f_d)
        upper_bound = np.argmax(f>f_g)
        if upper_bound == 0:
            upper_bound = -1
        srednia_z_pasma.append(np.mean(widmo_amp[lower_bound:upper_bound]))

    plt.bar(np.arange(10), srednia_z_pasma)
    plt.ylim([-250, 0])
    plt.xticks(np.arange(10))
    plt.xlabel("Oktawa")
    plt.ylabel("Średnia amplituda [dB]")
    plt.title(f"Charakterystyka w pasmach oktaowych {title}")
    plt.show()

plot_octaves(y, sr, "przed filtracją", verbose=True)
plot_octaves(filtered_y_butter, sr, "butter")
plot_octaves(filtered_y_cheby1, sr, "cheby")
```

Oktawa: 1 częstotliwość dolna: 11.313708498984761 środkowa: 16 górna: 22.627416997969522
 Oktawa: 2 częstotliwość dolna: 22.27386360737625 środkowa: 31.5 górna: 44.5477272147525
 Oktawa: 3 częstotliwość dolna: 44.5477272147525 środkowa: 63 górna: 89.095454429505
 Oktawa: 4 częstotliwość dolna: 88.38834764831844 środkowa: 125 górna: 176.7766952966369
 Oktawa: 5 częstotliwość dolna: 176.7766952966369 środkowa: 250 górna: 353.5533905932738
 Oktawa: 6 częstotliwość dolna: 353.5533905932738 środkowa: 500 górna: 707.1067811865476
 Oktawa: 7 częstotliwość dolna: 707.1067811865476 środkowa: 1000 górna: 1414.213562373095
 Oktawa: 8 częstotliwość dolna: 1414.213562373095 środkowa: 2000 górna: 2828.42712474619
 Oktawa: 9 częstotliwość dolna: 2828.42712474619 środkowa: 4000 górna: 5656.85424949238
 Oktawa: 10 częstotliwość dolna: 5656.85424949238 środkowa: 8000 górna: 11313.70849898476





1.6 Tercje

Definiuje się trzy częstotliwości charakterystyczne dla pasma tercji: - f_d - częstotliwość dolna, - f_o - Częstotliwość środkowa, - f_g - częstotliwość górna,

które pozostają w następujących zależnościach:

$$\begin{aligned}\frac{f_g}{f_d} &= \sqrt[3]{2} \\ f_d &= f_o \cdot \sqrt[6]{2} \\ f_g &= f_o \cdot \sqrt[6]{2} = f_d \cdot \sqrt[3]{2} \\ f_o &= f_d \cdot \sqrt[6]{2} = f_g \cdot \sqrt[6]{2}.\end{aligned}$$

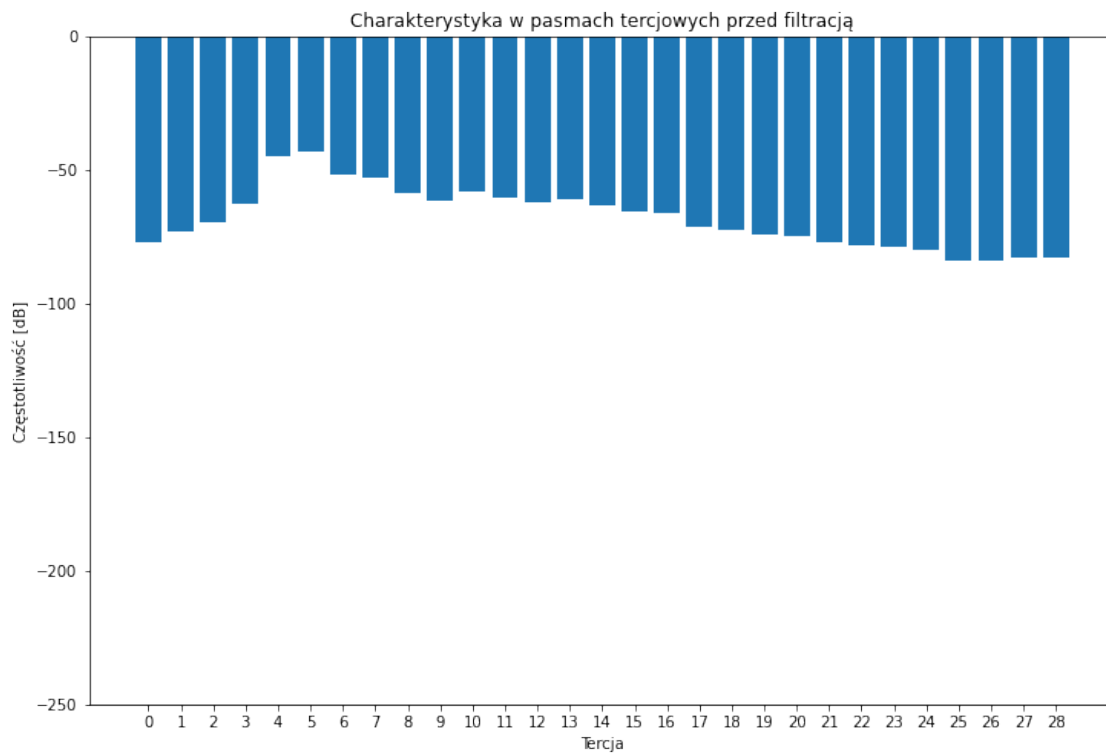
Źródło: [https://www.wikiwand.com/pl/Tercja_\(akustyka\)](https://www.wikiwand.com/pl/Tercja_(akustyka))

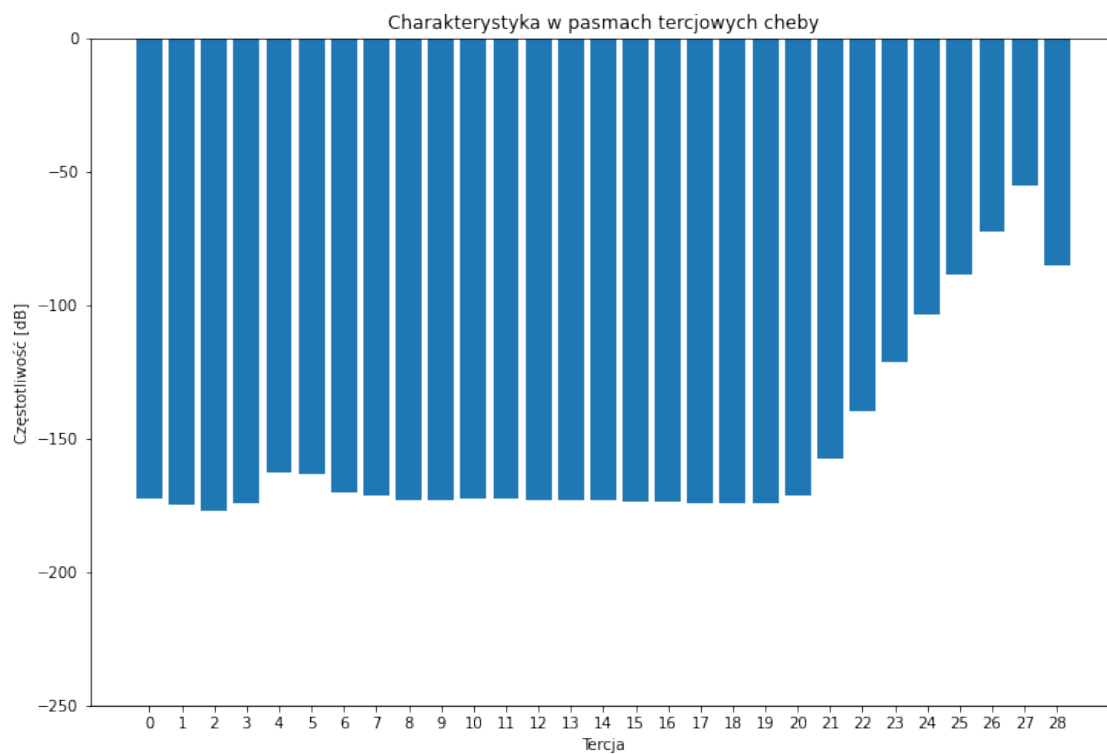
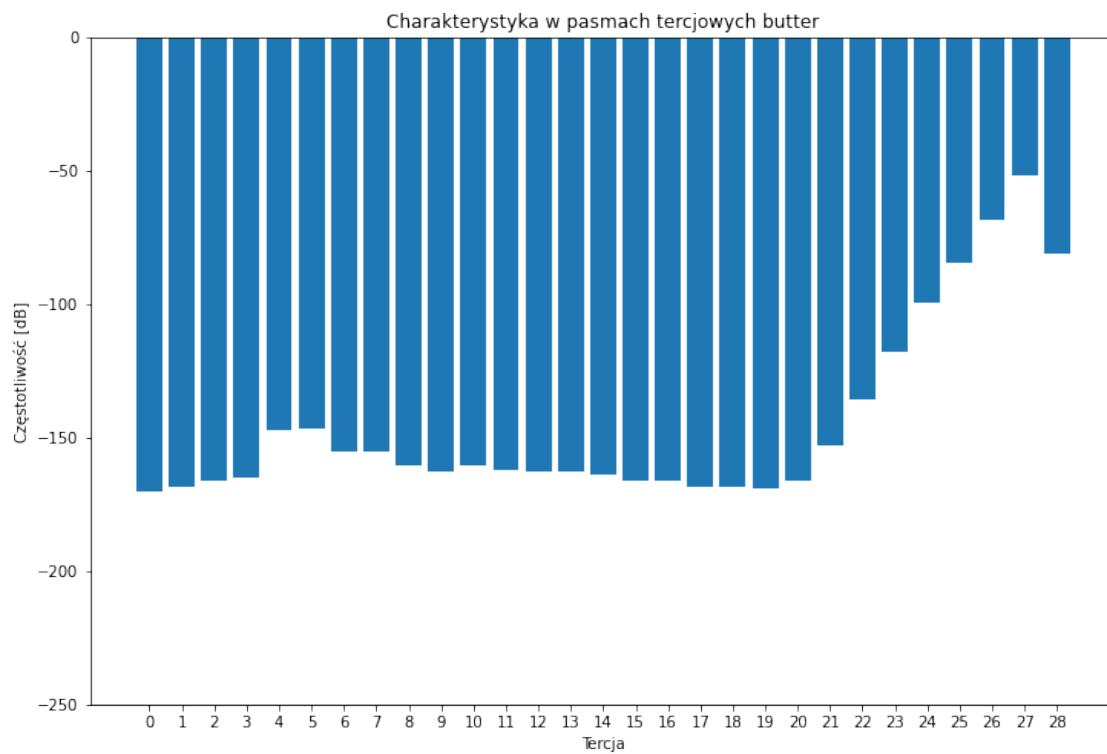
```
[118]: def plot_tert(y, fs, title, verbose=False):
    # Widmo
    widmo_amp = 20 * np.log10(np.abs(np.fft.rfft(y)) / (len(y)//2))
    # Oś X - częstotliwości
    f = np.fft.rfftfreq(len(y), 1/fs)
    # Częstotliwości środkowe do oktaw
    czestotliwosci_srodkowe = [12.5, 16, 20, 25, 31.5, 40, 50, 63, 80, 100,
    ↪125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500,
    ↪3150, 4000, 5000, 6300, 80000]
    srednia_z_pasma = []
    for idx, f_o in enumerate(czestotliwosci_srodkowe):
        f_d = f_o * np.power(2, -1/6)
        f_g = f_o * np.power(2, 1/6)
        if verbose:
            print("Tercja:", idx+1, "częstotliwość dolna:", f_d, "środkowa:",
    ↪f_o, "górna:", f_g)
        lower_bound = np.argmax(f>f_d)
        upper_bound = np.argmax(f>f_g)
        if upper_bound == 0:
            upper_bound = -1
        srednia_z_pasma.append(np.mean(widmo_amp[lower_bound:upper_bound]))
    plt.figure(figsize=(12, 8))
    plt.bar(np.arange(29), srednia_z_pasma)
    plt.ylim([-250, 0])
    plt.xticks(np.arange(29))
    plt.xlabel("Tercja")
    plt.ylabel("Częstotliwość [dB]")
    plt.title(f"Charakterystyka w pasmach tercjowych {title}")
    plt.show()

plot_tert(y, sr, "przed filtracją", verbose=True)
plot_tert(filtered_y_butter, sr, "butter")
plot_tert(filtered_y_cheby1, sr, "cheby")
```

Tercja: 1 częstotliwość dolna: 11.13623397675424 środkowa: 12.5 górna:
 14.030775603867163
 Tercja: 2 częstotliwość dolna: 14.254379490245428 środkowa: 16 górna:
 17.95939277294997
 Tercja: 3 częstotliwość dolna: 17.817974362806787 środkowa: 20 górna:
 22.44924096618746
 Tercja: 4 częstotliwość dolna: 22.27246795350848 środkowa: 25 górna:
 28.061551207734325
 Tercja: 5 częstotliwość dolna: 28.063309621420686 środkowa: 31.5 górna:
 35.35755452174525
 Tercja: 6 częstotliwość dolna: 35.635948725613574 środkowa: 40 górna:
 44.89848193237492
 Tercja: 7 częstotliwość dolna: 44.54493590701696 środkowa: 50 górna:
 56.12310241546865
 Tercja: 8 częstotliwość dolna: 56.12661924284137 środkowa: 63 górna:
 70.7151090434905
 Tercja: 9 częstotliwość dolna: 71.27189745122715 środkowa: 80 górna:
 89.79696386474984
 Tercja: 10 częstotliwość dolna: 89.08987181403393 środkowa: 100 górna:
 112.2462048309373
 Tercja: 11 częstotliwość dolna: 111.36233976754241 środkowa: 125 górna:
 140.30775603867164
 Tercja: 12 częstotliwość dolna: 142.5437949024543 środkowa: 160 górna:
 179.59392772949968
 Tercja: 13 częstotliwość dolna: 178.17974362806785 środkowa: 200 górna:
 224.4924096618746
 Tercja: 14 częstotliwość dolna: 222.72467953508482 środkowa: 250 górna:
 280.6155120773433
 Tercja: 15 częstotliwość dolna: 280.6330962142069 środkowa: 315 górna:
 353.5755452174525
 Tercja: 16 częstotliwość dolna: 356.3594872561357 środkowa: 400 górna:
 448.9848193237492
 Tercja: 17 częstotliwość dolna: 445.44935907016963 środkowa: 500 górna:
 561.2310241546866
 Tercja: 18 częstotliwość dolna: 561.2661924284138 środkowa: 630 górna:
 707.151090434905
 Tercja: 19 częstotliwość dolna: 712.7189745122714 środkowa: 800 górna:
 897.9696386474984
 Tercja: 20 częstotliwość dolna: 890.8987181403393 środkowa: 1000 górna:
 1122.4620483093731
 Tercja: 21 częstotliwość dolna: 1113.623397675424 środkowa: 1250 górna:
 1403.0775603867162
 Tercja: 22 częstotliwość dolna: 1425.4379490245428 środkowa: 1600 górna:
 1795.9392772949968
 Tercja: 23 częstotliwość dolna: 1781.7974362806785 środkowa: 2000 górna:
 2244.9240966187463
 Tercja: 24 częstotliwość dolna: 2227.246795350848 środkowa: 2500 górna:
 2806.1551207734324

Tercja: 25 częstotliwość dolna: 2806.3309621420685 środkowa: 3150 górna: 3535.755452174525
Tercja: 26 częstotliwość dolna: 3563.594872561357 środkowa: 4000 górna: 4489.8481932374925
Tercja: 27 częstotliwość dolna: 4454.493590701696 środkowa: 5000 górna: 5612.310241546865
Tercja: 28 częstotliwość dolna: 5612.661924284137 środkowa: 6300 górna: 7071.51090434905
Tercja: 29 częstotliwość dolna: 71271.89745122714 środkowa: 80000 górna: 89796.96386474984

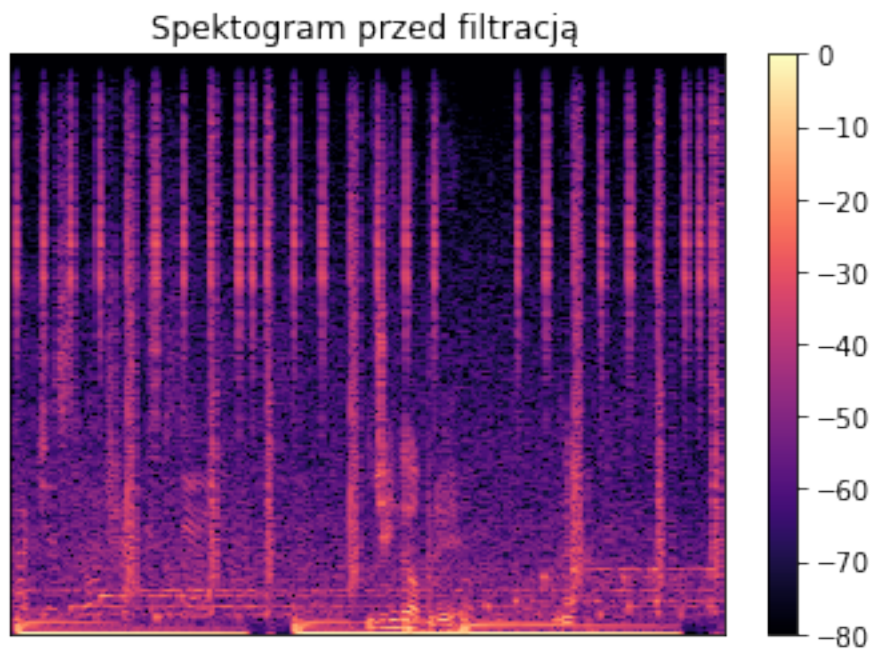




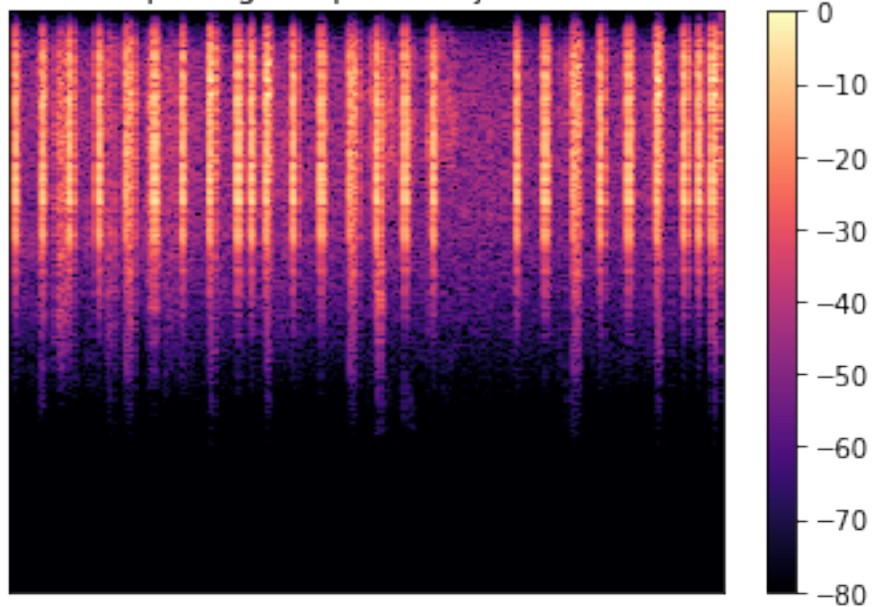
1.7 Spektrogram

```
[119]: def spectrogram(y):  
        D = librosa.stft(y) # STFT of y  
        S_db = librosa.amplitude_to_db(np.abs(D), ref=np.max)  
        plt.figure()  
        librosa.display.specshow(S_db)  
        plt.colorbar()
```

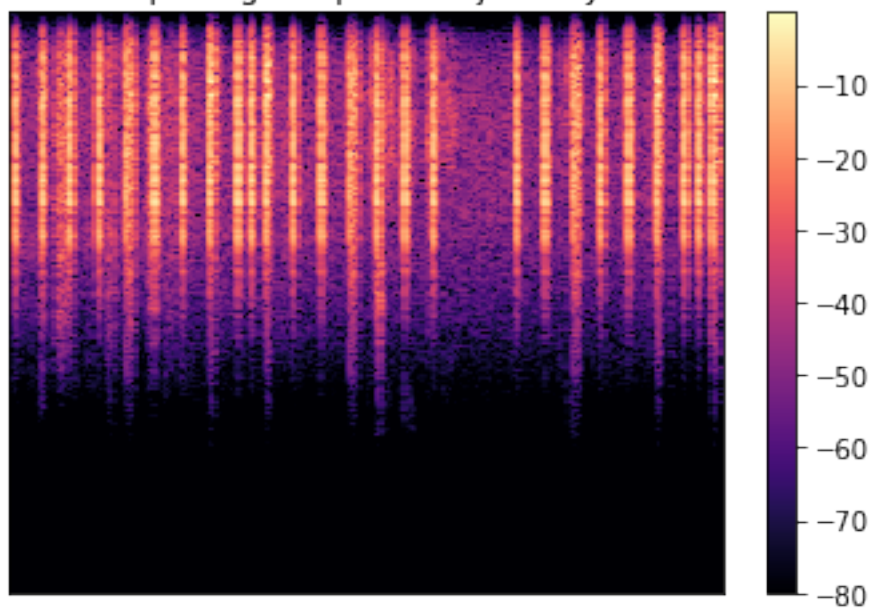
```
[120]: spectrogram(y)  
        plt.title("Spektrogram przed filtracją")  
        plt.show()  
  
        spectrogram(filtered_y_butter)  
        plt.title("Spektrogram po filtracji butter")  
        plt.show()  
  
        spectrogram(filtered_y_cheby1)  
        plt.title("Spektrogram po filtracji cheby")  
        plt.show()
```



Spektrogram po filtracji butter



Spektrogram po filtracji cheby



1.8 Wnioski

Podczas zajęć zapoznano się z filtrami NOI. Przetestowano działanie filtra dolnozaporowego Czebyszewa oraz Butterwortha.

Na testowanym sygnale faktycznie po filtracji nie słychać niskich częstotliwości. Ciężko porównać jakość uzyskanej filtracji z obu filtrów na podstawie wygenerowanych sygnałów. Na podstawie wykresów z charakterystyk częstotliwościowych można powiedzieć, że filtr Butterwortha ma mniej zakłóceń (falowań) już po osiągnięciu pasma, które ma zasadniczo być filtrowane. Wydaje mi się, że jest to uzyskane kosztem mniej idealnego stanu przejściowego.

Na podstawie tabel z wikipedii wygenerowano wykresy oktauwowe i tercjowe dla uzyskanych sygnałów.