Laboratorium 1: Analiza częstotliwościowa sygnałów czasu dyskretnego

Autorzy:


```
Mateusz Ostaszewski
Karol Kuc
```


```python
import numpy as np
from matplotlib import pyplot as plt
```

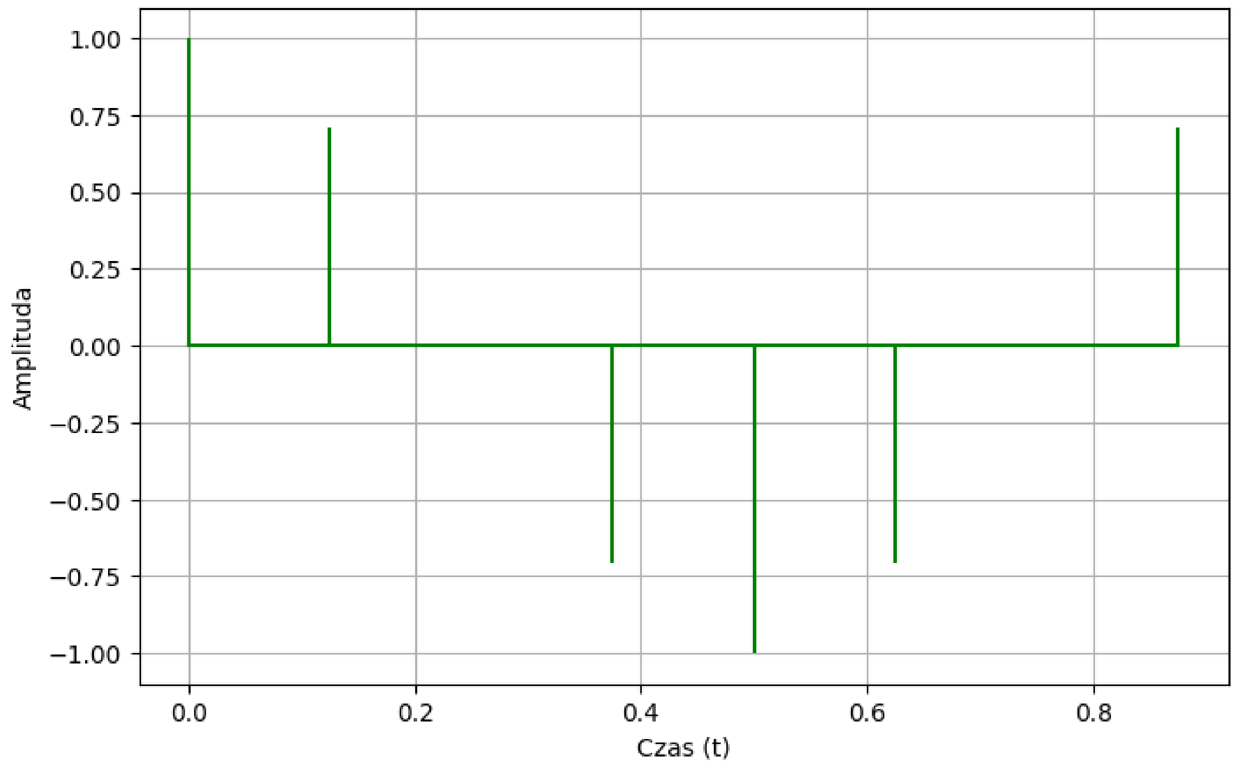
Zad1

Dany jest sygnał rzeczywisty $s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) + A_3 \sin(2\pi f_3 t)$, gdzie $A_1 = 0.2$, $f_1 = 2000$ Hz, $A_2 = 0.5$, $f_2 = 6000$ Hz, $A_3 = 0.6$, $f_3 = 10000$ Hz. Przy założeniu, że częstotliwość próbkowania wynosi $f_s = 48000$ Hz, a liczba próbek sygnału wynosi $N_1 = 2048$, przedstawić wykres widmowej gęstości mocy sygnału $s(t)$. Czy dla podanej liczby próbek mamy do czynienia ze zjawiskiem przecieku widma? Czy sytuacja uległaby zmianie dla liczby próbek $N_2 = \frac{3}{2} N_1$? Odpowiedź uzasadnić.

A

Sygnał spróbkowany

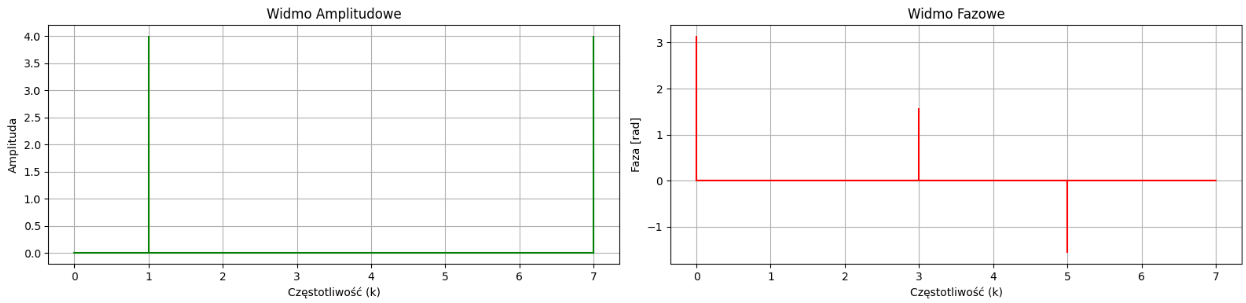
```python
# Parametry
N = 8
t = np.arange(N) / N  # Normalizacja czasu do jednego okresu
s = np.cos(2 * np.pi * t)  # Sygnał sinusoidalny

# Wykres sygnału spróbkowanego
plt.figure(figsize=(8, 5))
plt.stem(t, s, 'g', markerfmt=" ", basefmt="-g")
plt.xlabel('Czas (t)')
plt.ylabel('Amplituda')
plt.grid(True)
```



Widmo amplitudowe i fazowe

```
# FFT
fft_result = np.fft.fft(s)
amplitude_spectrum = np.abs(fft_result)
phase_spectrum = np.angle(fft_result)

# wykres dla widma amplitudowego i fazowego
fig, axs = plt.subplots(1, 2, figsize=(16, 4))

# amplitudowe
axs[0].stem(range(N), amplitude_spectrum, 'g', markerfmt=" ",
basefmt="-g")
axs[0].set_title('Widmo Amplitudowe')
axs[0].set_xlabel('Częstotliwość (k)')
axs[0].set_ylabel('Amplituda')
axs[0].grid(True)

# Widmo fazowe
axs[1].stem(range(N), phase_spectrum, 'r', markerfmt=" ", basefmt="-
r")
axs[1].set_title('Widmo Fazowe')
axs[1].set_xlabel('Częstotliwość (k)')
axs[1].set_ylabel('Faza [rad]')
axs[1].grid(True)
```

```
plt.tight_layout()
```



Weryfikacja twierdzenia

Dany jest sygnał rzeczywisty $s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) + A_3 \sin(2\pi f_3 t)$, gdzie $A_1 = 0.2$, $f_1 = 2000$ Hz, $A_2 = 0.5$, $f_2 = 6000$ Hz, $A_3 = 0.6$, $f_3 = 10000$ Hz. Przy założeniu, że częstotliwość próbkowania wynosi $f_s = 48000$ Hz, a liczba próbek sygnału wynosi $N_1 = 2048$, przedstawić wykres widmowej gęstości mocy sygnału $s(t)$. Czy dla podanej liczby próbek mamy do czynienia ze zjawiskiem przecieku widma? Czy sytuacja uległaby zmianie dla liczby próbek $N_2 = \frac{3}{2} N_1$? Odpowiedź uzasadnić.

Aby zweryfikować eksperymentalnie twierdzenie Parsevala, należy porównać całkowitą energię sygnału w dziedzinie czasu z całkowitą energią w dziedzinie częstotliwości.

```
energy_in_time_domain = np.sum(np.abs(s)**2)
energy_in_freqency_domain = np.sum(np.abs(fft_result)**2) / N

print(f"Całkowita energia sygnału w dziedzinie czasu wynosi: {energy_in_time_domain}")
print(f"Calkowita Energia sygnału w dziedzinie częstotliwości wynosi: {energy_in_freqency_domain}")
print(f"Różnica między energią w dziedzinie czasu i częstotliwości wynosi: {energy_in_time_domain - energy_in_freqency_domain}")

Całkowita energia sygnału w dziedzinie czasu wynosi: 4.0
Calkowita Energia sygnału w dziedzinie częstotliwości wynosi: 4.0
Różnica między energią w dziedzinie czasu i częstotliwości wynosi: 0.0
```

Wyniki potwierdzają twierdzenie Parsevala, mówiące, że całkowita energia sygnału w dziedzinie czasu jest równa całkowitej energii sygnału w dziedzinie częstotliwości.

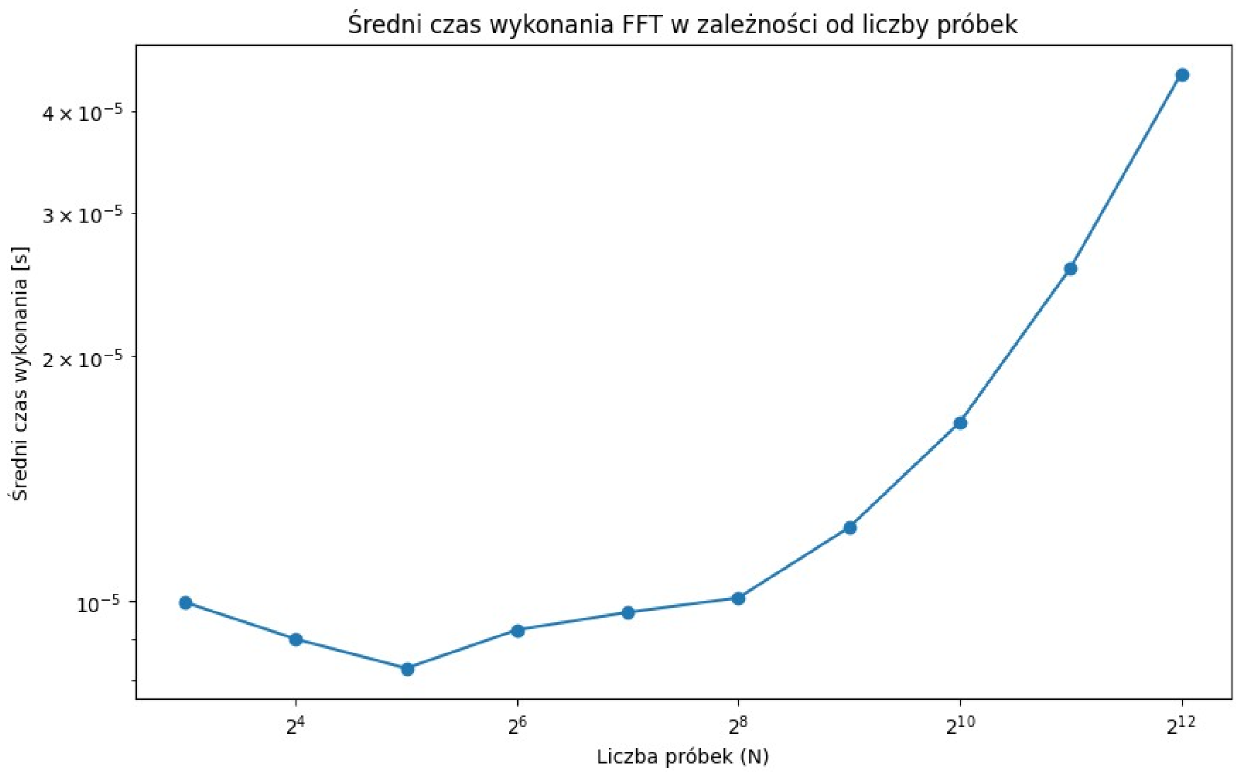
B

Wizualizacja złożoności obliczeniowej poprzez pomiary czasów

```
import timeit
from scipy.fft import fft
```

```python
l_values = np.arange(3, 13)
N_values = 2**l_values
num_tests = 200 # duża liczba testów aby wyniki były uśrednione

average_times = []

for N in N_values:
    times = []
    for _ in range(num_tests):
        t = np.arange(N) / N
        s = np.cos(2 * np.pi * t)
        time = timeit.timeit('fft(s)', globals=globals(), number=1000)
/ 1000
        times.append(time)

    # Uśrednianie czasów wykonania dla danej wartości N
    average_time = np.mean(times)
    average_times.append(average_time)


plt.figure(figsize=(10, 6))
plt.plot(N_values, average_times, 'o-')
plt.xscale('log', base=2)
plt.yscale('log')
plt.xlabel('Liczba próbek (N)')
plt.ylabel('Średni czas wykonania [s]')
plt.title('Średni czas wykonania FFT w zależności od liczby próbek')
```

```
Text(0.5, 1.0, 'Średni czas wykonania FFT w zależności od liczby
próbek')
```



Komentarz do Wykresu

Kształt wykresu wyraźnie przypomina wykres $ n\log(n) $ co jest zgodne z teoretyczną złożonością obliczeniową algorytmu FFT.

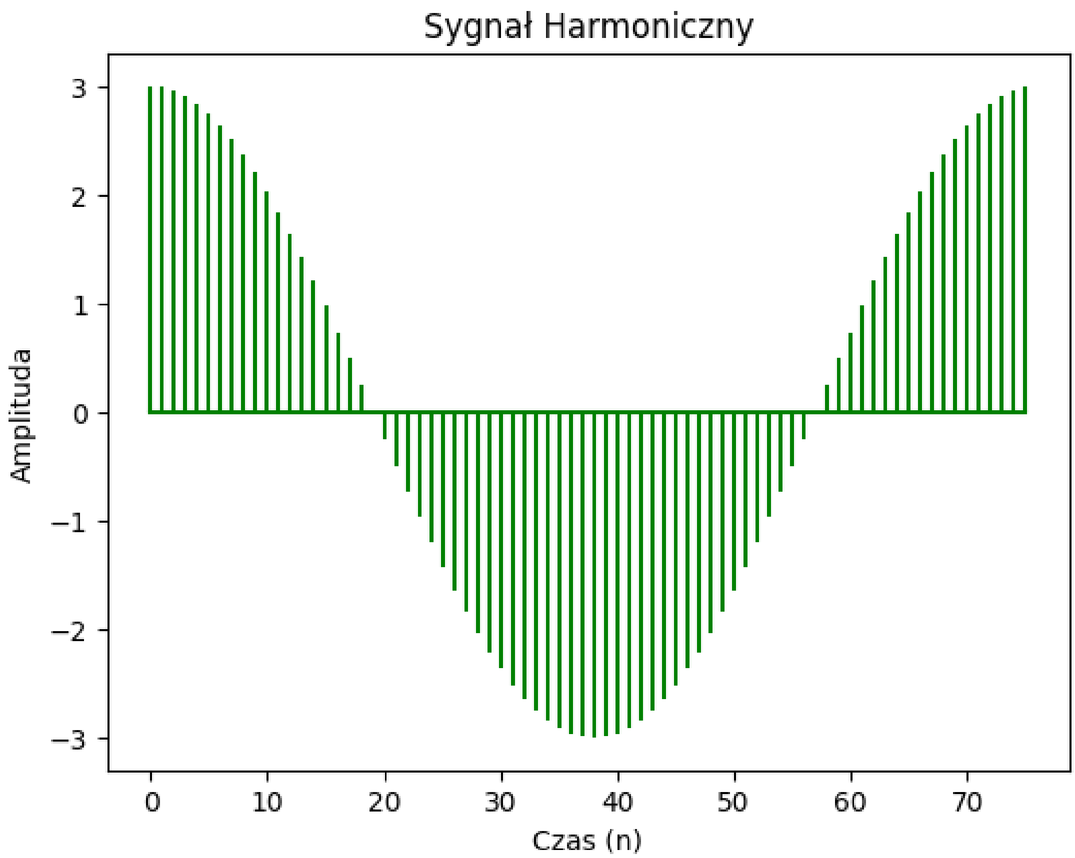
Zad2

Dany jest sygnał rzeczywisty $s(t) = A_1\sin(2\pi f_1 t) + A_2\sin(2\pi f_2 t) + A_3\sin(2\pi f_3 t)$, gdzie $A_1 = 0.2$, $f_1 = 2000$ Hz, $A_2 = 0.5$, $f_2 = 6000$ Hz, $A_3 = 0.6$, $f_3 = 10000$ Hz. Przy założeniu, że częstotliwość próbkowania wynosi $f_s = 48000$ Hz, a liczba próbek sygnału wynosi $N_1 = 2048$, przedstawić wykres widmowej gęstości mocy sygnału $s(t)$. Czy dla podanej liczby próbek mamy do czynienia ze zjawiskiem przecieku widma? Czy sytuacja uległaby zmianie dla liczby próbek $N_2 = \frac{3}{2}N_1$? Odpowiedź uzasadnić.

Generowanie zadanego sygnału harmonicznego

```
import matplotlib.pyplot as plt
import numpy as np

# Paramtery
A = 3
N = 76

n = np.arange(N)
```

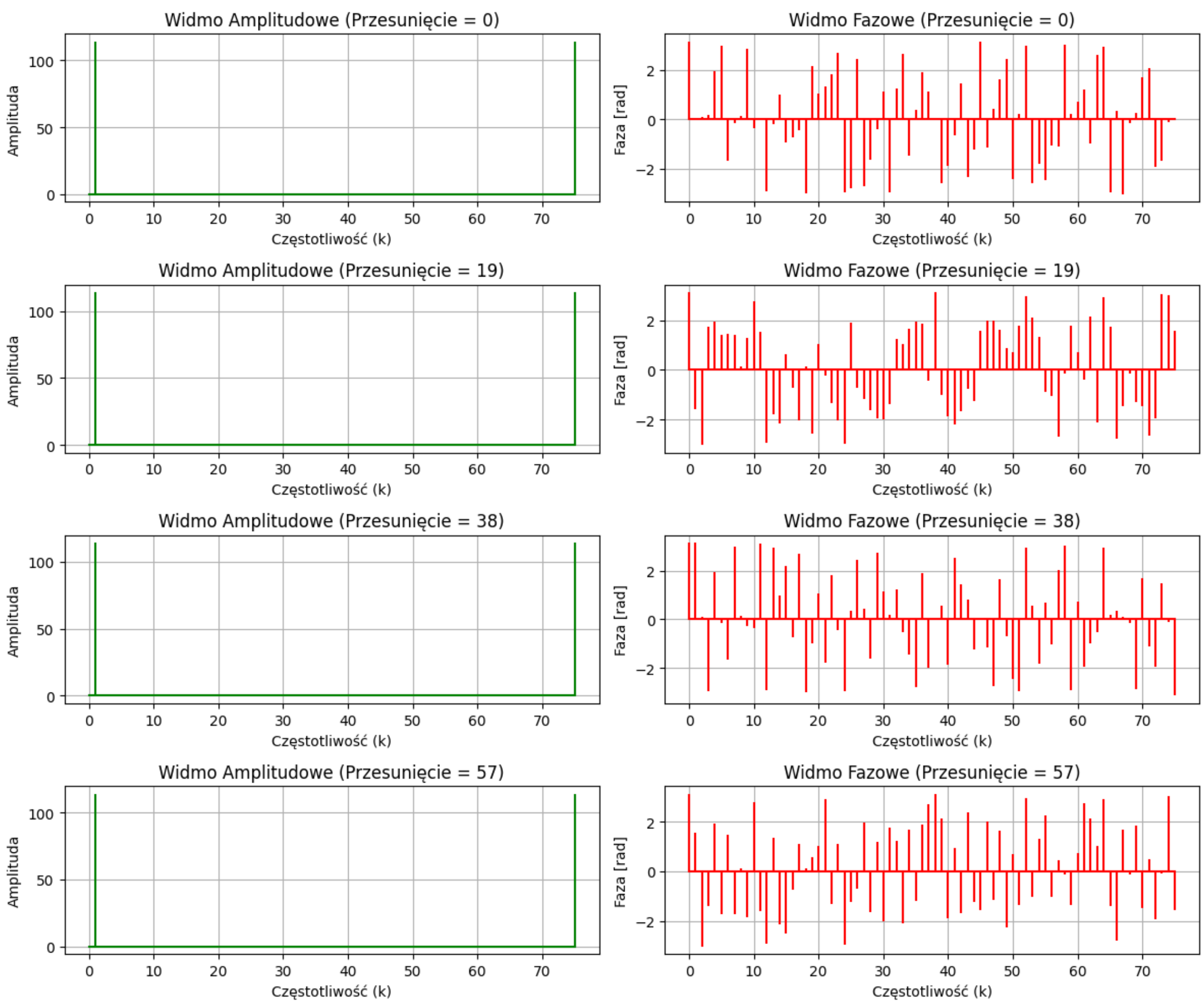
```
s = A * np.cos(2 * np.pi * n / N)

plt.stem(n, s, 'g', markerfmt=" ", basefmt="-g")
plt.title('Sygnał Harmoniczny')
plt.xlabel('Czas (n)')
plt.ylabel('Amplituda')

Text(0, 0.5, 'Amplituda')
```



Wykreślanie widma amplitudowego oraz fazowego dla wszystkich przesunięć

```
shifts = [0, N//4, N//2, 3*N//4]

fig, axs = plt.subplots(len(shifts), 2, figsize=(12, 10))

for i, shift in enumerate(shifts):
    shifted_signal = np.roll(s, shift)
    fft_result = np.fft.fft(shifted_signal)

    amplitude_spectrum = np.abs(fft_result)
    phase_spectrum = np.angle(fft_result)
```

```
    axs[i, 0].stem(n, amplitude_spectrum, 'g', markerfmt=" ",
basefmt="-g")
    axs[i, 0].set_title(f'Widmo Amplitudowe (Przesunięcie = {shift})')
    axs[i, 0].set_ylabel('Amplituda')
    axs[i, 0].grid(True)
    axs[i, 0].set(xlabel='Częstotliwość (k)')

    axs[i, 1].stem(n, phase_spectrum, 'r', markerfmt=" ", basefmt="-
r")
    axs[i, 1].set_title(f'Widmo Fazowe (Przesunięcie = {shift})')
    axs[i, 1].set_ylabel('Faza [rad]')
    axs[i, 1].grid(True)
    axs[i, 1].set(xlabel='Częstotliwość (k)')

plt.tight_layout()
```



Wnioski

Widmo Amplitudowe

Widmo amplitudowe sygnału przedstawia stałość wartości amplitud składowych częstotliwościowych sygnału, niezależnie od przesunięcia czasowego. To oznacza, że przesunięcie czasowe sygnału nie wpływa na obecność ani intensywność składowych częstotliwościowych sygnału w analizowanym zakresie. Ta niezmienność amplitudy w widmie jest fundamentalną właściwością, pozwalającą na identyfikację i analizę składowych częstotliwościowych sygnału bez względu na ich położenie w czasie.

Widmo Fazowe

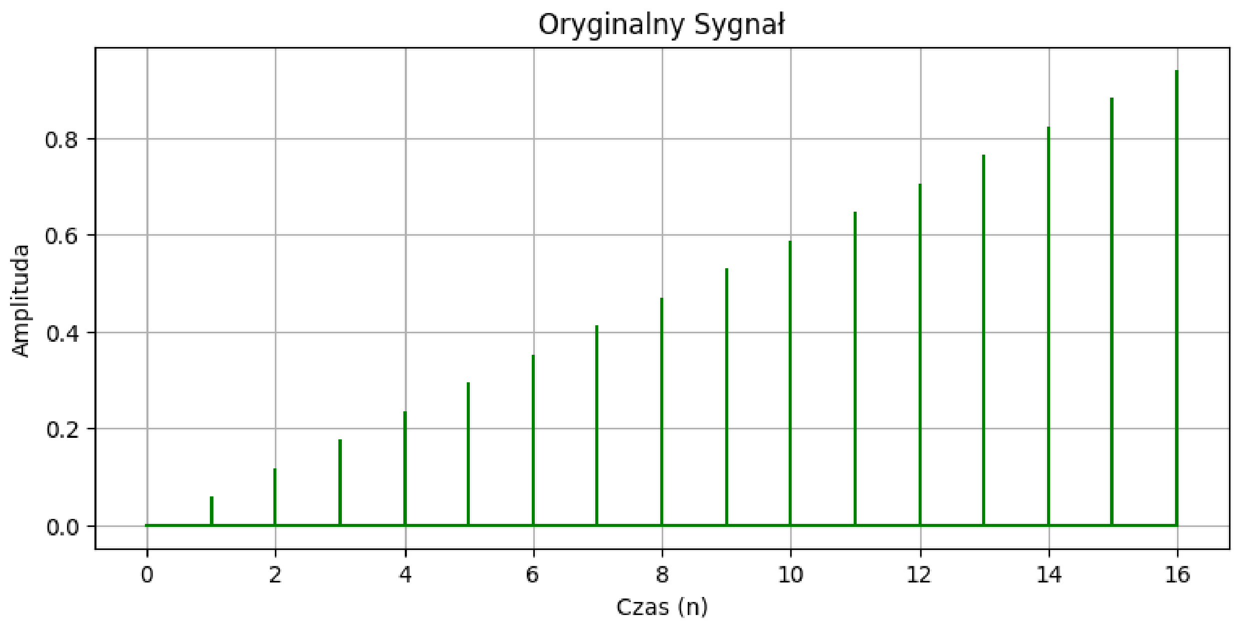
Widmo fazowe sygnału ulega zmianom w zależności od przesunięcia czasowego. Zmiana ta wskazuje, że przesunięcie czasowe wprowadza modyfikację fazy składowych częstotliwościowych sygnału. Informacja o fazie może dostarczać istotnych danych o charakterystyce sygnału. Przykładowo, w dziedzinach takich jak akustyka czy sejsmologia, analiza fazy może być wykorzystana do precyzyjnej lokalizacji źródeł dźwięku lub analizy ruchów sejsmicznych. Zmiana fazowa w odpowiedzi na przesunięcie czasowe podkreśla dynamikę sygnału

Zad3

Dany jest sygnał rzeczywisty $s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) + A_3 \sin(2\pi f_3 t)$, gdzie $A_1 = 0.2$, $f_1 = 2000$ Hz, $A_2 = 0.5$, $f_2 = 6000$ Hz, $A_3 = 0.6$, $f_3 = 10000$ Hz. Przy założeniu, że częstotliwość próbkowania wynosi $f_s = 48000$ Hz, a liczba próbek sygnału wynosi $N_1 = 2048$, przedstawić wykres widmowej gęstości mocy sygnału $s(t)$. Czy dla podanej liczby próbek mamy do czynienia ze zjawiskiem przecieku widma? Czy sytuacja uległaby zmianie dla liczby próbek $N_2 = \frac{3}{2} N_1$? Odpowiedź uzasadnić.

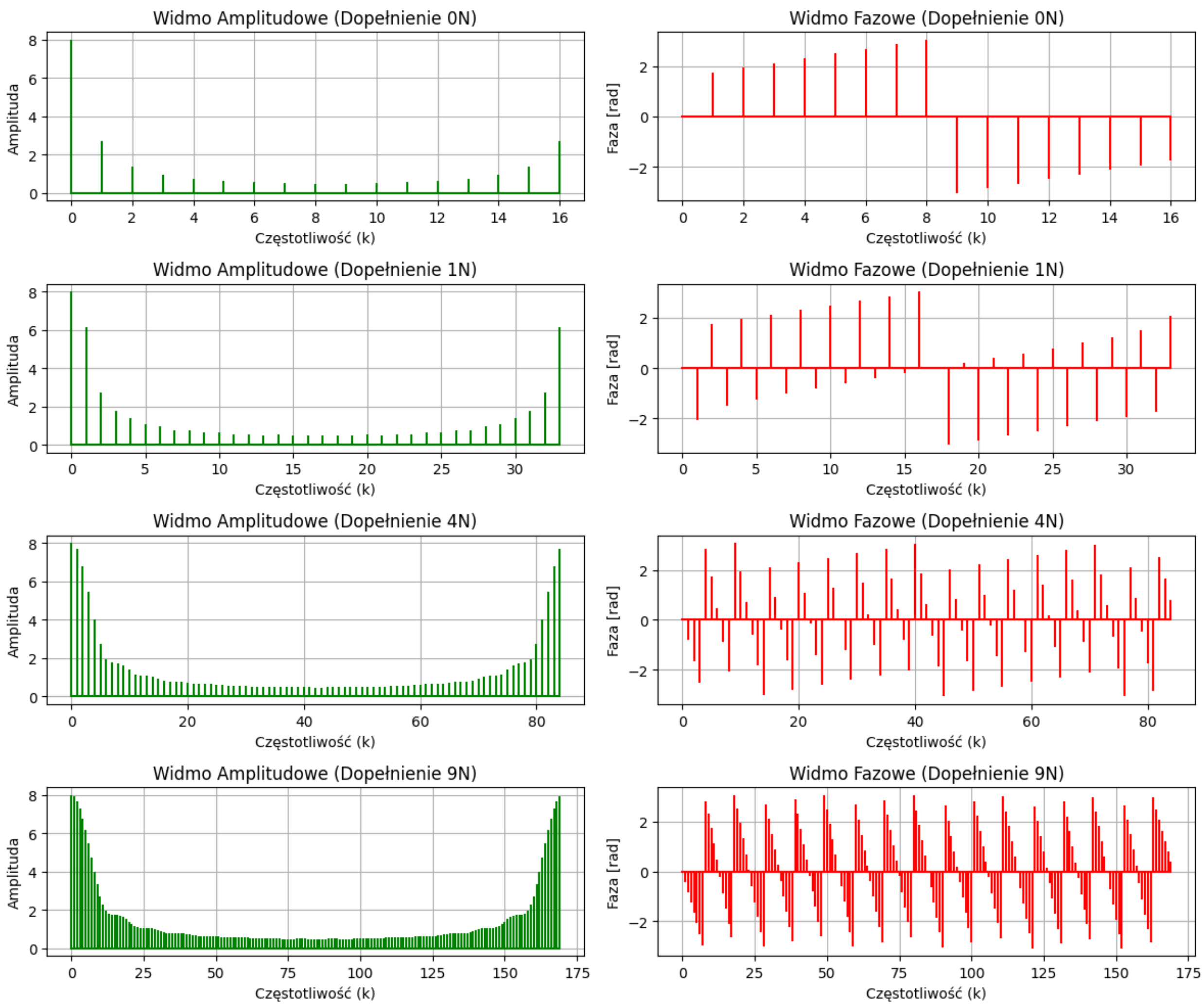
Generowanie sygnału bazowego

```
N = 17
A = 1
n_base = np.arange(N)
s_base = A * np.mod(n_base, N) / N

# Wykres bazowego sygnału
plt.figure(figsize=(9, 4))
plt.stem(n_base, s_base, 'g', markerfmt=" ", basefmt="-g")
plt.title('Oryginalny Sygnał')
plt.xlabel('Czas (n)')
plt.ylabel('Amplituda')
plt.grid(True)
plt.show()
```



Dopełnienie zerami

```
zero_paddings = [0, 1, 4, 9]

fig, axs = plt.subplots(len(zero_paddings), 2, figsize=(12, 10))

for i, padding in enumerate(zero_paddings):
    padded_signal = np.pad(s_base, (0, padding * N), 'constant')
    fft_result = np.fft.fft(padded_signal)

    amplitude_spectrum = np.abs(fft_result)
    phase_spectrum = np.angle(fft_result)

    axs[i, 0].stem(range(len(amplitude_spectrum)), amplitude_spectrum,
'g', markerfmt=" ", basefmt="-g")
    axs[i, 0].set_title(f'Widmo Amplitudowe (Dopełnienie {padding}N)')
    axs[i, 0].set_ylabel('Amplituda')
    axs[i, 0].grid(True)
    axs[i, 0].set(xlabel='Częstotliwość (k)')

    axs[i, 1].stem(range(len(phase_spectrum)), phase_spectrum, 'r',
markerfmt=" ", basefmt="-r")
    axs[i, 1].set_title(f'Widmo Fazowe (Dopełnienie {padding}N)')
    axs[i, 1].set_ylabel('Faza [rad]')
    axs[i, 1].grid(True)
    axs[i, 1].set(xlabel='Częstotliwość (k)')

plt.tight_layout()
```



Wnioski

Na podstawie powyższych wykresow można stwierdzić, iż dopełnianie zerami zarówno widma amplitudowego jak i fazowego w obu przypadkach powoduje zwiększenie ilości punktów dzięki czemu wraz ze wzrostem dopełnianej liczby zer, wykresy sygnału dyskretnego zaczynają coraz bardziej przypominać wykresy ciągłe - punkty zagęszczają się. Daje to możliwość dokładniejszej obserwacji i analizy wyników.

Zad4

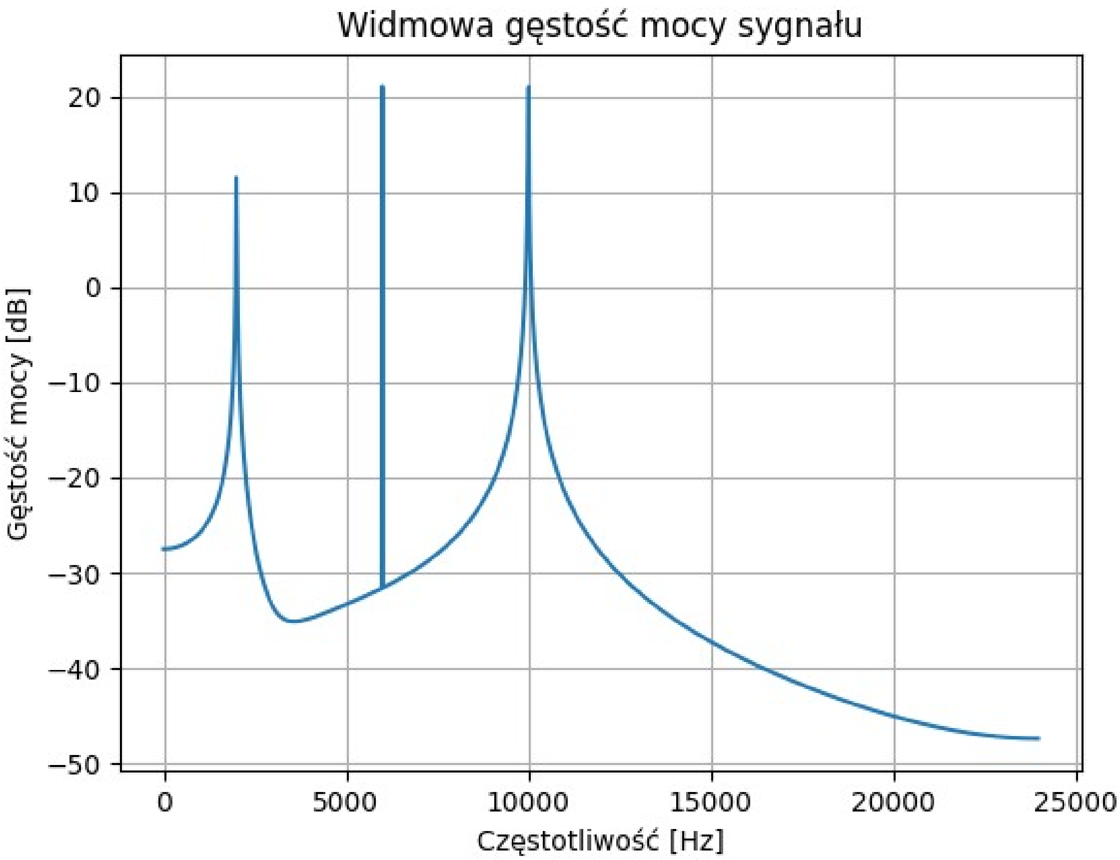
Dany jest sygnał rzeczywisty $s(t) = A_1 \sin(2\pi f_1 t) + A_2 \sin(2\pi f_2 t) + A_3 \sin(2\pi f_3 t)$, gdzie $A_1 = 0.2$, $f_1 = 2000$ Hz, $A_2 = 0.5$, $f_2 = 6000$ Hz, $A_3 = 0.6$, $f_3 = 10000$ Hz. Przy założeniu, że częstotliwość próbkowania wynosi $f_s = 48000$ Hz, a liczba próbek sygnału wynosi $N_1 = 2048$, przedstawić wykres widmowej gęstości mocy sygnału $s(t)$. Czy dla podanej liczby próbek mamy do czynienia ze zjawiskiem przecieku widma? Czy sytuacja uległaby zmianie dla liczby próbek $N_2 = \frac{3}{2} N_1$? Odpowiedź uzasadnić.

Generowanie sygnału

```python
# Parametry
A1, f1 = 0.2, 2000
A2, f2 = 0.5, 6000
A3, f3 = 0.6, 10000
fs = 48000
N1 = 2048
N2 = 3072

t = np.arange(N1) / fs

# sygnał
s = A1 * np.sin(2 * np.pi * f1 * t) + A2 * np.sin(2 * np.pi * f2 * t)
+ A3 * np.sin(2 * np.pi * f3 * t)
```

Wykres widmowej gestości mocy sygnalu

```python
psd = np.abs(np.fft.fft(s))**2 / N1
freqs = np.fft.fftfreq(N1, 1/fs)

plt.plot(freqs[:N1 // 2], 10 * np.log10(psd[:N1 // 2]))  # Skala dB
plt.title('Widmowa gęstość mocy sygnału')
plt.xlabel('Częstotliwość [Hz]')
plt.ylabel('Gęstość mocy [dB]')
plt.grid(True)
```



Wnioski i odpowiedzi

Na przedstawionym powyżej wykresie widać, że główne piki są otoczone przez mniejsze piki boczne co może sugerować występowanie zjawiska wycieku widma. Jednakże co najważniejsze zgodnie z wzorem na $\text{Rozdzielczość częstotliwościową} = \frac{f_s}{N}$ każda składowa częstotliwościowa w sygnale powinna pasować do wielokrotności podstawowej rozdzielczości częstotliwościowej, aby uniknąć przecieku. I tak dla N1 wartość ta będzie wynosiła 23.44 Hz a dla N2 15.63 Hz.

N1 = 2048

- f1 = 2000 / 23.44 ≈ 85.37
- f2 = 6000 / 23.44 ≈ 256.10
- f3 = 10000 / 23.44 ≈ 426.84

N2 = 3072

- f1 = 2000 / 15.63 ≈ 127.95
- f2 = 6000 / 15.63 ≈ 383.86
- f3 = 10000 / 15.63 ≈ 639.77

Po podzieleniu każdej składowej częstotliwości sygnału przez dane N w żadnym z dwóch przypadków nie otrzymujemy liczby całkowitej co wskazuje na to, iż w obu sytuacjach dochodzi do przecieku widma.