

Ćwiczenie 2

Celem ćwiczenia było:

1. Wygeneruj ciąg próbek odpowiadający fali sinusoidalnej o częstotliwości 50 Hzi długości 65536.
2. Wyznacz dyskretną transformatę Fouriera tego sygnału i przedstaw jego widmo amplitudowe na wykresie w zakresie częstotliwości $[0, fs/2]$, gdzie fs oznacza częstotliwość próbkowania.
3. Wygeneruj ciąg próbek mieszaniny dwóch fal sinusoidalnych (tzn. ich kombinacji liniowej) o częstotliwościach 50 i 60 Hz. Wykonaj zadanie z punktu 2 dla tego sygnału.
4. Powtórz eksperymenty dla różnych czasów trwania sygnałów, tzn. dla różnych częstotliwości próbkowania.
5. Wyznacz odwrotne transformaty Fouriera ciągów wyznaczonych w zadaniu 2 i porównaj z ciągami oryginalnymi.

```
In [31]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [ ]: #Generacja widma sygnału sin_wave1 na podstawie fft

def displayWave(wave,fs,length,name):
    t = np.arange(length) / fs
    plt.figure(figsize=(20,5))
    plt.subplot(4,1,1)
    plt.plot(wave)
    plt.grid(True)
    plt.title(f"Sygnał {name} przed transformacją")

    fourier1 = np.fft.fft(wave)

    widmo = np.abs(fourier1)
    abs_widmo = widmo / np.max(widmo)
    freq = np.fft.fftfreq(len(t),1/fs)
    pos_freq = freq[:len(freq)//2]
    pos_widmo = abs_widmo[:len(abs_widmo)//2]

    plt.subplot(4,1,2)
    plt.plot(widmo[:length//2])
    plt.xlim(right = fs/2)
    plt.grid(True)
    plt.title(f"Sygnał {name} po transformacji")

    fourier2 = np.fft.ifft(fourier1)

    plt.subplot(4,1,3)
    plt.plot(fourier2)
    plt.grid(True)
    plt.title(f"Sygnał {name} po odwrotnej transformacji")

    plt.subplot(4,1,4)
    plt.plot(pos_freq,pos_widmo)
    plt.xlim(right = fs/2)
    plt.grid(True)
    plt.title("Widmo amplitudowe")
    plt.xlabel('Częstotliwość (Hz)')
    plt.ylabel('Amplituda')

    plt.subplots_adjust(hspace=0.75)
    plt.show()
```

Zadanie nr1

Jednocześnie generowana jest fala sinusoidalna do zadania 2 oraz mieszanina próbek do zadania 3.

```
In [34]: fs = 44100
freq1 = 50
freq2 = 60
length = 65536

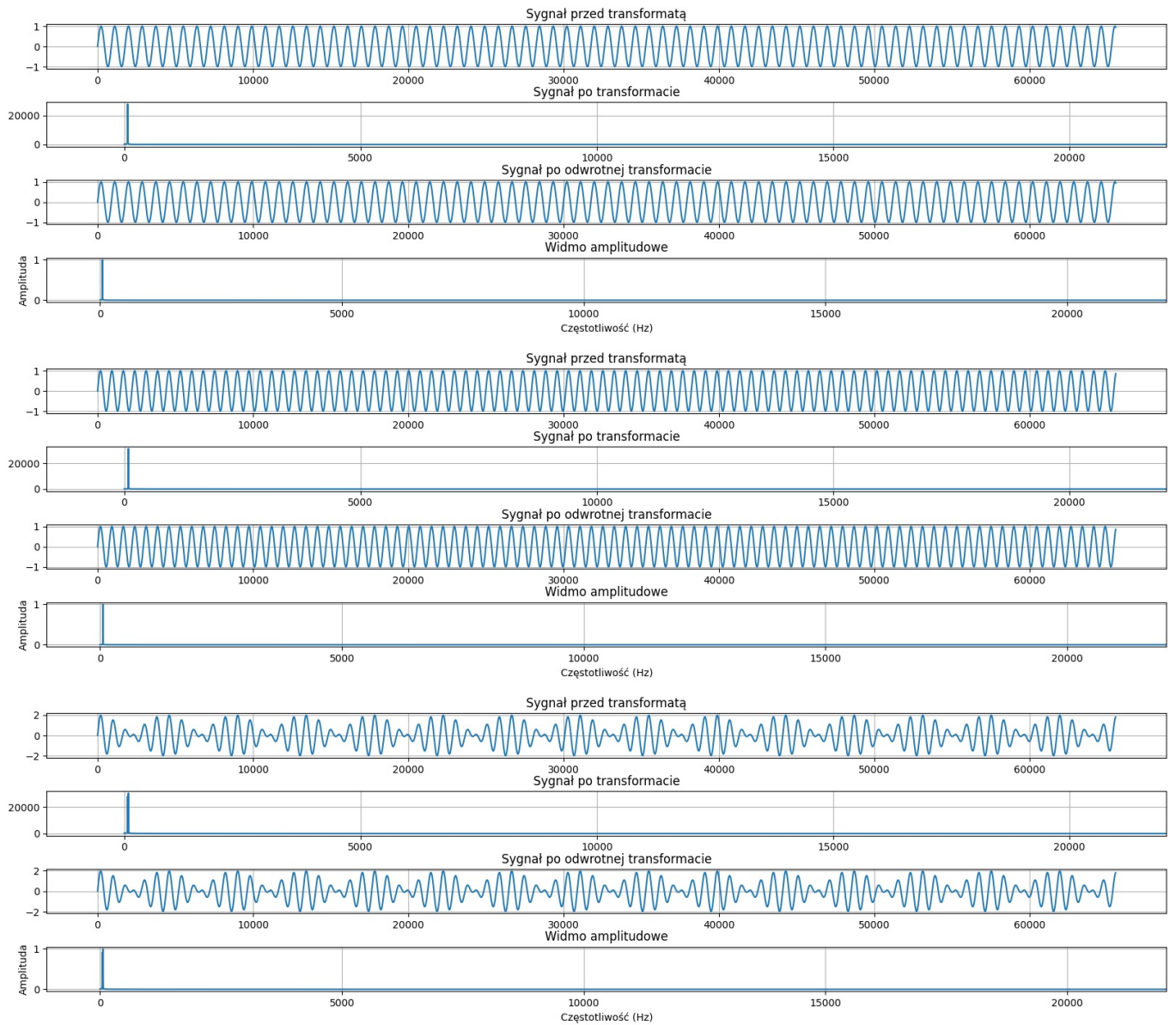
t = np.arange(length) / fs
sin_wave1 = np.sin(2 * np.pi * freq1 * t)

t = np.arange(length) / fs
sin_wave2 = np.sin(2 * np.pi * freq2 * t)

mixed_wave = sin_wave1 + sin_wave2

displayWave(sin_wave1,fs,length)
```

```
displayWave(sin_wave2,fs,length)
displayWave(mixed_wave,fs,length)
```



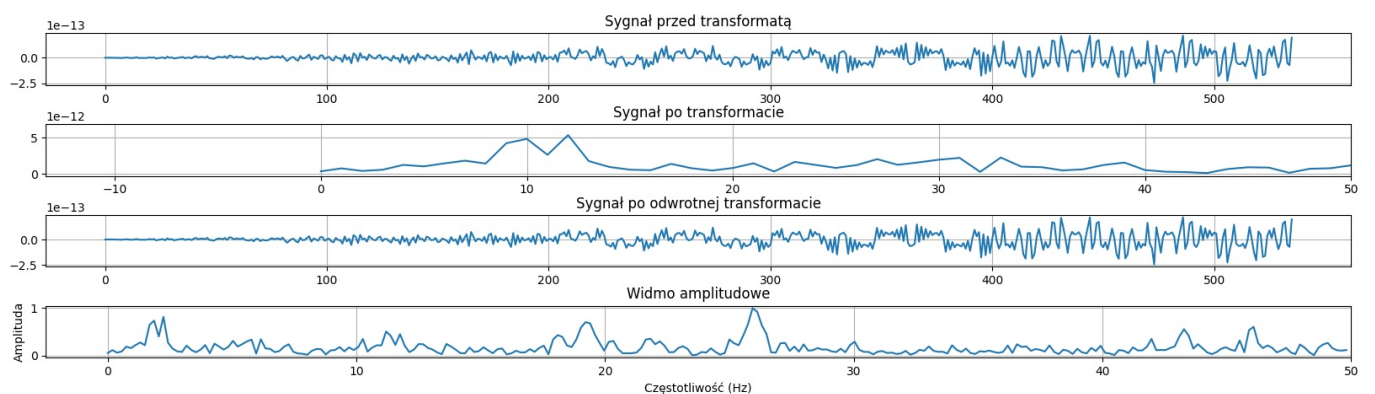
Do dalszych badań wybrano częstotliwości próbkowania 100Hz oraz 120Hz (Dwukrotności częstotliwości sygnału)

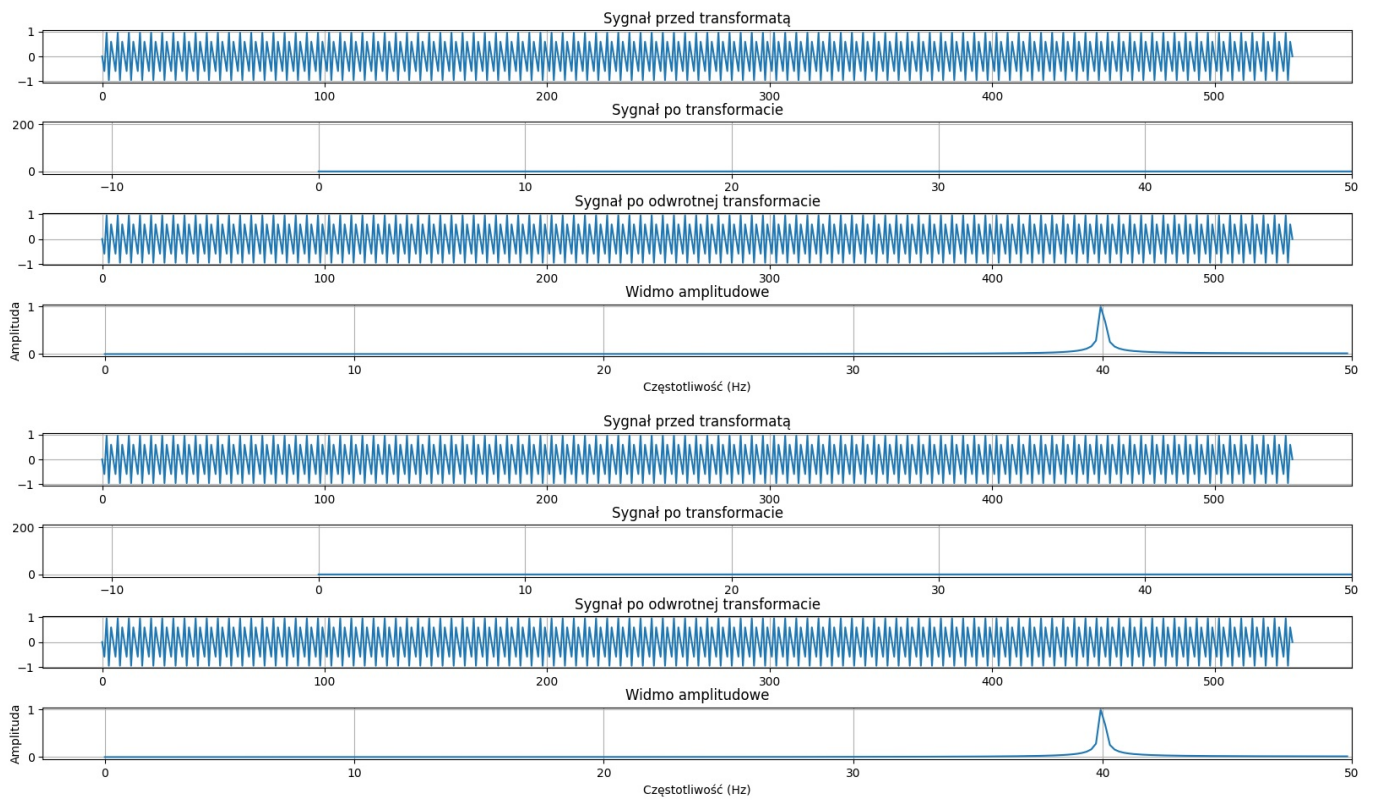
```
In [ ]: fs = 100
freq1 = 50
freq2 = 60
length = 536

t = np.arange(length) / fs
sin_wave1 = np.sin(2 * np.pi * freq1 * t)

t = np.arange(length) / fs
sin_wave2 = np.sin(2 * np.pi * freq2 * t)

displayWave(sin_wave1,fs,length)
displayWave(sin_wave2,fs,length)
```





```
In [ ]: fs = 120
freq1 = 50
freq2 = 60
length = 536

t = np.arange(length) / fs
sin_wave1 = np.sin(2 * np.pi * freq1 * t)

t = np.arange(length) / fs
sin_wave2 = np.sin(2 * np.pi * freq2 * t)

displayWave(sin_wave1, fs, length)
displayWave(sin_wave2, fs, length)
```

