Bogumił Wierzchowski 291498

Analiza sygnału w dziedzinie czasu i częstości

Ćwiczenie 3: Czas i częstość

1. **Wstęp**

Celem ćwiczenia jest porównanie transformaty Fouriera w krótkim oknie czasowym (STFT) i transformat biliniowych - Wigner-Ville oraz innych transformat z klasy Cohena.

Użyte oprogramowanie: *Python ver. 3.9.7*

Użyte biblioteki: *numpy, scipy, matplotlib*

1. **Kod źródłowy**

import numpy as np

import matplotlib.pyplot as plt

import scipy.signal as sig

from scipy.io import wavfile

from scipy.signal import hilbert

from scipy.signal import convolve2d

plt.rcParams["figure.figsize"] = [15, 8]

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

def int\_autocorr(x):

    N = x.size

    Rx = np.zeros((N, N), dtype='complex')

    for ti in range(N - 1):

        taumax = min([ti, N - ti - 1, int(round(N / 2.) - 1)])

        tau = np.arange(-taumax, taumax + 1)

        Rx[tau - tau[0], ti] = x[ti + tau] \* np.conj(x[ti - tau])

    return Rx

def cohen(x, fs, type\_='WV'):

    L = 30

    N = x.size

    t = np.arange(0, N).astype(float) / fs

    f = np.arange(0, N).astype(float) \* (fs / (2 \* N))

    CD = int\_autocorr(x)

    G = np.zeros((L, L), dtype=int)

    G[L // 2 - 1, L // 2 - 1] = 1

    CD = convolve2d(CD, G, "same")

    return np.fft.fft(CD, axis=0), f, t

N = 512

fs = 500

dt = 1/fs

t = np.arange(N)\*dt

f1 = 20

f2 = 100

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

plt.plot(t, chirp)

plt.xlabel('czas [s]')

plt.ylabel('amplituda')

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

plt.show()

plt.rcParams['font.size'] = '11'

fig, ((ax1, ax2), (ax3, ax4), (ax5, ax6)) = plt.subplots(3, 2)

fig.suptitle('Porównanie spektrogramu sygnału świergotowego 20-100 Hz')

nfft = 16

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax1.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax1.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

nfft = 32

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax2.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax2.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

nfft = 64

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax3.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax3.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

nfft = 128

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax4.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax4.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

nfft = 256

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax5.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax5.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

N = 512 \* 4

t = np.arange(N)\*dt

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

nfft = 512

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5))

ax6.pcolormesh(tt, ff, Sxx, shading='gouraud')

ax6.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title=f'nfft = {nfft}', ylim=(0,150))

plt.show()

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

N = 512

t = np.arange(N)\*dt

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

hilb\_chrip = sig.hilbert(chirp)

x\_wv, f\_wv, t\_wv = cohen(hilb\_chrip, fs, type\_='WV')

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

fig.suptitle('Transformata WV sygnału świergotowego 20-100 Hz')

pcm1 = ax1.pcolormesh(t\_wv, f\_wv, np.abs(x\_wv))

ax1.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='Moduł transformaty', ylim=(0,120))

fig.colorbar(pcm1, ax=ax1)

pcm2 = ax2.pcolormesh(t\_wv, f\_wv, np.log(np.abs(x\_wv)))

ax2.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='Logarytm modułu transformaty', ylim=(0,120))

fig.colorbar(pcm2, ax=ax2)

plt.show()

nfft = 128

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

fig.suptitle('Transformata STFT vs WV sygnału świergotowego 20-100 Hz')

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5), scaling='spectrum', mode='magnitude')

pcm1 = ax1.pcolormesh(tt, ff, 20 \* np.log10(Sxx), shading='gouraud')

ax1.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='STFT', ylim=(0,120))

fig.colorbar(pcm1, ax=ax1)

pcm2 = ax2.pcolormesh(t\_wv, f\_wv, 20 \* np.log10(np.abs(x\_wv)), shading='gouraud')

ax2.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='WV', ylim=(0,120))

fig.colorbar(pcm2, ax=ax2)

plt.show()

f2 = 200

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

hilb\_chrip = sig.hilbert(chirp)

x\_wv, f\_wv, t\_wv = cohen(hilb\_chrip, fs, type\_='WV')

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)

fig.suptitle('Transformata STFT vs WV sygnału świergotowego')

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5), scaling='spectrum', mode='magnitude')

pcm1 = ax1.pcolormesh(tt, ff, 20 \* np.log10(Sxx), shading='gouraud')

ax1.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='STFT 20-200 Hz', ylim=(0,220))

fig.colorbar(pcm1, ax=ax1)

pcm2 = ax2.pcolormesh(t\_wv, f\_wv, 20 \* np.log10(np.abs(x\_wv)), shading='gouraud')

ax2.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='WV 20-200 Hz', ylim=(0,220))

fig.colorbar(pcm2, ax=ax2)

f2 = 500

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

hilb\_chrip = sig.hilbert(chirp)

x\_wv, f\_wv, t\_wv = cohen(hilb\_chrip, fs, type\_='WV')

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5), scaling='spectrum', mode='magnitude')

pcm3 = ax3.pcolormesh(tt, ff, 20 \* np.log10(Sxx), shading='gouraud')

ax3.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='STFT 20-500 Hz')

fig.colorbar(pcm3, ax=ax3)

pcm4 = ax4.pcolormesh(t\_wv, f\_wv, 20 \* np.log10(np.abs(x\_wv)), shading='gouraud')

ax4.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='WV 20-500 Hz')

fig.colorbar(pcm4, ax=ax4)

plt.show()

f2 = 200

f2\_prim = f2 + f1

chirp = sig.chirp(t, f1, t[-1], f2) + sig.chirp(t, 2 \* f1, t[-1], f2\_prim)

hilb\_chrip = sig.hilbert(chirp)

x\_wv, f\_wv, t\_wv = cohen(hilb\_chrip, fs, type\_='WV')

fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2)

fig.suptitle('Transformata STFT vs WV sygnału świergotowego')

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5), scaling='spectrum', mode='magnitude')

pcm1 = ax1.pcolormesh(tt, ff, 20 \* np.log10(Sxx), shading='gouraud')

ax1.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='STFT 20-200 Hz + 40-220 Hz', ylim=(0,220))

fig.colorbar(pcm1, ax=ax1)

pcm2 = ax2.pcolormesh(t\_wv, f\_wv, 20 \* np.log10(np.abs(x\_wv)), shading='gouraud')

ax2.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='WV 20-200 Hz + 40-220 Hz', ylim=(0,220))

fig.colorbar(pcm2, ax=ax2)

f2\_prim = 2 \* f2

chirp = sig.chirp(t, f1, t[-1], f2) + sig.chirp(t, 2 \* f1, t[-1], f2\_prim)

hilb\_chrip = sig.hilbert(chirp)

x\_wv, f\_wv, t\_wv = cohen(hilb\_chrip, fs, type\_='WV')

ff, tt, Sxx = sig.spectrogram(chirp, fs=fs, noverlap=nfft/2, nfft=nfft, nperseg=int(nfft/1.5), scaling='spectrum', mode='magnitude')

pcm3 = ax3.pcolormesh(tt, ff, 20 \* np.log10(Sxx), shading='gouraud')

ax3.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='STFT 20-200 Hz + 40-400 Hz')

fig.colorbar(pcm3, ax=ax3)

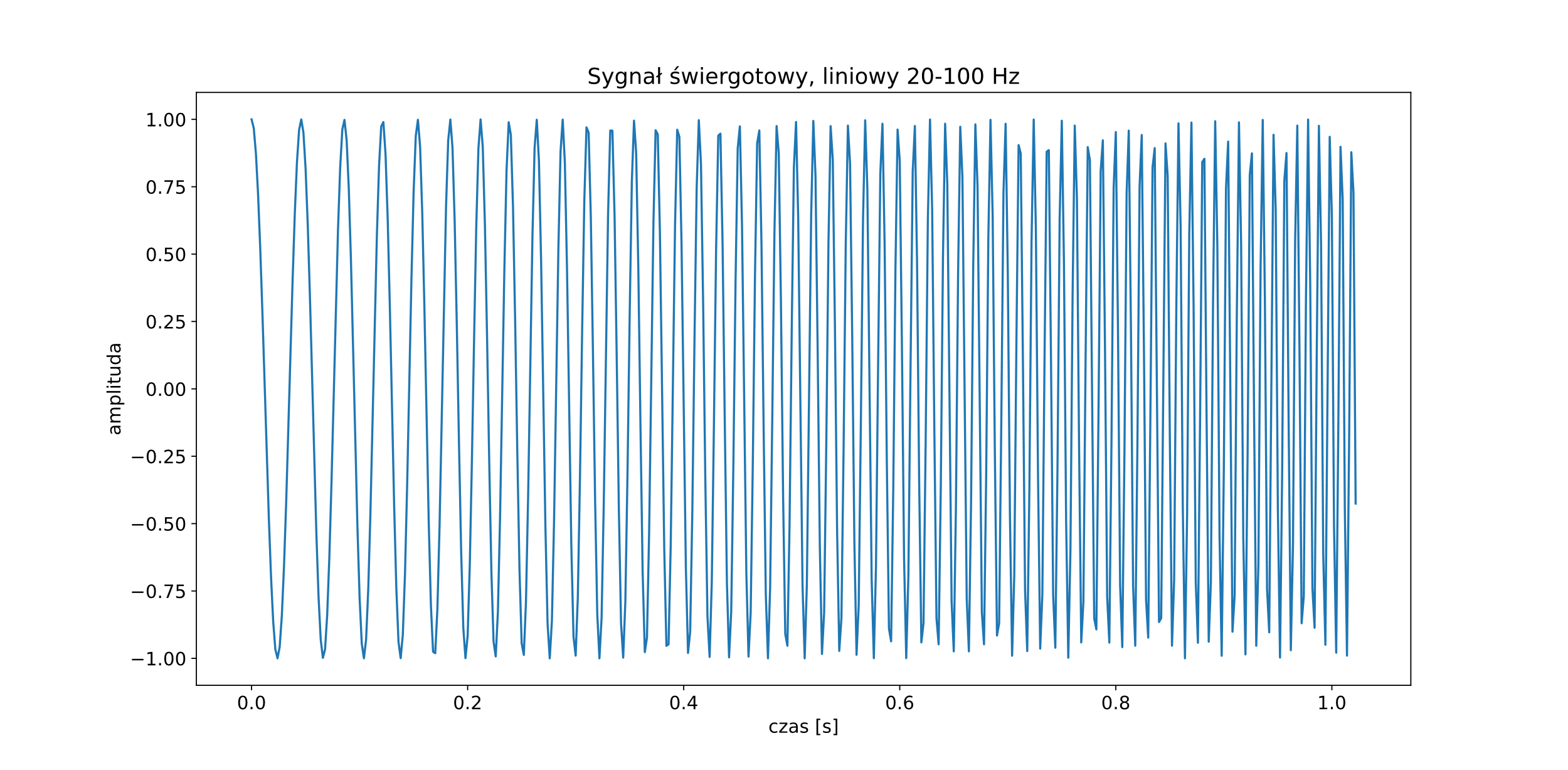
pcm4 = ax4.pcolormesh(t\_wv, f\_wv, 20 \* np.log10(np.abs(x\_wv)), shading='gouraud')

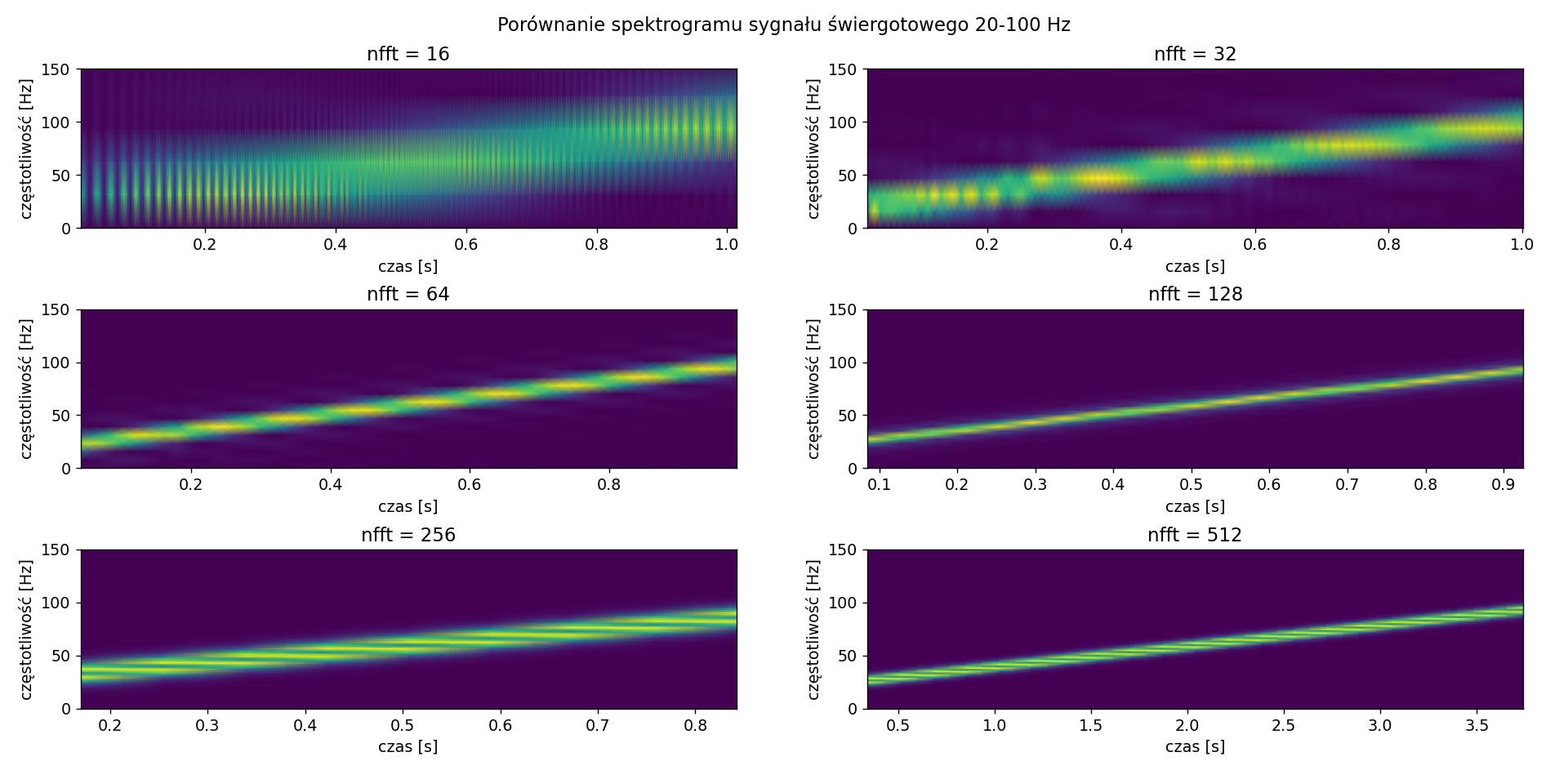
ax4.set(xlabel='czas [s]', ylabel='częstotliwość [Hz]', title='WV 20-200 Hz + 40-400 Hz')

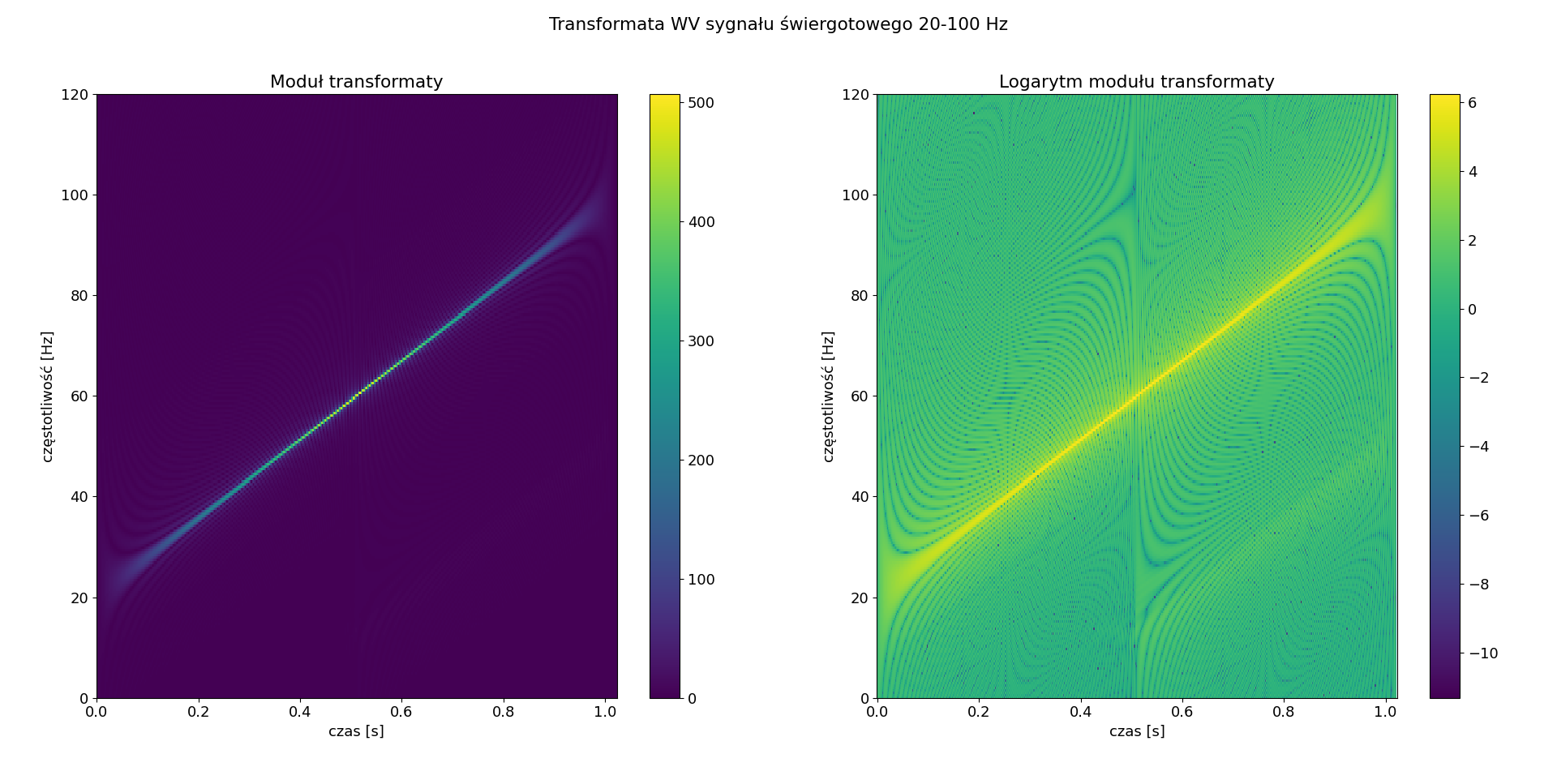
fig.colorbar(pcm4, ax=ax4)

plt.show()

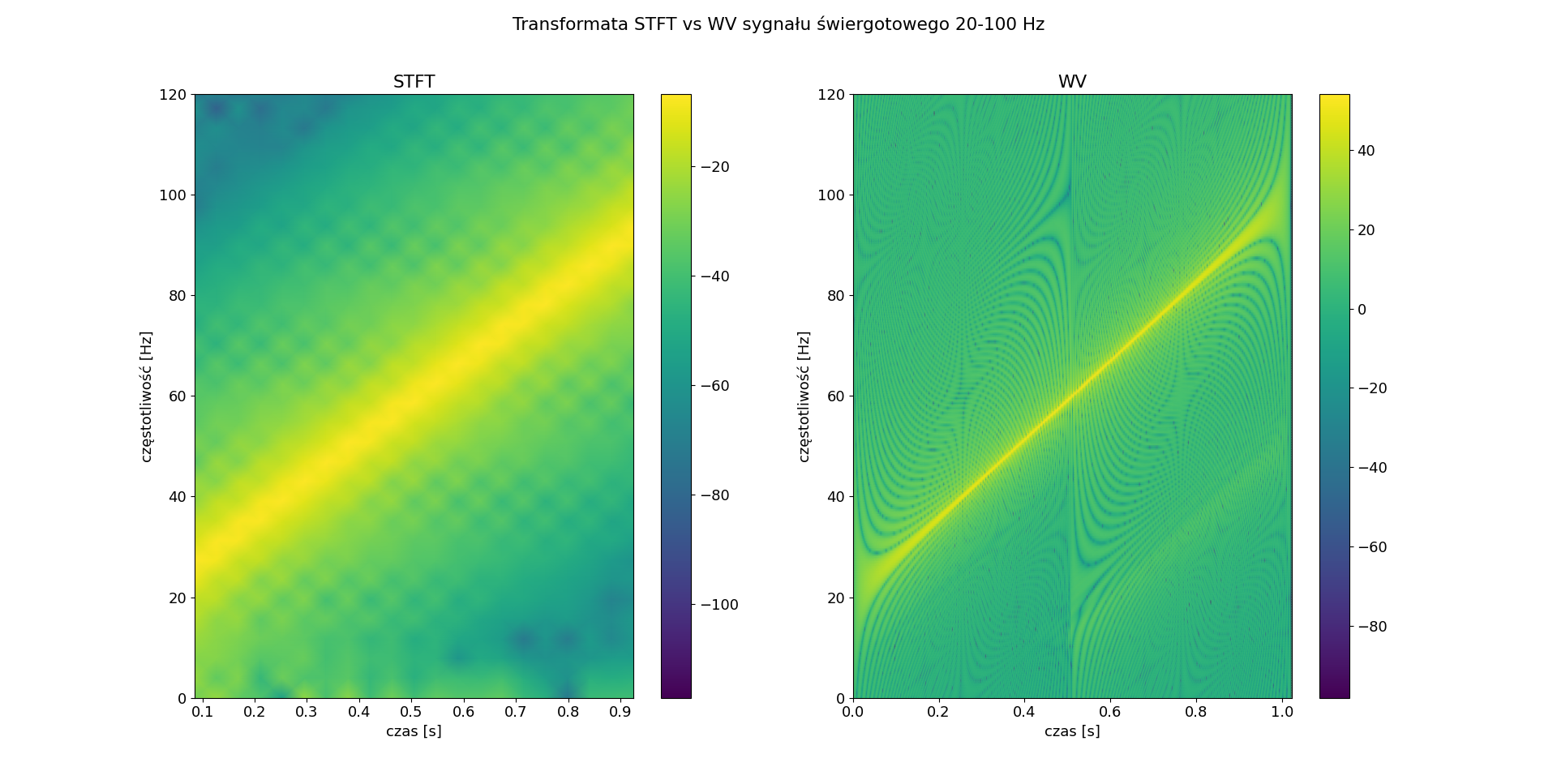
1. **Wyniki**

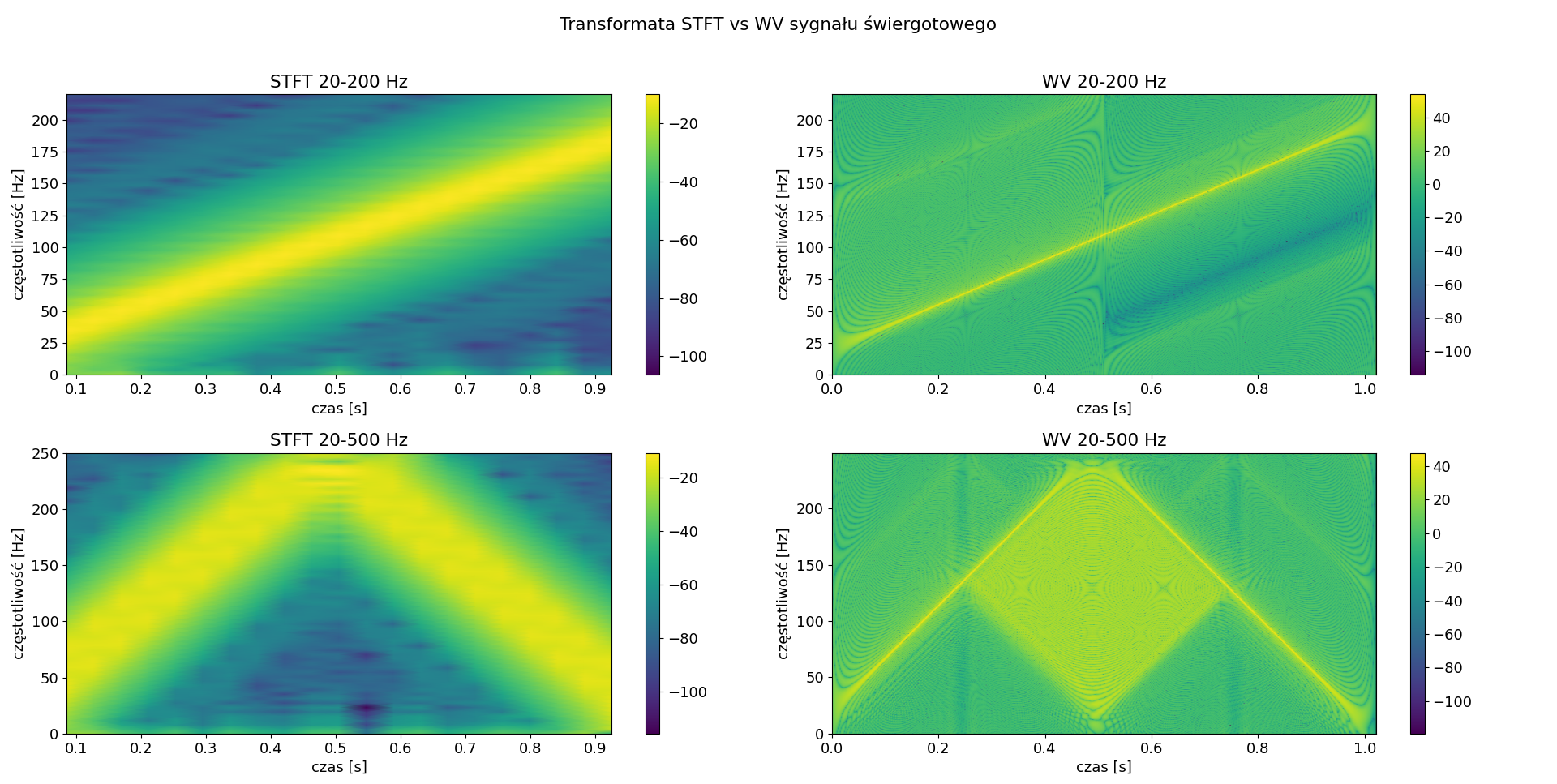
Wykres na poniższym rysunku przedstawia świergotowy sygnał liniowy o długości , częstości próbkowania , oraz częstościach początkowej i końcowej :

Poniżej przedstawiono porównanie spektrogramów sygnału świergotowego dla wartości *nfft* z zakresu 16-256, oraz dla wartości 512 gdzie długość sygnału zwiększono 4-krotnie:

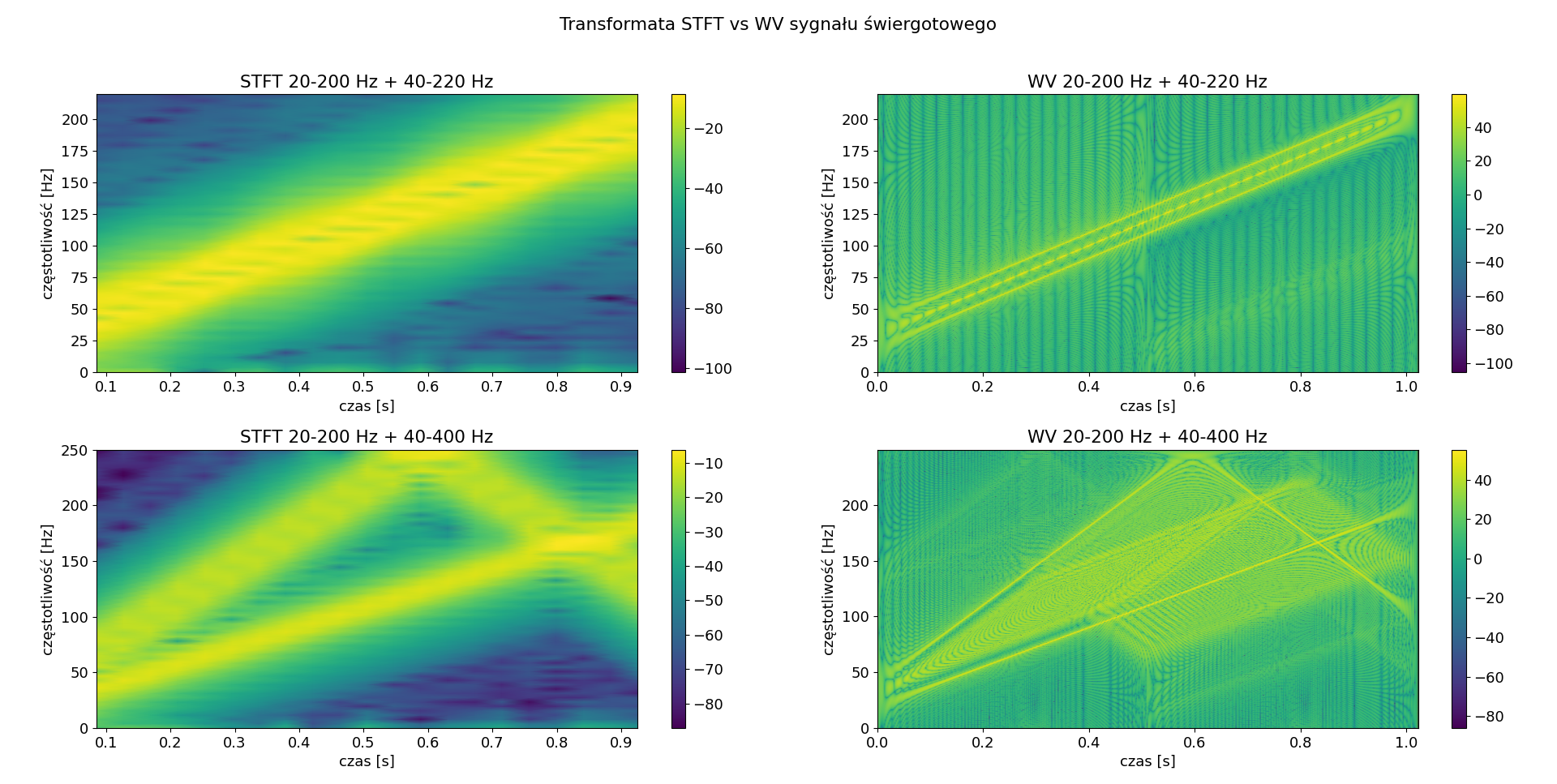
Transformatę Wigner-Ville z sygnału świergotowego przedstawia poniższy wykres:

Transformatę logarytmu VW i STFT (zlogarytmowana) dla sygnału świergotowego o długości

 oraz przedstawia wykres poniżej:

Transformatę logarytmu VW i STFT (zlogarytmowana) dla sygnału świergotowego o długości

oraz i dla częstotliwości końcowej przedstawia wykres poniżej:

Transformatę logarytmu VW i STFT (zlogarytmowana) dla sumy sygnałów świergotowych o długości

oraz i dla częstotliwości końcowej

przedstawia wykres poniżej:

1. **Wnioski**

* Sygnał świergotowy to taki, który zmienia swoją częstotliwość w czasie,
* Długość okna FFT, czyli parametr *nfft* ma znaczenie w przypadku wykonywania spektrogramu sygnału – nie może być ani zbyt mały (słaba rozdzielczość częstotliwościowa) ani zbyt duży (słaba rozdzielczość czasowa),
* Transformatę Wigner-Ville można użyć do wytworzenia spektrogramu sygnału świergotowego,
* W przypadku sygnału świergotowego o częstotliwości końcowej otrzymaliśmy na spektrogramie lustrzane odbicie mw. w połowie czasu trwania sygnału. Wynika to wprost z tw. o próbkowaniu – nasza częstotliwość próbkowania wynosiła , zatem maksymalnie mogliśmy zmierzyć częstotliwość .
* W przypadku sumy sygnałów świergotowych również widać zastosowanie twierdzenia o próbkowaniu, a ponadto transformata WV dała znacznie lepsze rezultaty, pozwalając lepiej odróżnić od siebie obie składowe świergotów.