Bogumił Wierzchowski 291498

Analiza sygnału w dziedzinie czasu i częstości

Ćwiczenie 5: Transformata falkowa

1. **Wstęp**

Celem ćwiczenia jest wprowadzenie w analizę czas-częstość przy użyciu transformaty falkowej. Jako przykład sygnału niestacjonarnego wykorzystamy sygnał mowy oraz sygnał syntetyczny.

Użyte oprogramowanie: *Python ver. 3.9.7*

Użyte biblioteki: *numpy, scipy, matplotlib, ssqueezepy, math*

1. **Kod źródłowy**

import numpy as np

import matplotlib.pyplot as plt

import scipy.signal as sig

import scipy.io

import math

from ssqueezepy import cwt

from ssqueezepy.visuals import plot, imshow

from scipy.io import wavfile

plt.rcParams["figure.figsize"] = [15, 6]

plt.rcParams['font.size'] = '13'

f\_d = [.038580777748, .126969125396, -.077161555496, -.607491641386, .745687558934, -.226584265197]

x = np.arange(1, len(f\_d)+1)

f\_g = -((-1)\*\*x)\*f\_d

fig, (ax1, ax2) = plt.subplots(1, 2)

ax1.plot(x,f\_d)

ax1.set(xlabel='czas [a.u.]', ylabel='amplituda [a.u.]', title='Filtr dolnoprzepustowy')

ax2.plot(x, f\_g)

ax2.set(xlabel='czas [a.u.]', ylabel='amplituda [a.u.]', title='Filtr górnoprzepustowy')

plt.show()

N = 512

fs = 500

dt = 1/fs

t = np.arange(N)\*dt

f1 = 20

f2 = 100

signal = sig.chirp(t, f1, t[-1], f2)

plt.plot(signal)

plt.xlabel('nr próbki')

plt.ylabel('amplituda [a.u.]')

plt.title('Sygnał świergotowy, liniowy 20-100 Hz')

plt.show()

Wx, scales = cwt(signal, 'morlet')

imshow(Wx, yticks=scales, abs=1,

       title="Skalogram sygnału świergotowego",

       ylabel="skala", xlabel="nr.próbki")

for i in range(int(math.log2(N))):

    splot\_d = sig.convolve(signal, f\_d, mode='same')

    splot\_g = sig.convolve(signal, f\_g, mode='same')

    x = np.arange(1, len(splot\_d) + 1)

    fig, (ax1, ax2) = plt.subplots(1, 2)

    fig.suptitle(f'Wynik splotu filtrów z sygnałem świergotowym po {i+1} iteracji')

    ax1.plot(x, splot\_d)

    ax1.set(xlabel='nr próbki', ylabel='amplituda [a.u.]', title='Filtr dolnoprzepustowy')

    ax2.plot(x, splot\_g)

    ax2.set(xlabel='nr próbki', ylabel='amplituda [a.u.]', title='Filtr górnoprzepustowy')

    plt.savefig(f'semestr2/ASwDCiC/lab5/shots/{i+1}.svg')

    plt.close()

    decym\_splot\_d = splot\_d[::2]

    decym\_splot\_g = splot\_g[::2]

    signal = decym\_splot\_d

wav\_fname = 'semestr2/ASwDCiC/lab5/rabarbar8k.wav'

samplerate, signal = wavfile.read(wav\_fname)

length = signal.shape[0] / samplerate

time = np.linspace(0., length, signal.shape[0])

plt.plot(time,signal)

plt.xlabel('czas [s]')

plt.ylabel('amplituda [a.u.]')

plt.title('Sygnał z pliku $\it{rabarbar8k.wav}$')

plt.show()

Wx, scales = cwt(signal, 'morlet')

imshow(Wx, yticks=scales, abs=1,

       title="Skalogram sygnału głosowego",

       ylabel="skala", xlabel="nr.próbki")

for i in range(int(math.log2(signal.shape[0]))):

    splot\_d = sig.convolve(signal, f\_d, mode='same')

    splot\_g = sig.convolve(signal, f\_g, mode='same')

    x = np.arange(1, len(splot\_d) + 1)

    fig, (ax1, ax2) = plt.subplots(1, 2)

    fig.suptitle(f'Wynik splotu filtrów z sygnałem głosowym po {i+1} iteracji')

    ax1.plot(x, splot\_d)

    ax1.set(xlabel='nr próbki', ylabel='amplituda [a.u.]', title='Filtr dolnoprzepustowy')

    ax2.plot(x, splot\_g)

    ax2.set(xlabel='nr próbki', ylabel='amplituda [a.u.]', title='Filtr górnoprzepustowy')

    plt.savefig(f'semestr2/ASwDCiC/lab5/shots/{i+1}.svg')

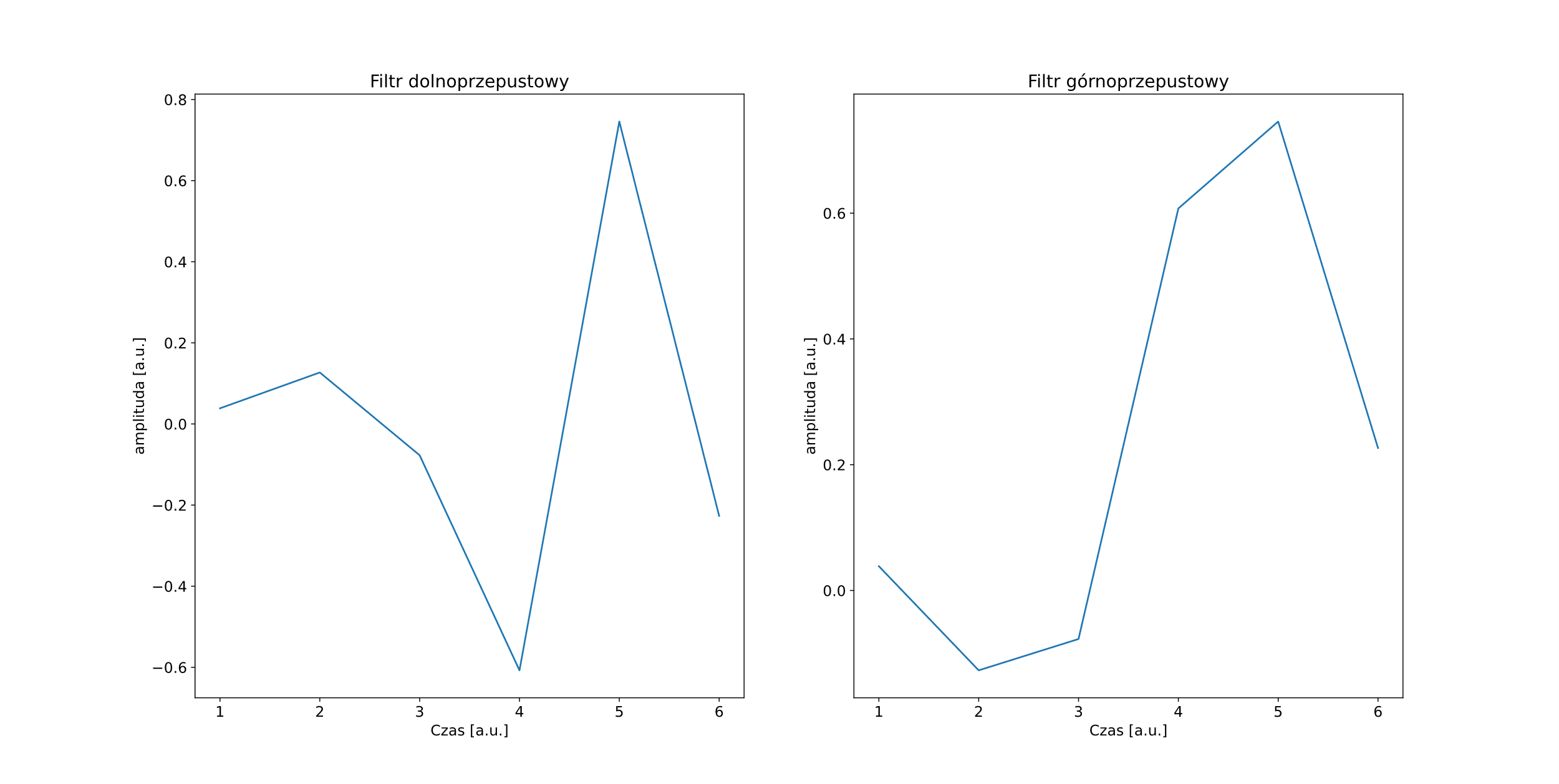
    plt.close()

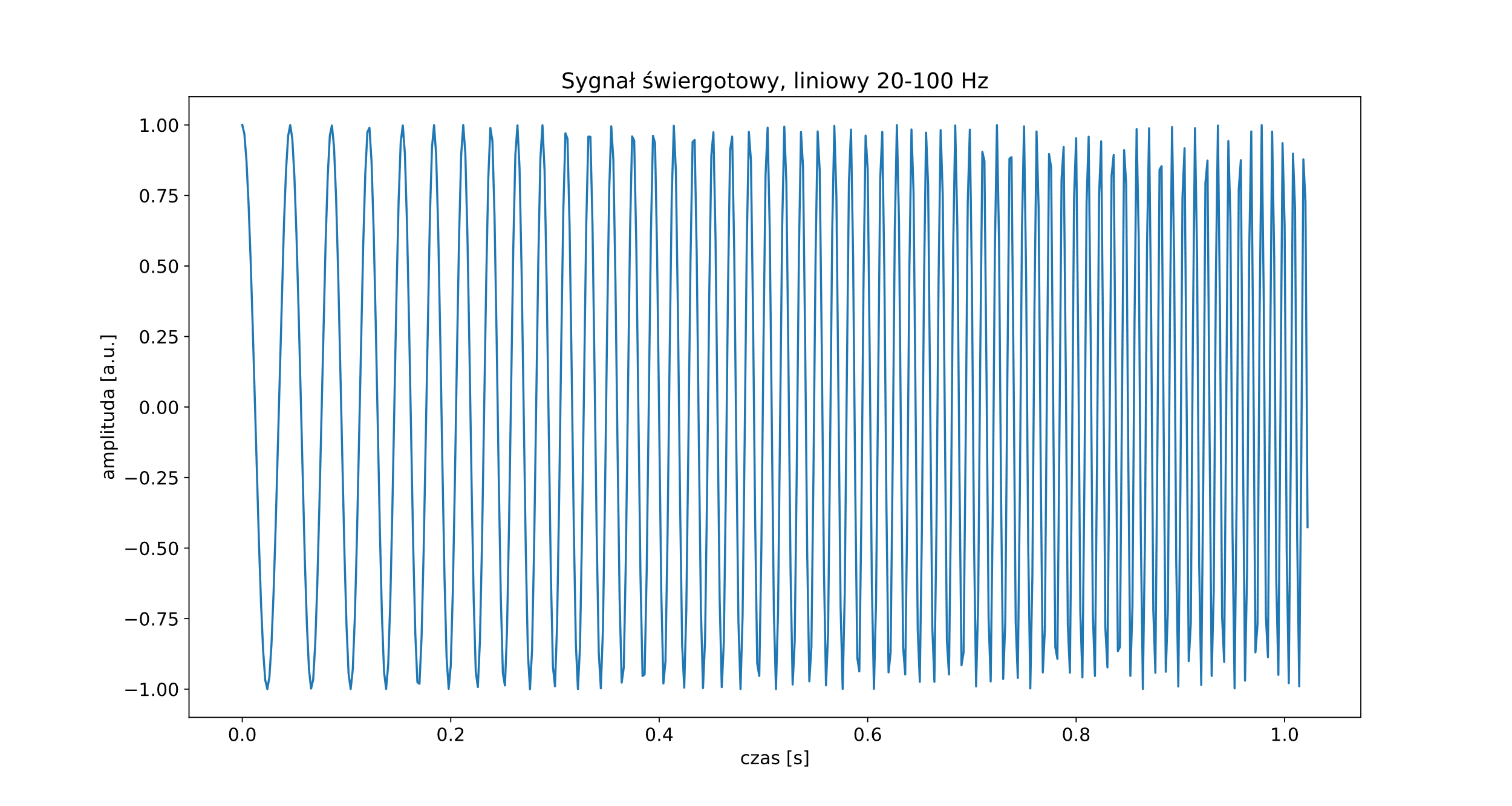
    decym\_splot\_d = splot\_d[::2]

    decym\_splot\_g = splot\_g[::2]

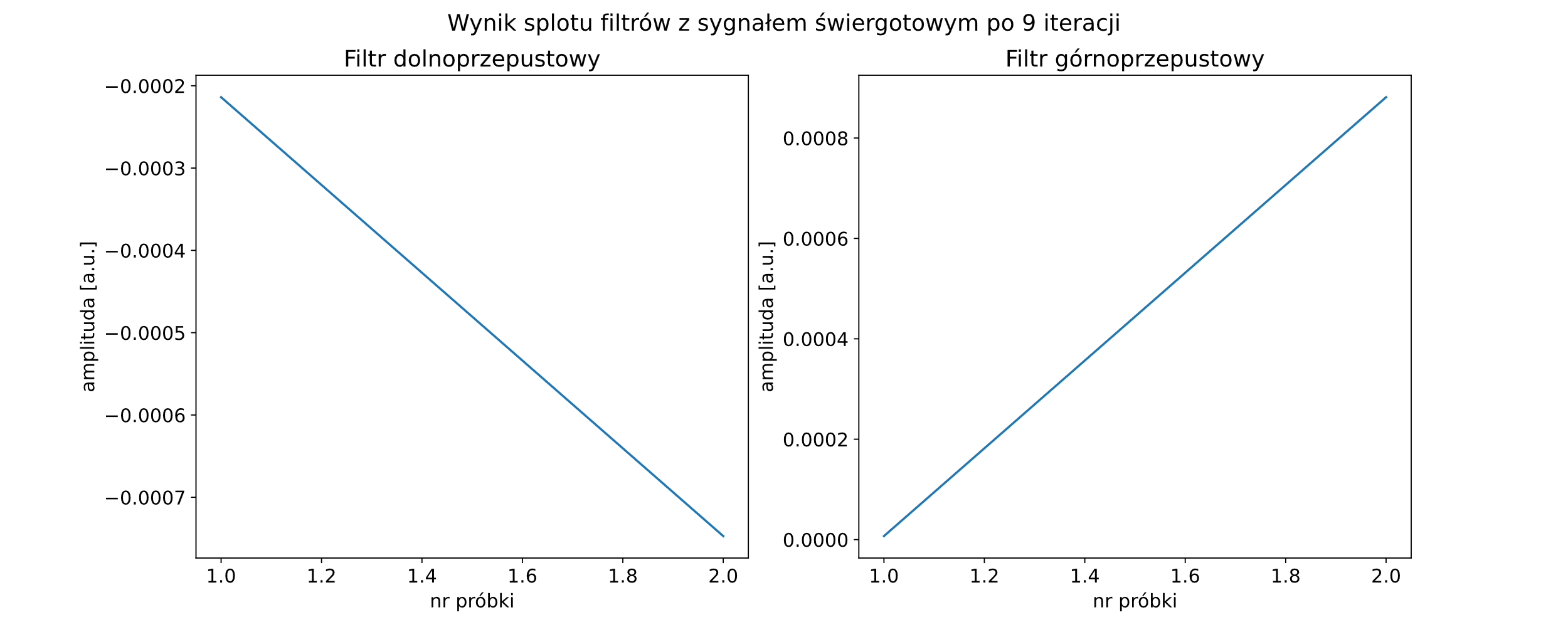
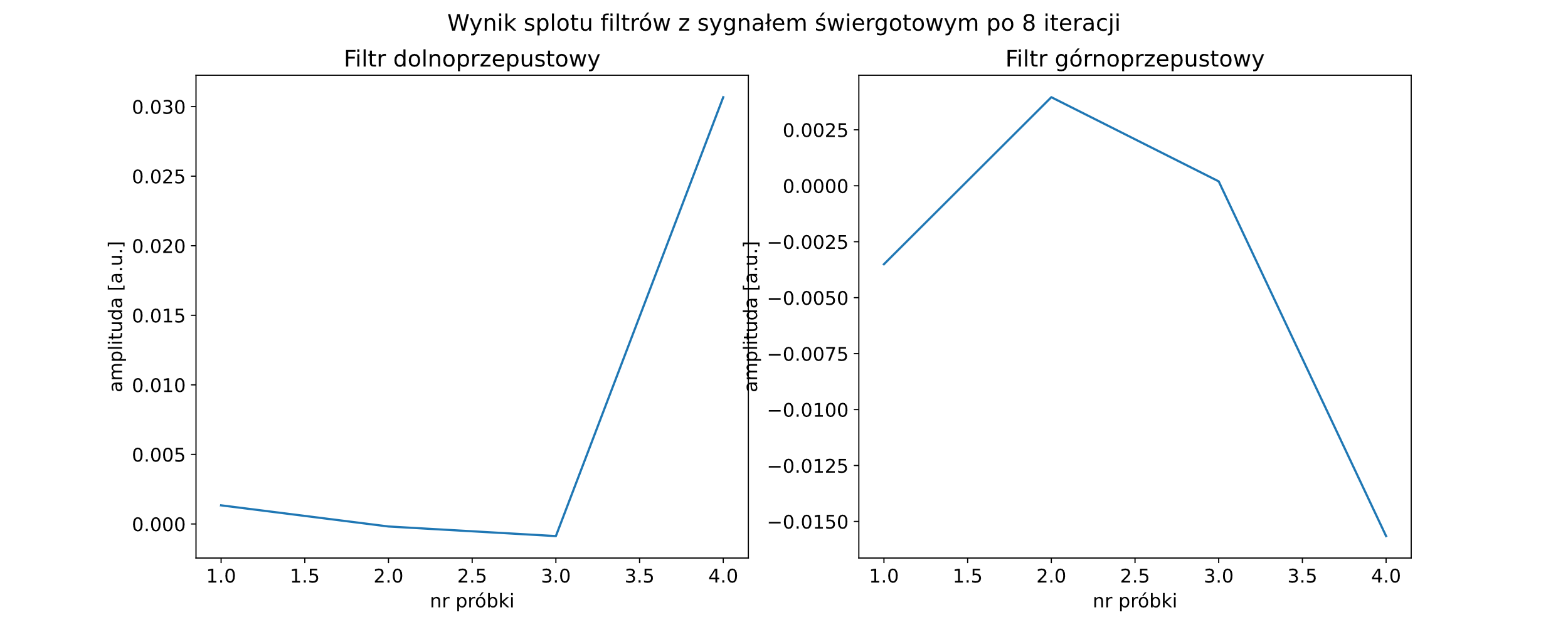
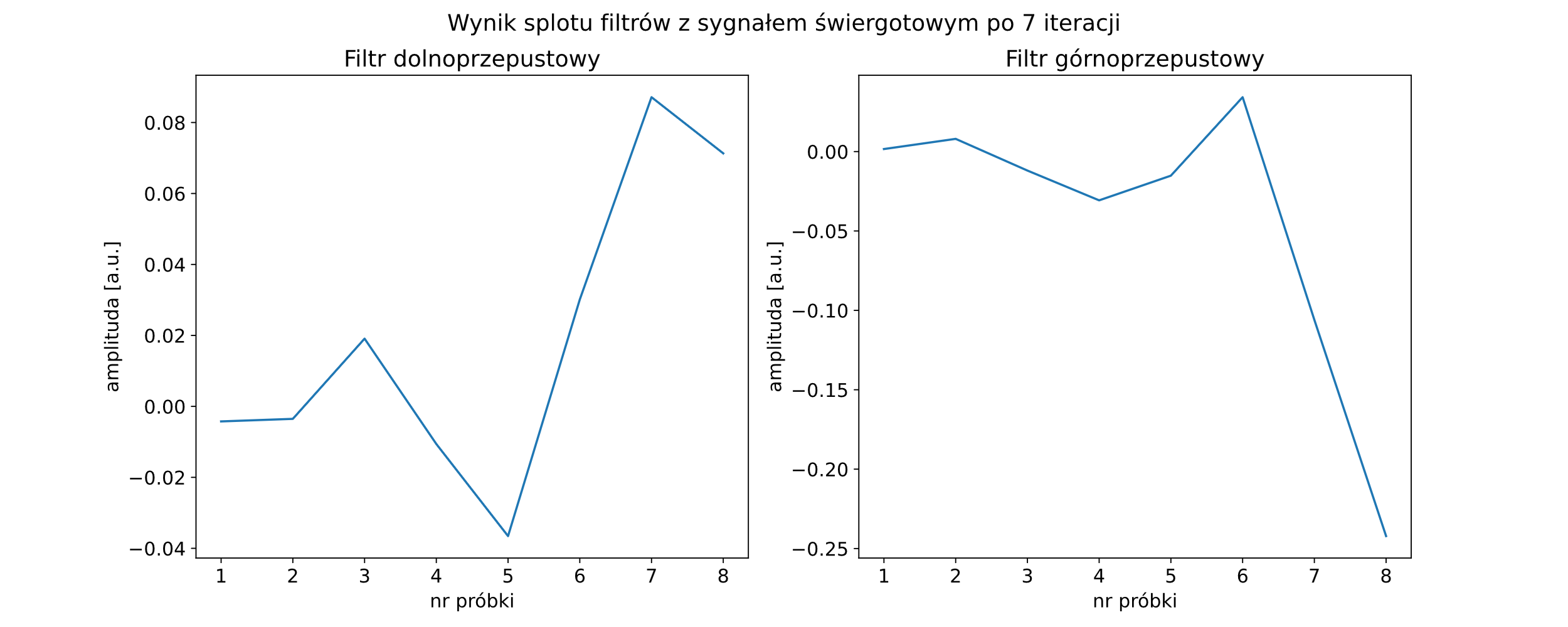
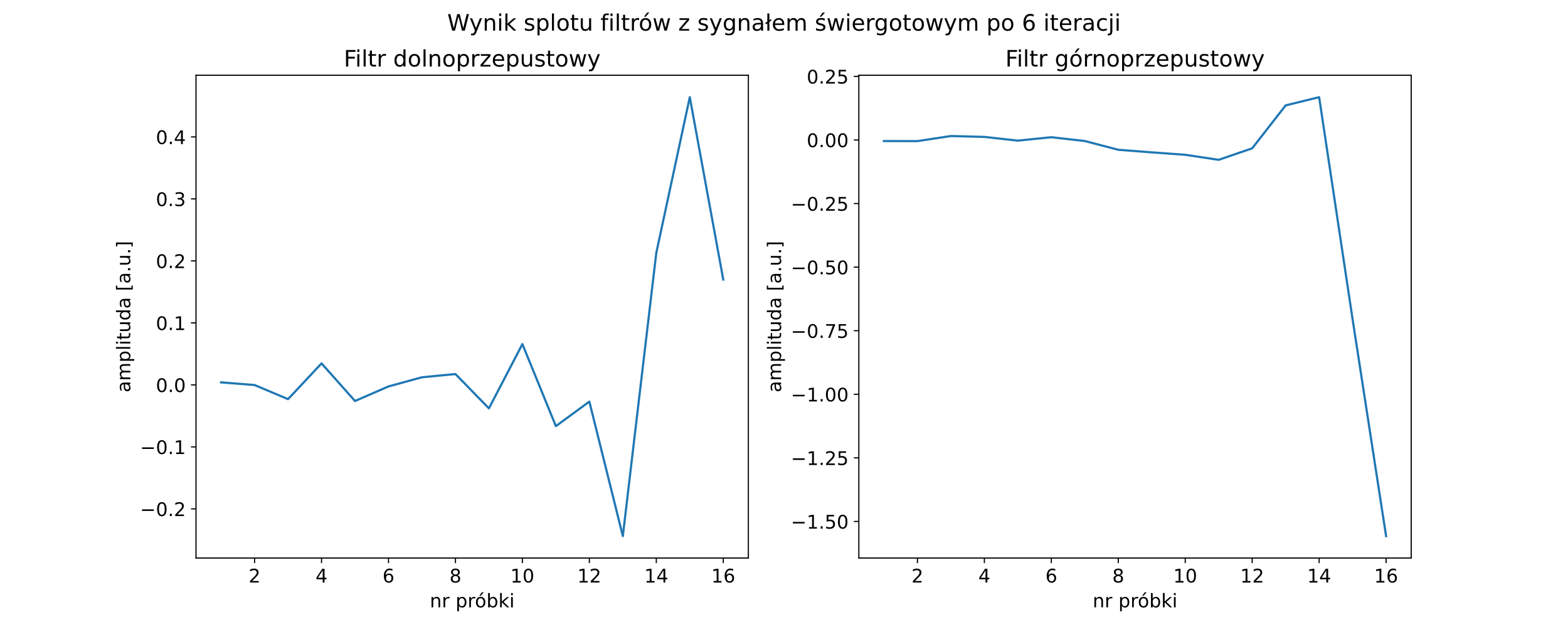
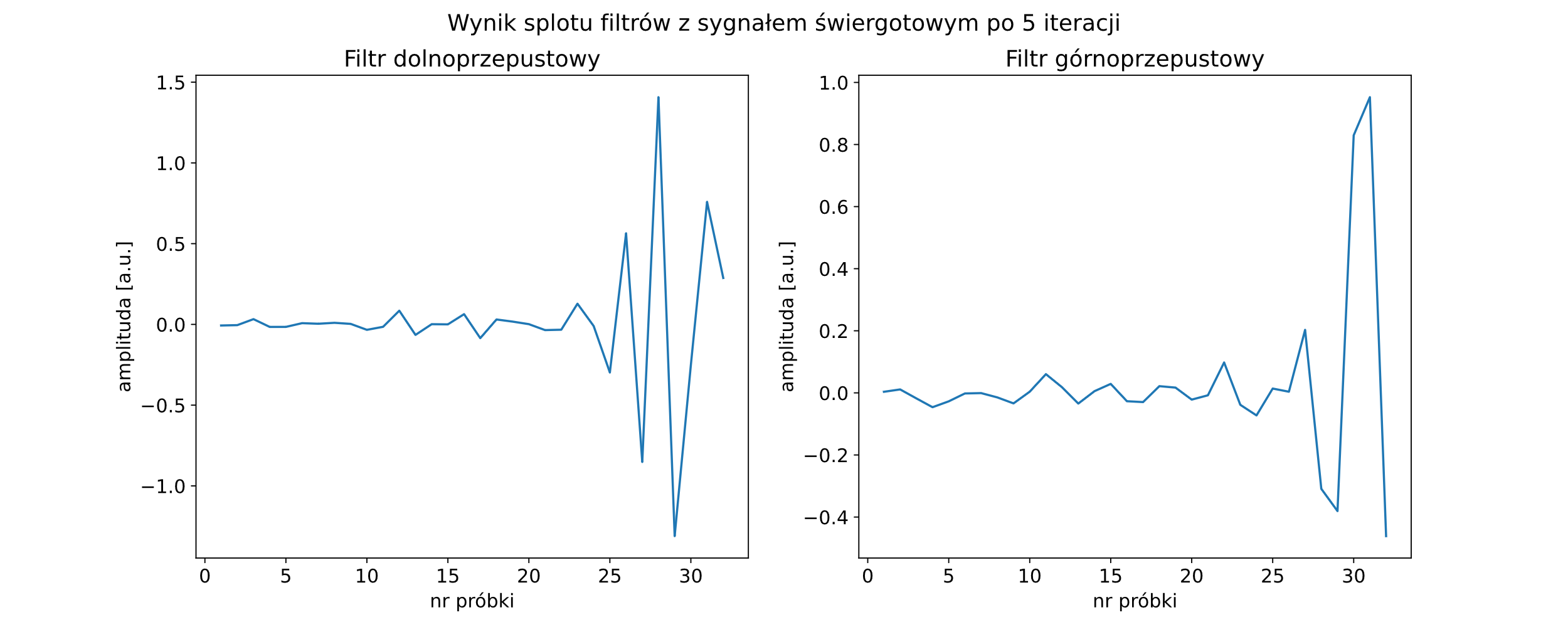
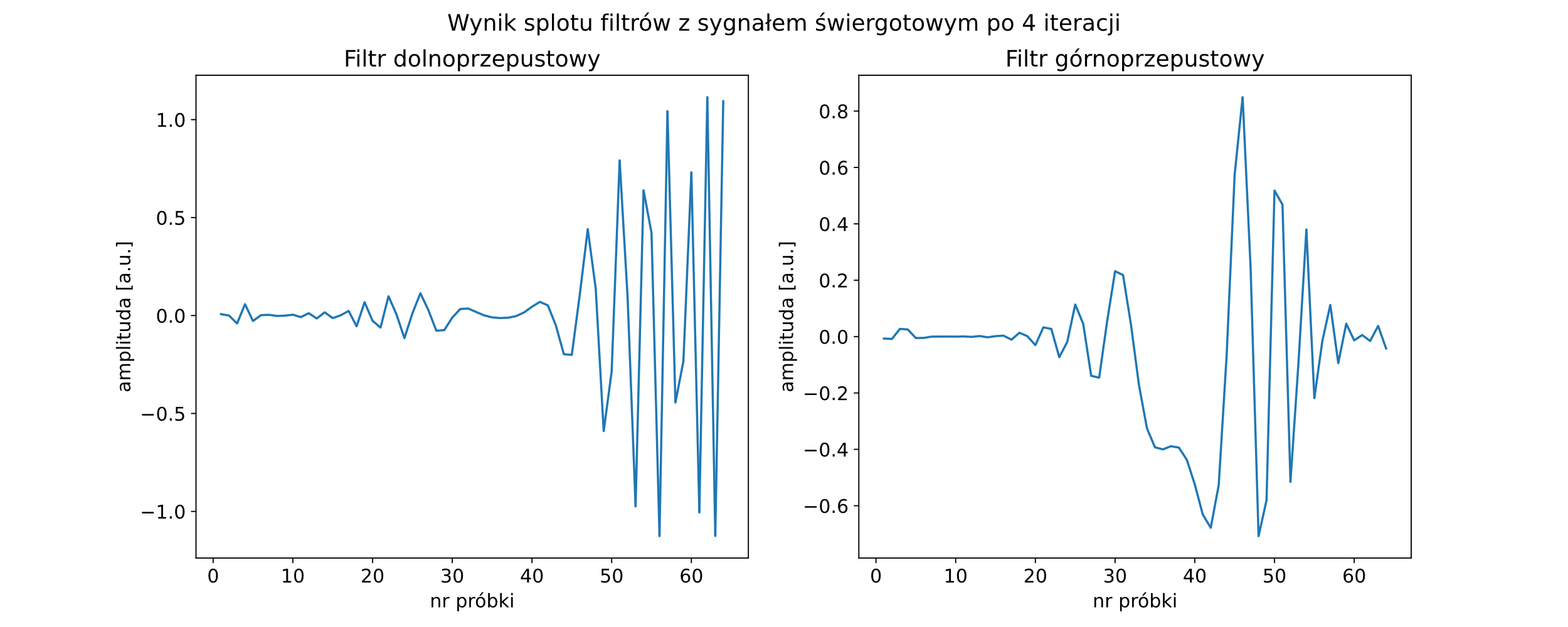
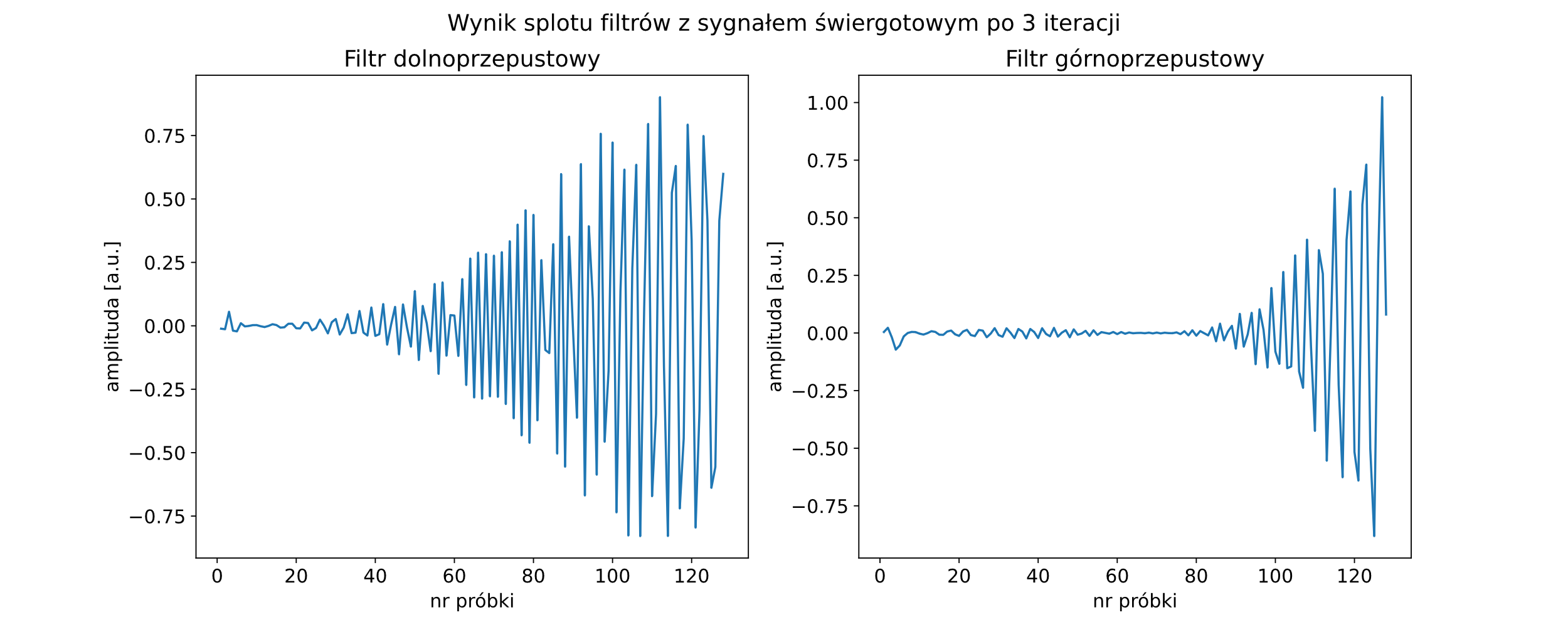
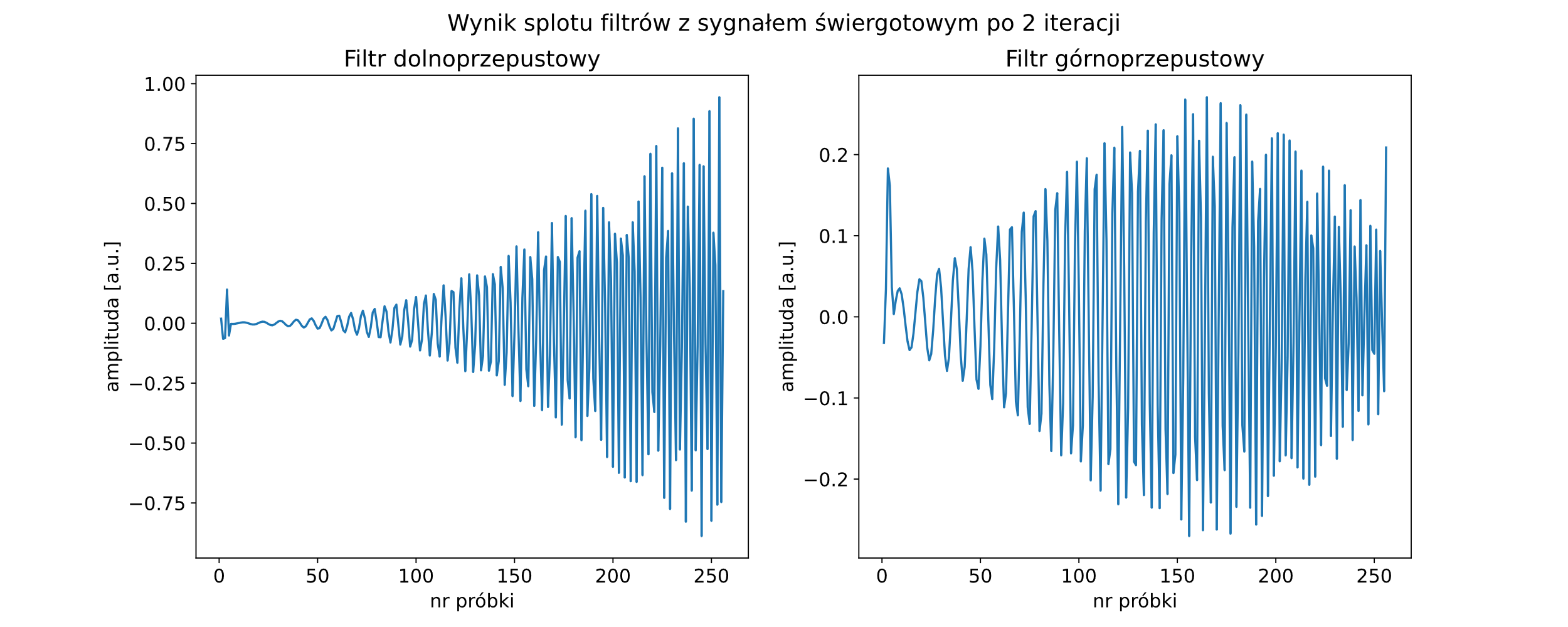
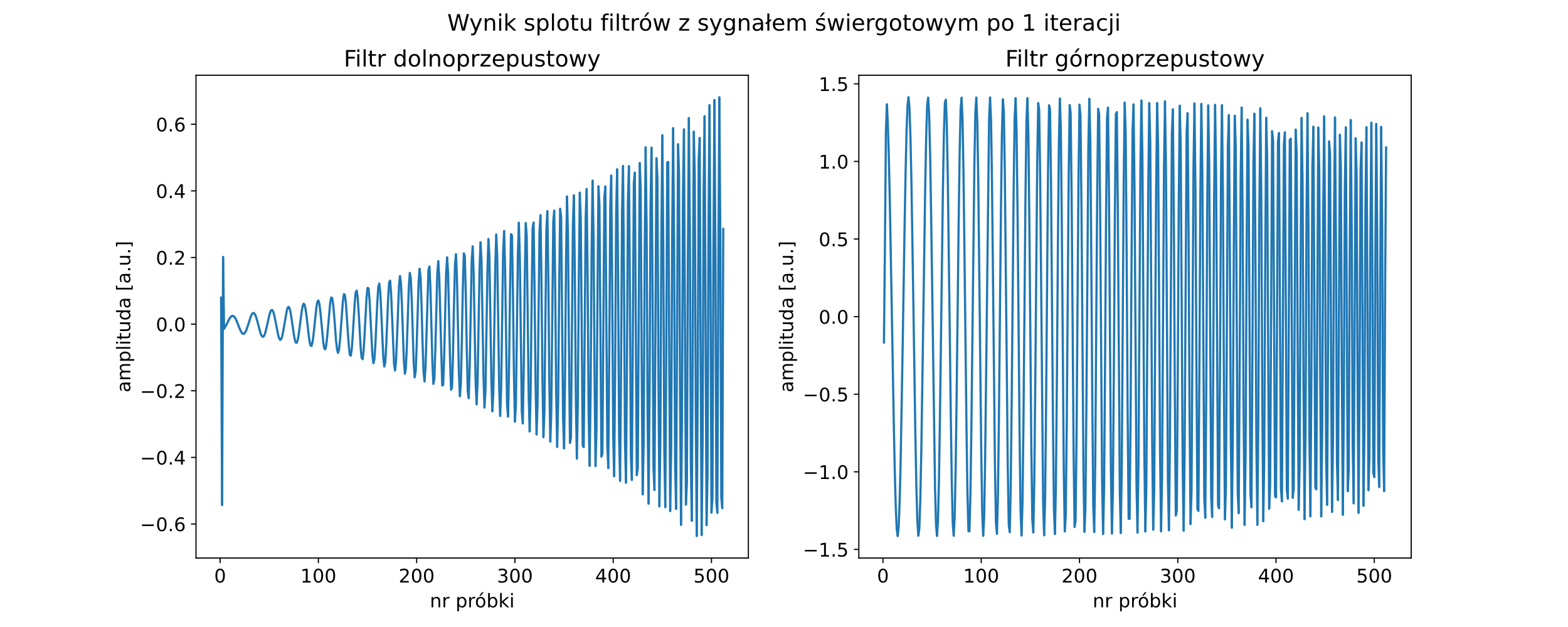
    signal = decym\_splot\_d

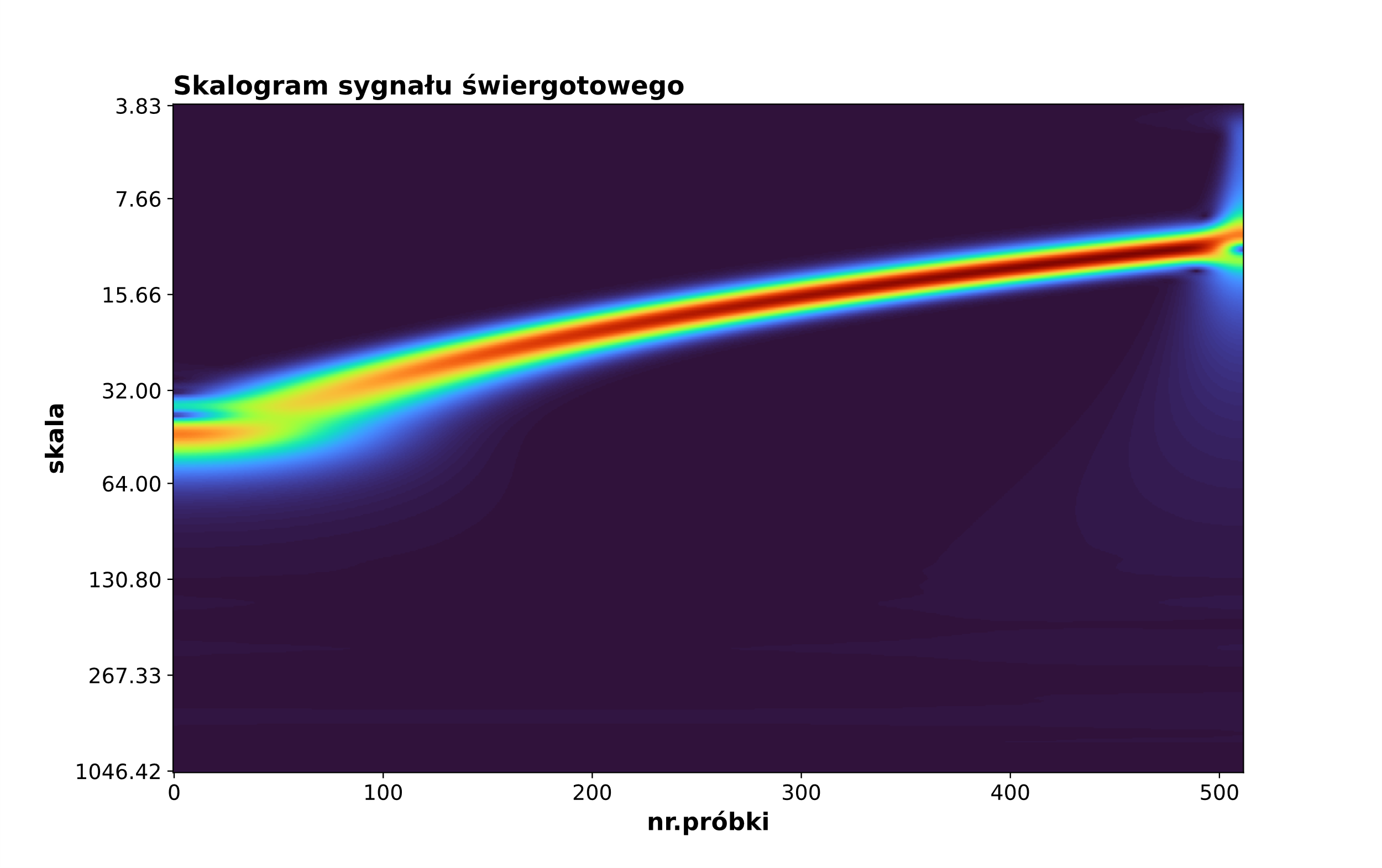
1. **Wyniki**

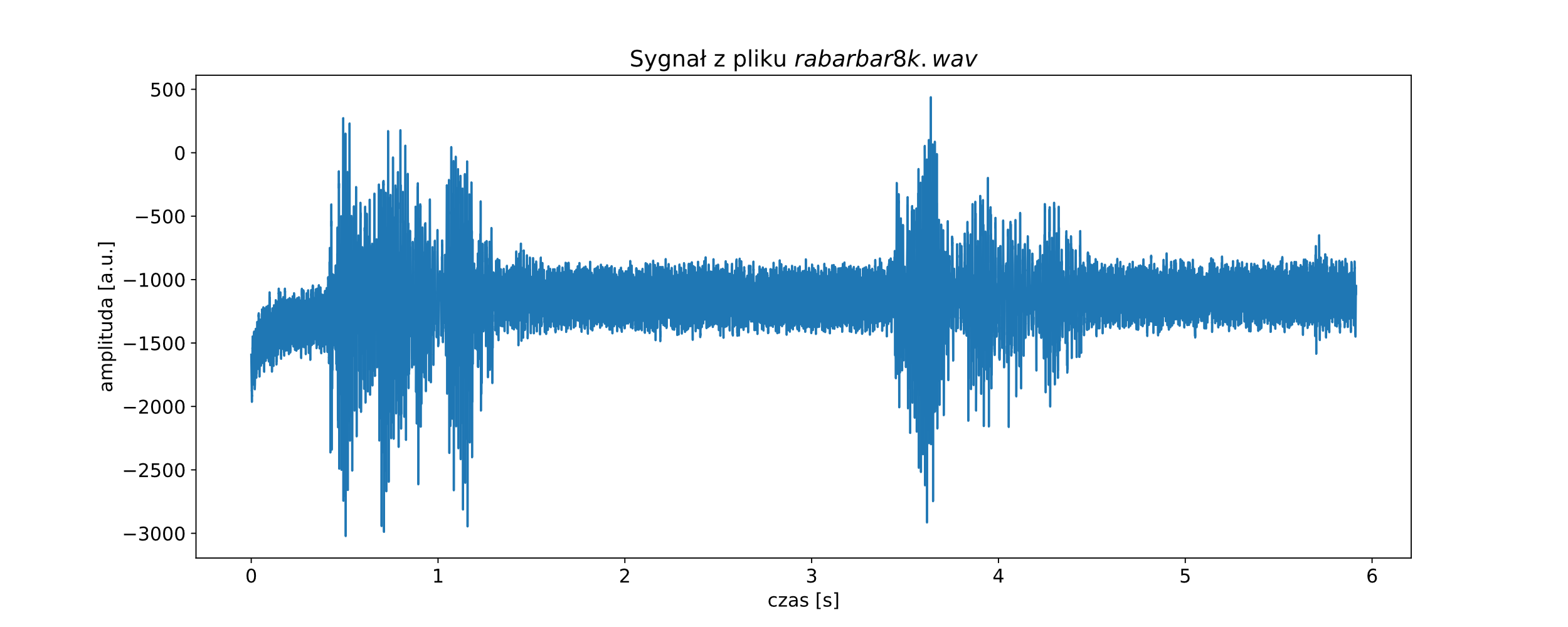
Wykres poniżej przedstawia falkę podstawową typu *Coiflet1* o współczynnikach podanych w instrukcji (lewa storna) oraz jej odbicie zwierciadlane (prawa strona).

Pierwszą częścią ćwiczenia jest przeprowadzenie analizy falkowej na sygnale nietrywialnym, o długości . W tym celu wybrano sygnał świergotowy liniowy 20-100 Hz o długości 512, analizowany na laboratorium nr. 3. Sygnał ten jest przedstawiony na wykresie poniżej.

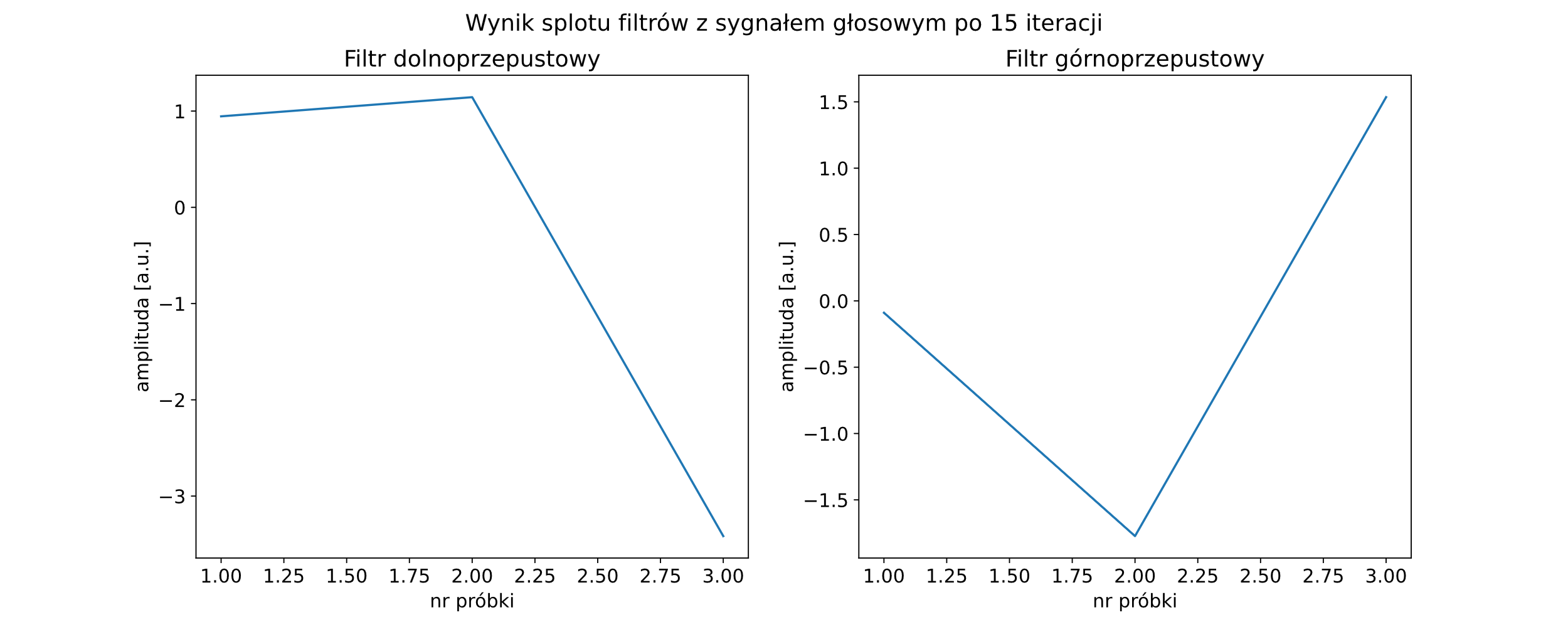
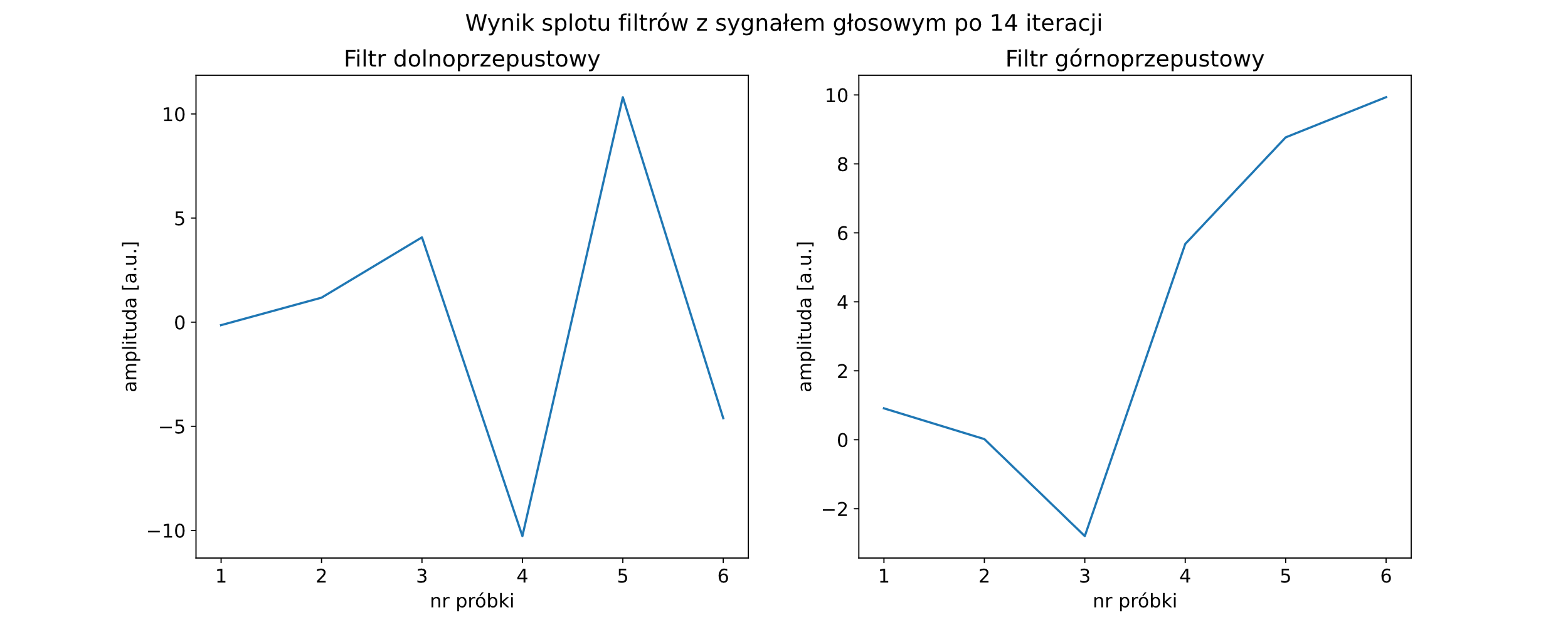
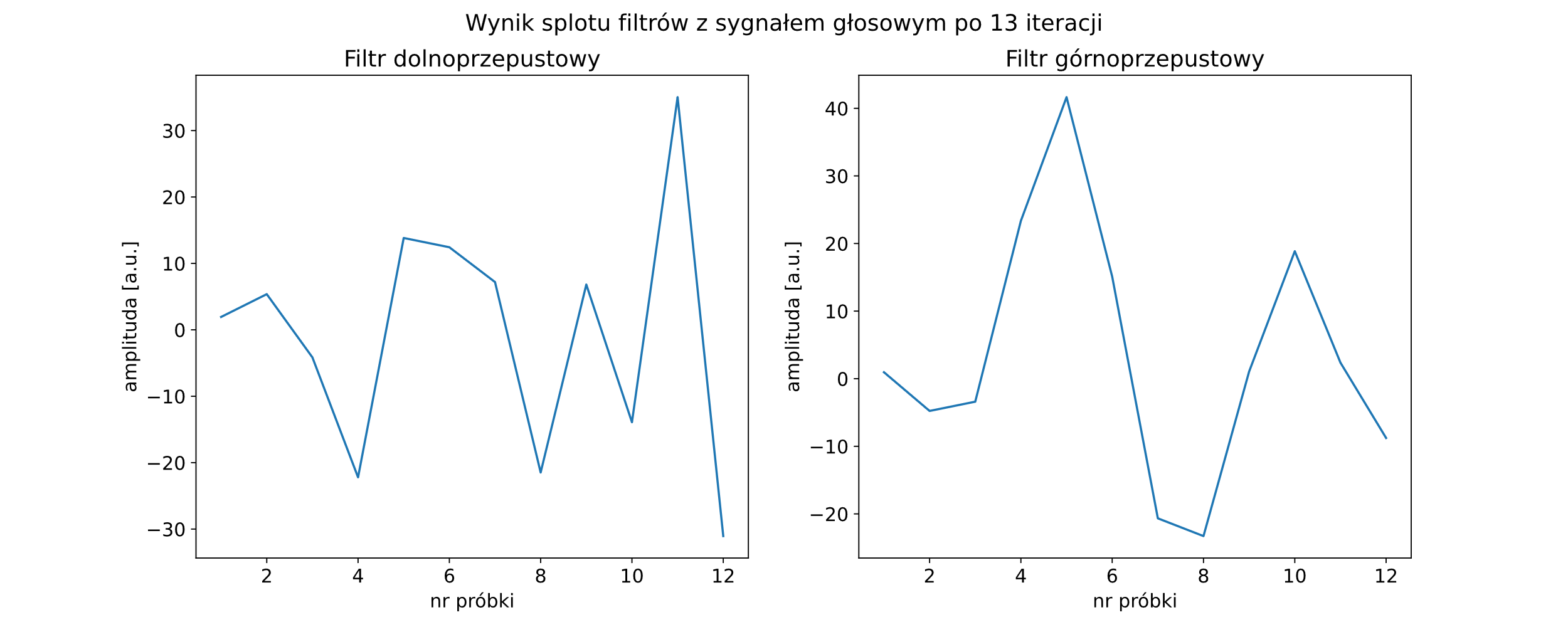
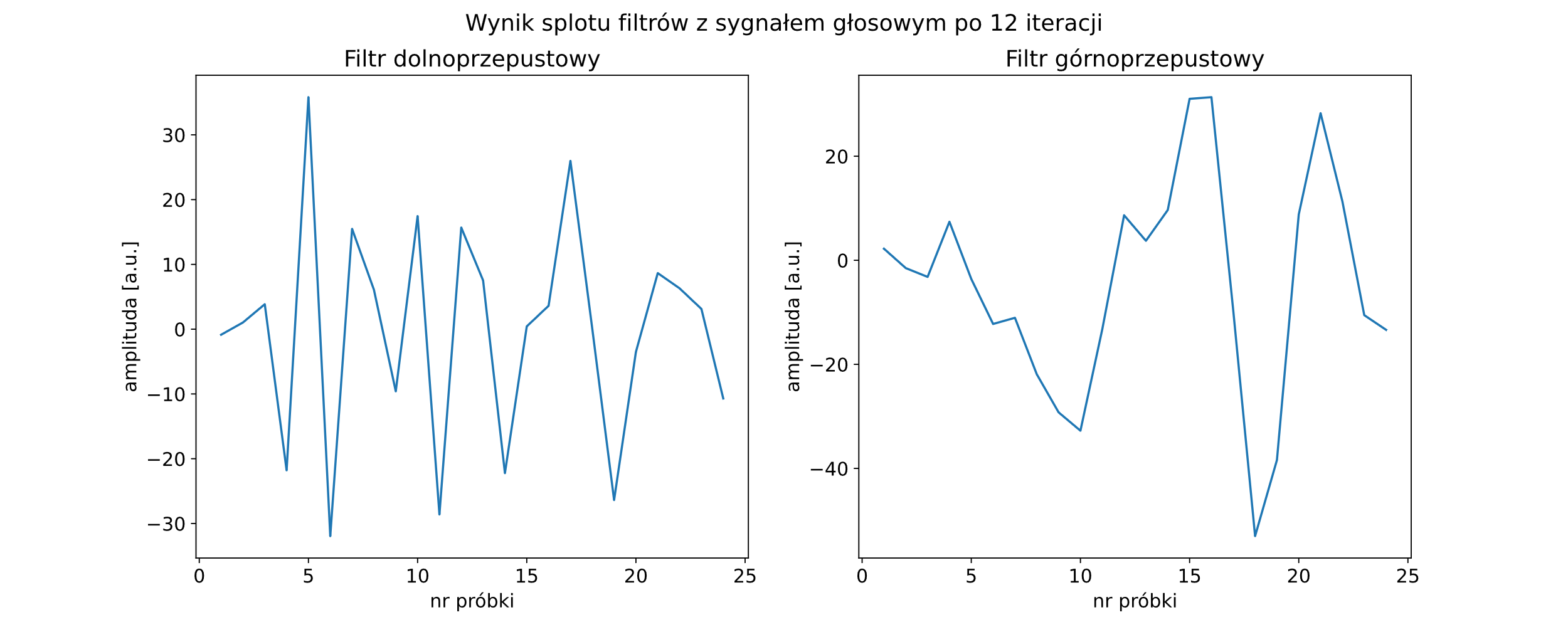
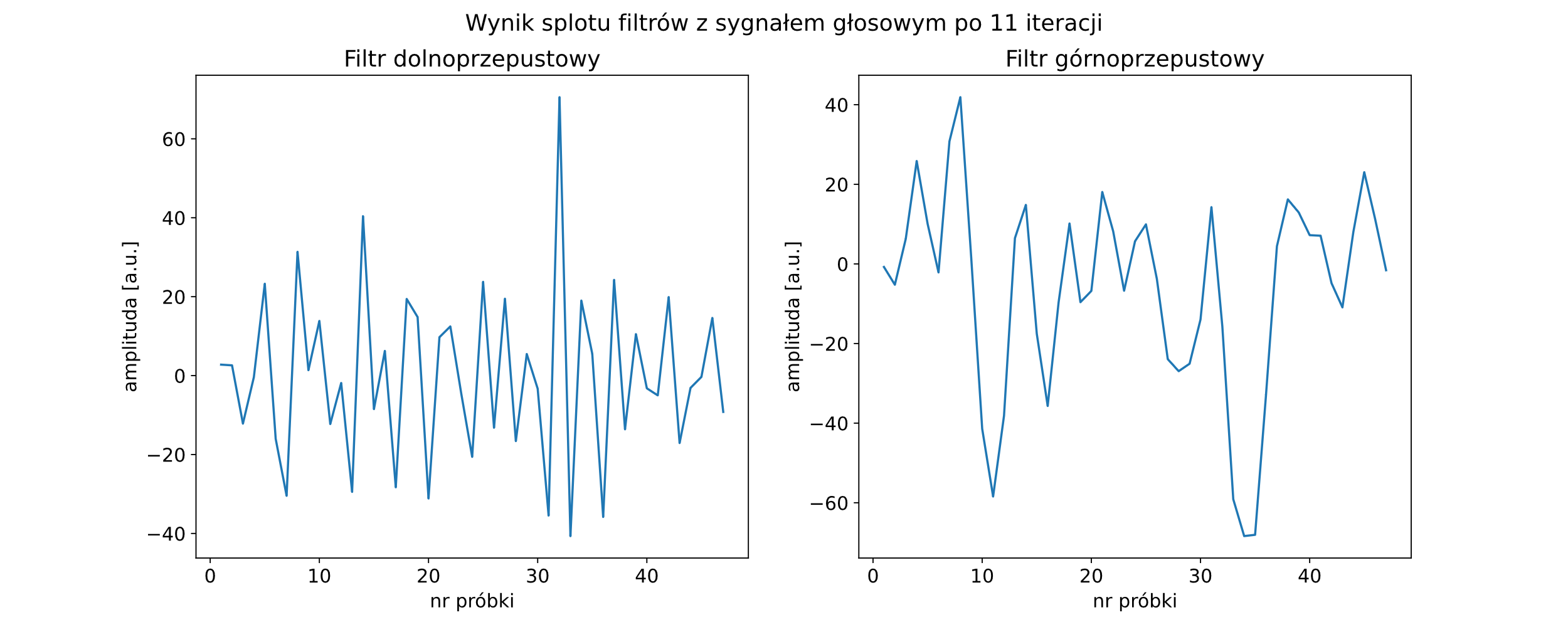
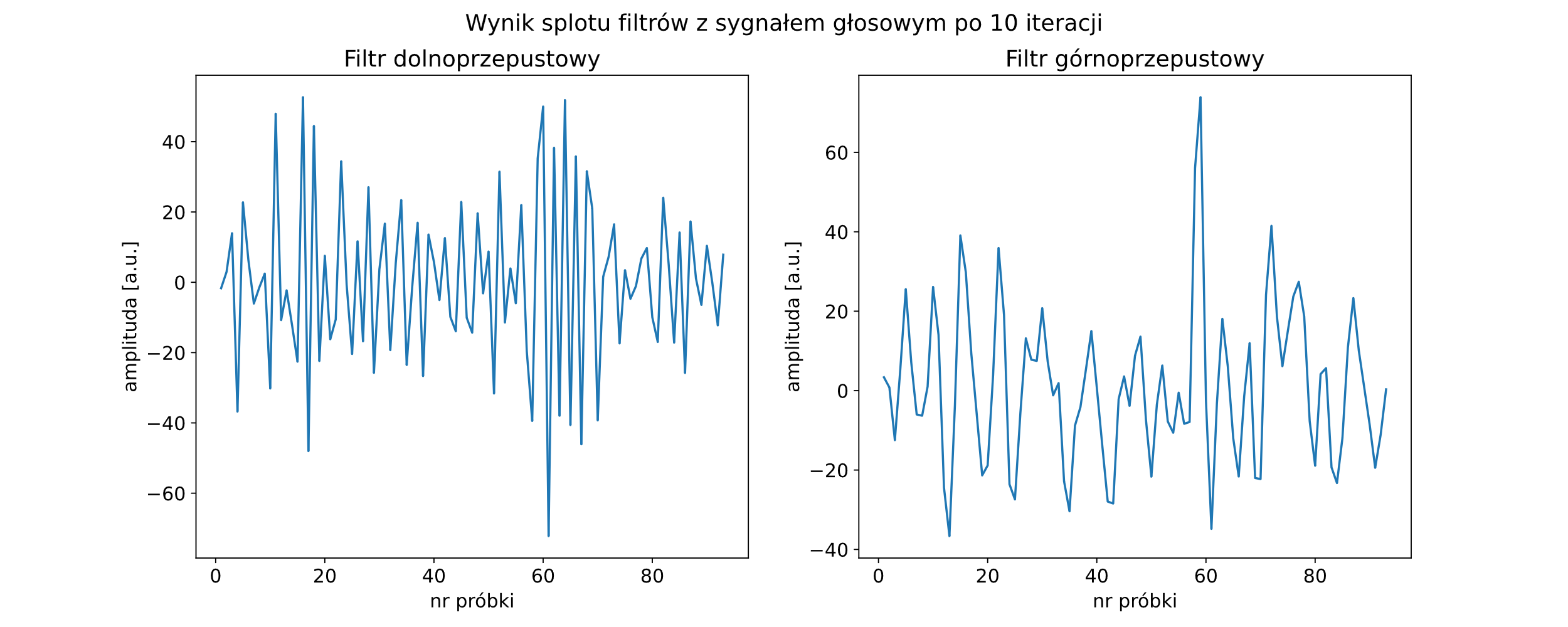
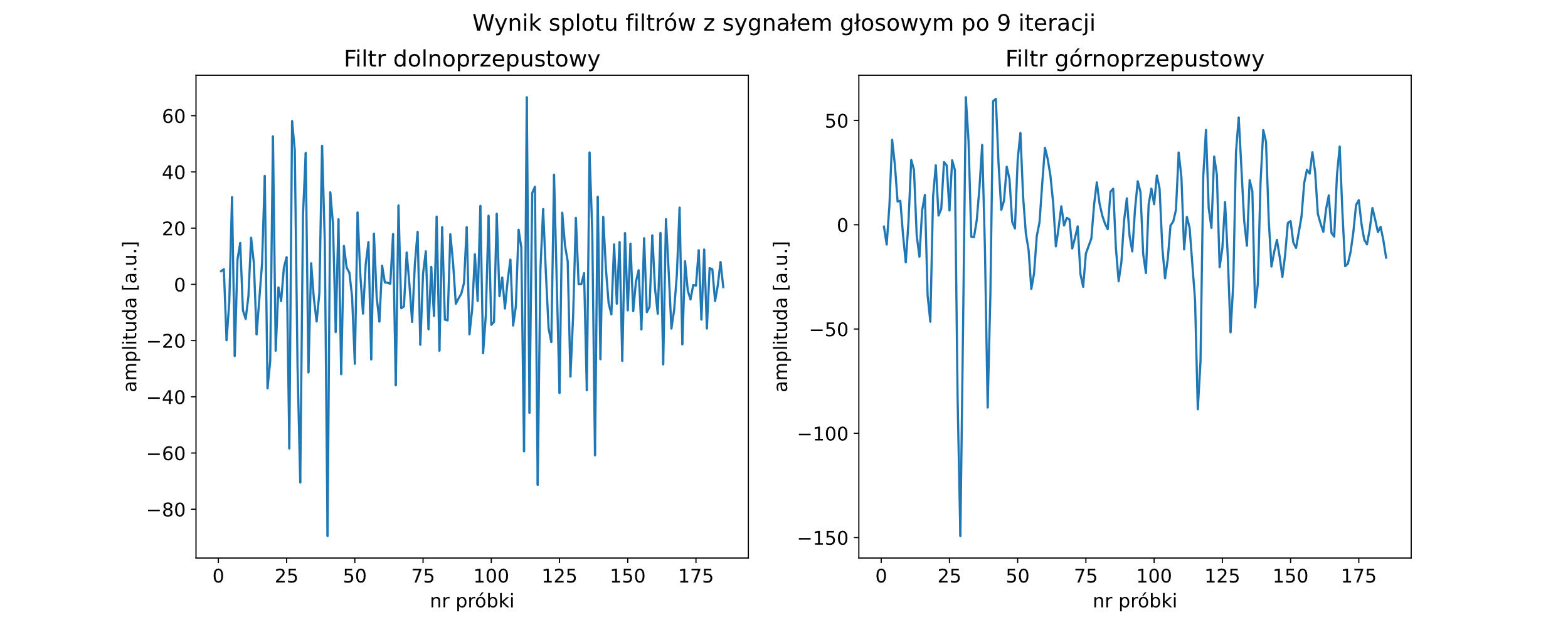
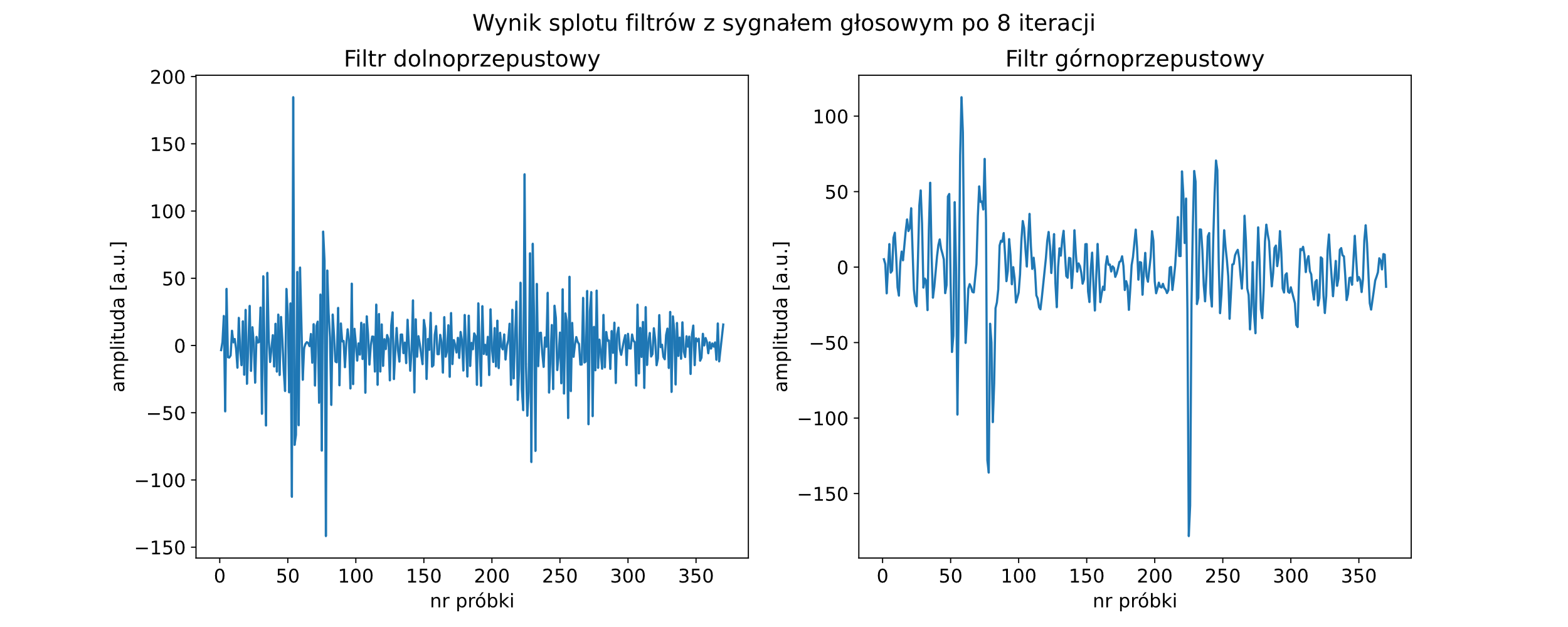
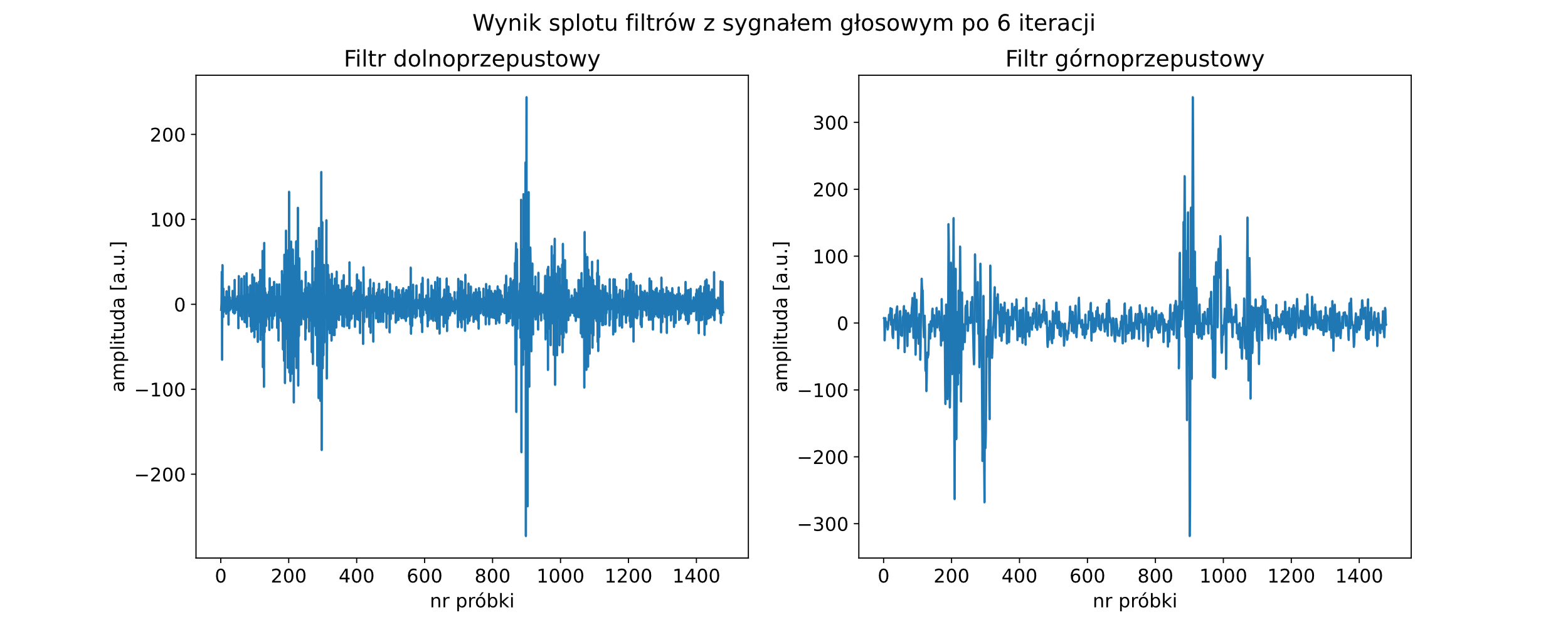
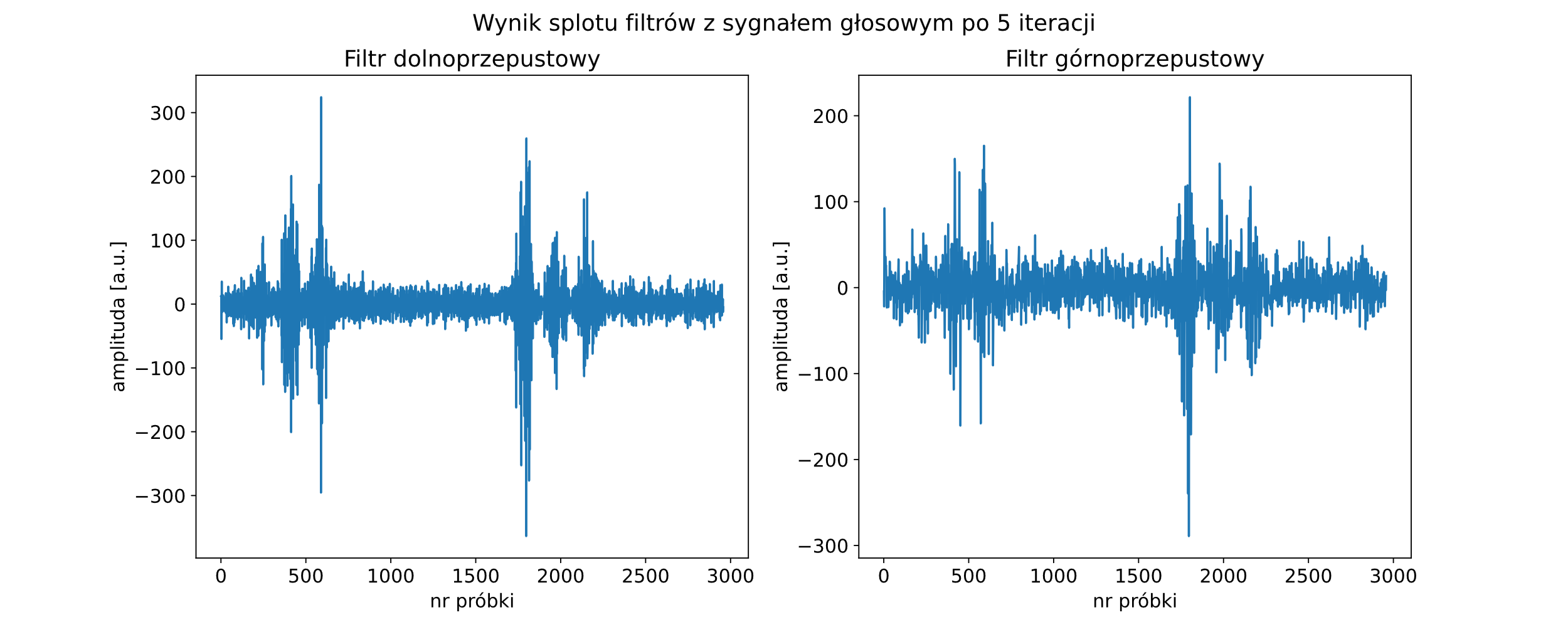
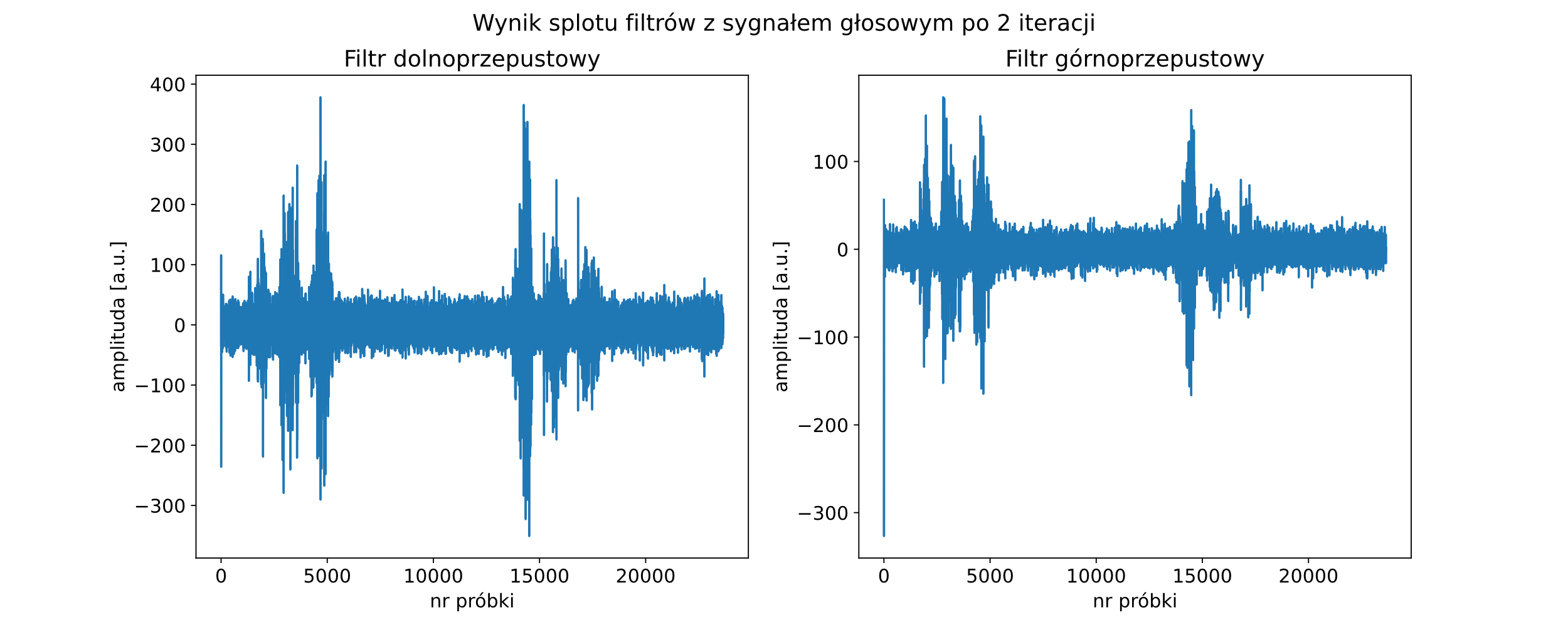
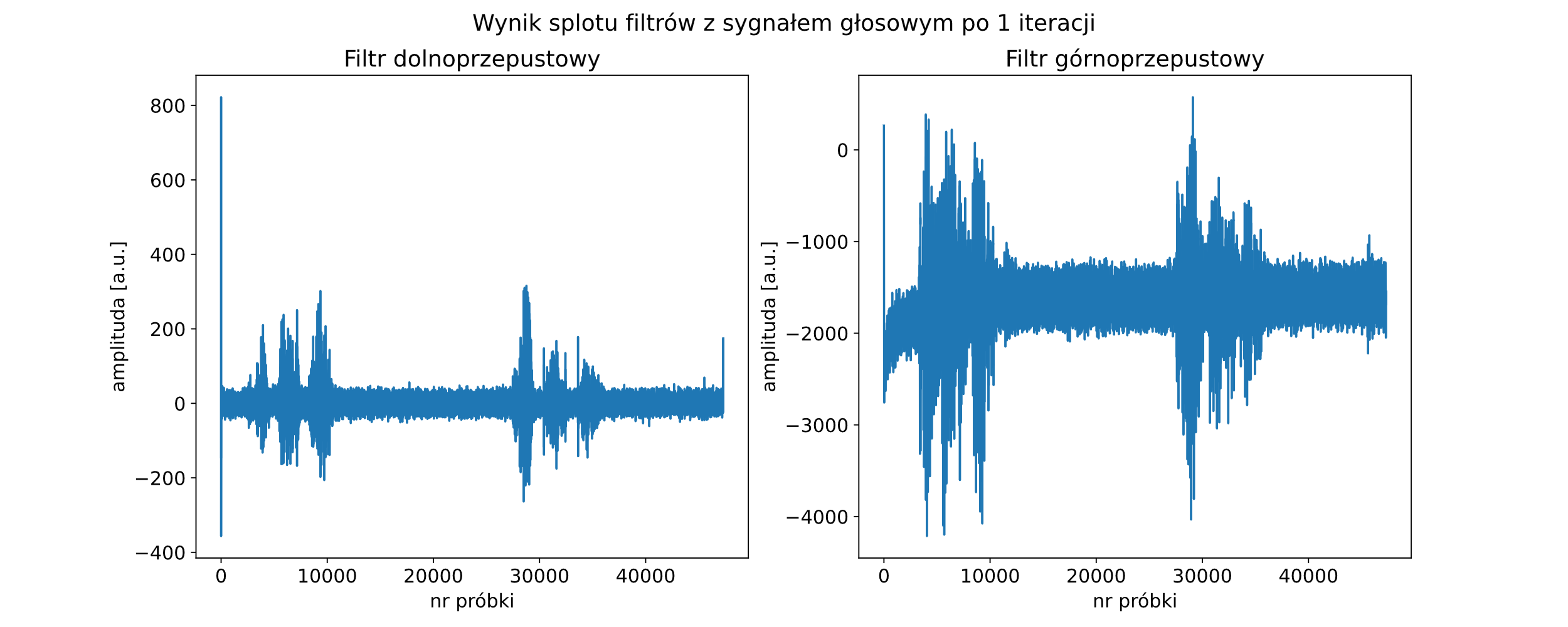
Wyniki splotów naszych filtrów z sygnałem świergotowym przedstawiają wykresy poniżej.

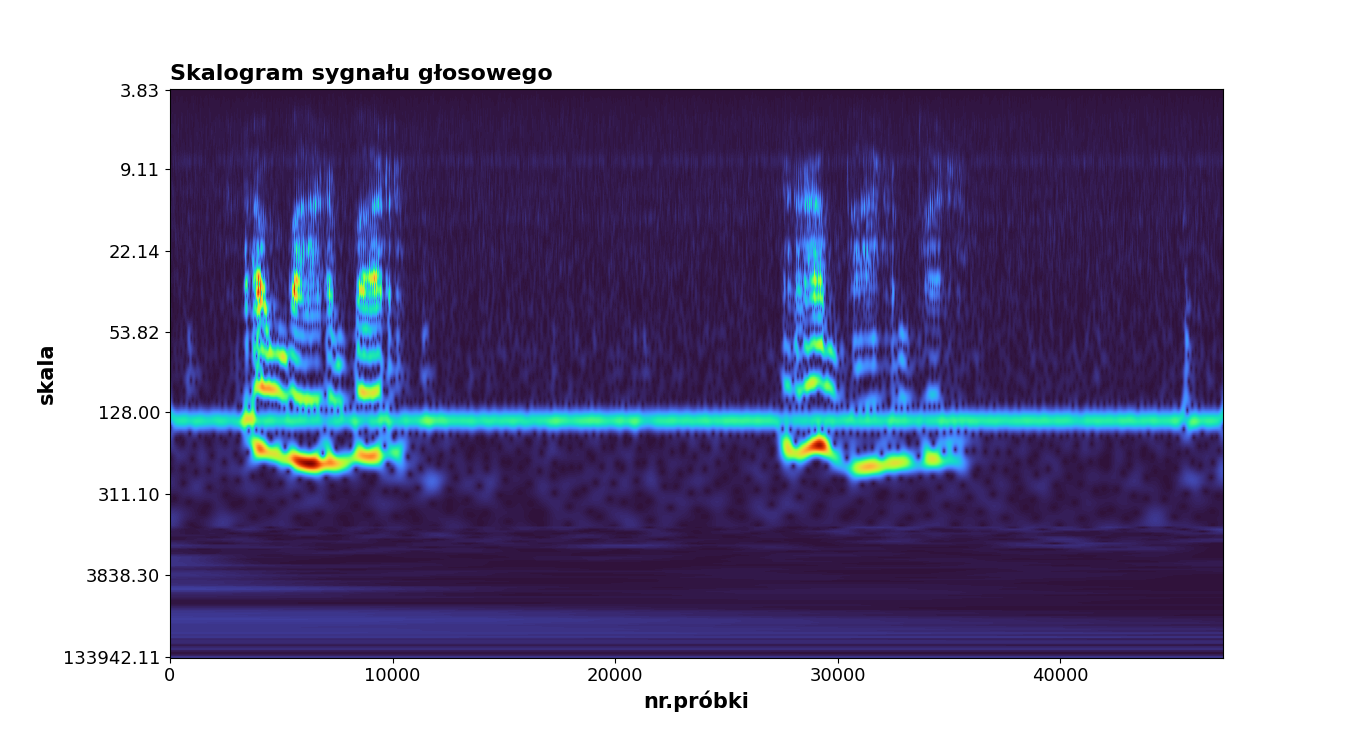


Skalogram sygnału świergotowego wykonany przy pomocy biblioteki *ssqueezepy* dla falki typu *morlet* przedstawia wykres poniżej (niestety przedstawiony w skali).

Kolejnym krokiem jest wczytanie sygnału głosowego z pliku *rabarbar8k.wav*. Sygnał ten jest przedstawiony na wykresie poniżej.

Powtarzając operacje wykonane na sygnale świergotowym, otrzymujemy następujące wyniki:



Skalogram dla sygnału głosowego (falka typu *morlet*) przedstawia wykres poniżej.

1. **Wnioski**

* Transformata falkowa umożliwia analizę sygnału w dziedzinie czas-skala.
* Transformata falkowa w odróżnieniu od transformaty Fouriera nie rozkłada sygnału w bazie fal sinusoidalnych, ale w bazie składowych będących odpowiednio przesuniętą i przeskalowaną falką podstawową.
* Falka jest falą o praktycznie ograniczonej długości i o zerowej wartości średniej.
* W transformacie falkowej sygnał przepuszczany jest na każdym poziomie przez parę filtrów: dolnoprzepustowy oraz górnoprzepustowy.
* Filtracja w dziedzinie częstotliwości jest równoważna operacji splotu w dziedzinie czasu.
* Odpowiedź układu w funkcji czasu można uzyskać przez operację splotu sygnału wejściowego z funkcją odpowiedzi na skok jednostkowy.
* Transformatę falkową można wykorzystać do kompresji plików (JPEG-2000).