

**ŁUKASZ STANISZEWSKI, NR INDEKSU: 304098**

# WPROWADZENIE DO MULTIMEDIÓW

SPRAWOZDANIE Z LABORATORIUM 1 – SYGNAŁY I WIDMA



# I. Polecenie 1.

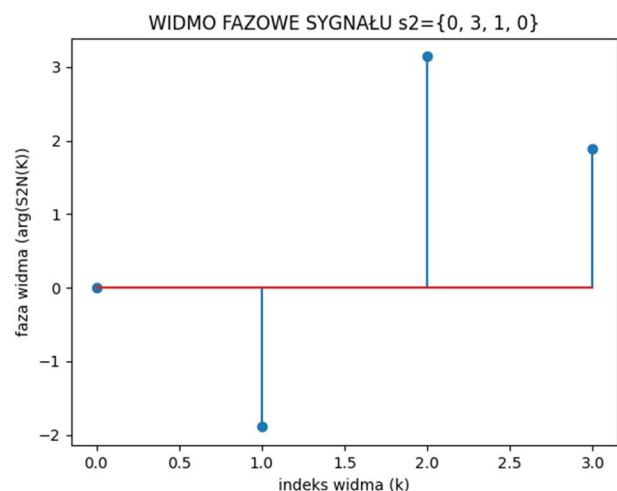
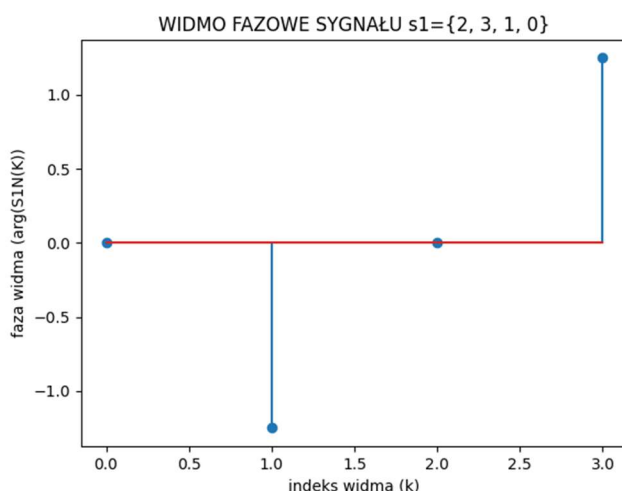
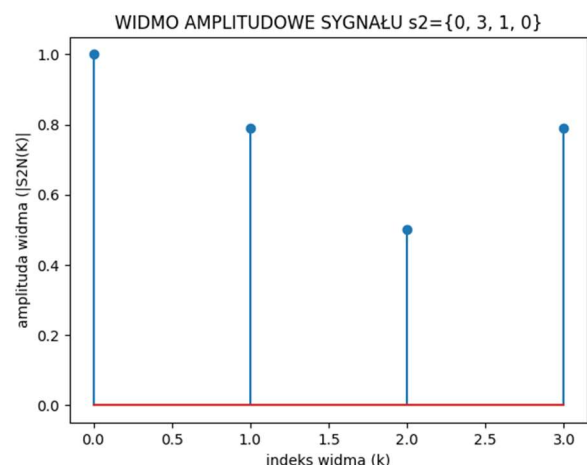
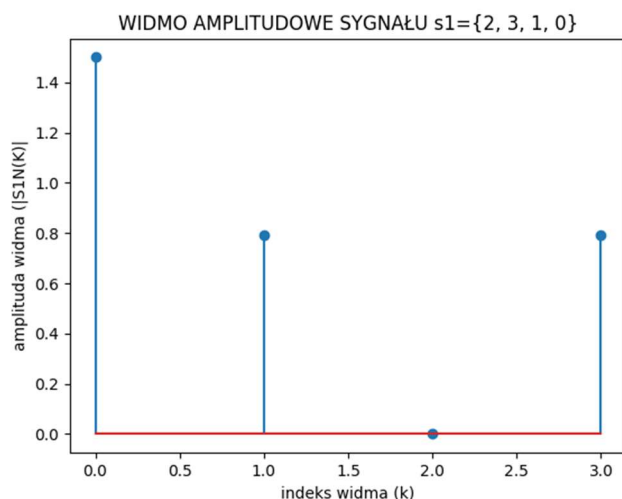
Dane są dwa sygnały o okresie podstawowym  $N = 4$ :  $s_1 = \{2, 3, 1, 0\}$  i  $s_2 = \{0, 3, 1, 0\}$ :

- Dla każdego sygnału wyznaczyć i wykreślić widmo amplitudowe i fazowe, obliczyć moc sygnału i sprawdzić słuszność twierdzenia Parsevala.
- Sprawdzić słuszność twierdzenia o dyskretnej transformacji Fouriera spłotu kołowego sygnałów  $s_1$  i  $s_2$ : wyznaczyć ręcznie spłot kołowy sygnałów  $s_1$  i  $s_2$ , a następnie wyznaczyć ten spłot ponownie za pomocą dyskretnej transformacji Fouriera.

UWAGA: ROZWIĄZANIE POLECENIA NR 1 ZNAJDUJE SIĘ W PLIKU **ex1.py**.

## 1. WYZNACZENIE WIDM SYGNAŁÓW

W rozwiązaniu skorzystałem z funkcji **fft()** z modułu **np.fft**, obliczającej szybką transformatę Fouriera. Następnie wyniki podzieliłem przez długość okresu  $N$ , aby wzór był zgodny z przekształceniem **DFT** prezentowanym na wykładzie. Następnie skorzystałem z funkcji **np.abs()** oraz **np.angle()** w celu otrzymania odpowiednio: widma amplitudowego i widma fazowego sygnałów. W wyniku tego działania otrzymałem 4 następujące wykresy:



UWAGA: należy pamiętać, że każde z powyższych widm jest **4-okresowe**.

## 2. Sprawdzenie twierdzenia Parsevala

Twierdzenie Parsevala określa, że moc średnia dla okresowego sygnału czasu dyskretnego może zostać obliczona jako suma widm amplitudowych podniesionych do kwadratu (widm mocy):

$$\frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 = \sum_{k=0}^{N-1} |X(k)|^2$$

Dlatego też w pliku z rozwiązaniem powstały dwie funkcje: **px\_per\_sig\_disc(sig, periodN)** oraz **parseval\_dft(sig)** obliczające odpowiednio lewą i prawą stronę powyższego równania:

```
Moc s1 na podstawie sygnału: 3.5
Moc s1 na podstawie widma mocy: 3.5
Moc s2 na podstawie sygnału: 2.5
Moc s2 na podstawie widma mocy: 2.5
```

Jak można zauważyć porównując wyniki, otrzymane wyniki się zgadzają co potwierdza słuszność twierdzenia Parsevala.

## 3. Sprawdzenie twierdzenia o DFT splotu kołowego sygnałów

Należy udowodnić, że:

$$\frac{1}{N} s_1(n) \otimes s_2(n) \stackrel{DFT}{\Longleftrightarrow} S_1(k) S_2(k)$$

W tym celu najpierw obliczona została ręcznie wartość  $s_1(n) \otimes s_2(n)$  z użyciem wzoru  $s_1(n) \otimes s_2(n) = \sum_{m=0}^{N-1} s_1(m) s_2(n-m)_N$ :

- dla  $n = 0$ :  $s_1(0) \otimes s_2(0) = s_1(0)s_2(0) + s_1(1)s_2(-1) + s_1(2)s_2(-2) + s_1(3)s_2(-3) = 2 * 0 + 3 * 0 + 1 * 1 + 0 * 0 = 1$
- dla  $n = 1$ :  $s_1(1) \otimes s_2(1) = s_1(0)s_2(1) + s_1(1)s_2(0) + s_1(2)s_2(-1) + s_1(3)s_2(-2) = 2 * 3 + 3 * 0 + 1 * 0 + 0 * 1 = 6$
- dla  $n = 2$ :  $s_1(2) \otimes s_2(2) = s_1(0)s_2(2) + s_1(1)s_2(1) + s_1(2)s_2(0) + s_1(3)s_2(-1) = 2 * 1 + 3 * 3 + 0 * 1 + 0 * 0 = 11$
- dla  $n = 3$ :  $s_1(3) \otimes s_2(3) = s_1(0)s_2(3) + s_1(1)s_2(2) + s_1(2)s_2(1) + s_1(3)s_2(0) = 2 * 0 + 3 * 1 + 1 * 3 + 0 * 0 = 6$

Następnie sploty podzielono przez długość okresu **N**, a wynikiem jest zbiór:  $\{\frac{1}{4}, \frac{3}{2}, \frac{11}{4}, \frac{3}{2}\}$ .

Wartość tą należy teraz porównać z przemnożonymi przez siebie widmami **S<sub>1</sub>(k)** i **S<sub>2</sub>(k)**, następnie iloczyn ten przekształcić z użyciem **np.fft.ifft()** oraz przemnożyć wynik przez **N** (dla uzgodnienia wzoru na **DFT**). Otrzymane wyniki:

```
s[0] = (0.25+0j)
s[1] = (1.5+0j)
s[2] = (2.75+0j)
s[3] = (1.5+0j)
```

Wyniki te (porównane z obliczeniami) potwierdzają słuszność powyższego twierdzenia.

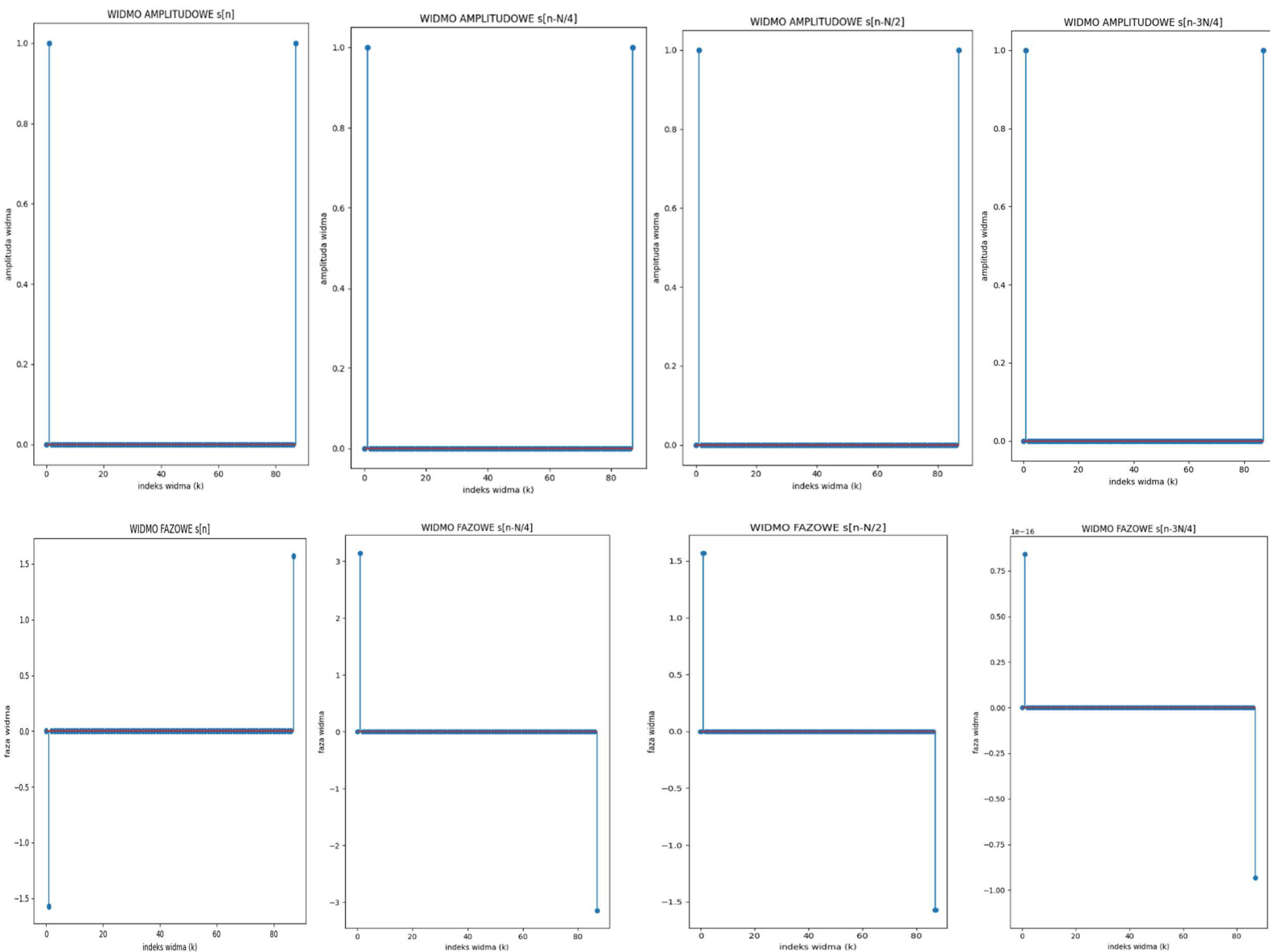
## II. POLECENIE II.

Zbadać wpływ przesunięcia w czasie na postać widma amplitudowego i widma fazowego dyskretnego sygnału harmonicznego  $s[n] = A \sin(2\pi \frac{n}{N})$  o amplitudzie  $A = 2$  i okresie podstawowym  $N = 88$ . W tym celu dla każdej wartości  $n_0 \in \{0, \frac{N}{4}, \frac{N}{2}, \frac{3N}{4}\}$  wykreślić widmo amplitudowe i fazowe przesuniętego sygnału  $s[n - n_0]$ . Skomentować otrzymane wyniki.

UWAGA: ROZWIĄZANIE POLECENIA NR 2 ZNAJDUJE SIĘ W PLIKU **ex2.py**.

### 1. Implementacja

W rozwiązaniu została zaimplementowana funkcja **signal()** wyliczająca wartość funkcji  $s[n] = A \sin(2\pi \frac{n-n_0}{N})$ . Za jej pomocą zostały policzone wartości przesuniętego sygnału  $s[n - n_0]$ , a następnie przetworzone z użyciem funkcji **np.fft.fft()** wraz z podzieleniem przez długość okresu **N**. W wyniku otrzymane zostały następujące wykresy:



## 2. Wnioski

- Na podstawie otrzymanych wykresów amplitud widm można zauważyć, że **przesunięcie w czasie zupełnie nie wpływa na charakterystykę widma amplitudowego danego sygnału**.
- Zmiany można zauważyć zaś w przypadku widm fazowych, mianowicie **zmieniają się wartości faz w zależności od przesunięcia**. Jako, że zbiór wartości widma fazowego to  $(-\pi, \pi]$ , tak więc można wywnioskować, że każde z kolejnych przesunięć zmniejsza – w przypadku takich okresów - wartość fazy o kolejno  $\{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$  dla przesunięć  $\{\frac{N}{4}, \frac{N}{2}, \frac{3N}{4}\}$  dla dwóch skrajnych prążków (u pozostałych by również było to widoczne, jednak w programie zostało zastosowane zerowanie wartości w wektorze transformat dla tych elementów, których amplituda była bardzo bliska 0 w celu **odszumienia widm fazowych**).
- Wnioski te można wysunąć również wzorując się na wzorze:

$$x(n - n_0)_N \xLeftrightarrow{DFT} X(k) e^{-\frac{2\pi j k n_0}{N}},$$

widać tu mianowicie, że przesunięcie powoduje zmianę argumentu liczby zespolonej, a moduł się nie zmienia, co zgadza się z wysuniętymi wnioskami na podstawie wykresów.

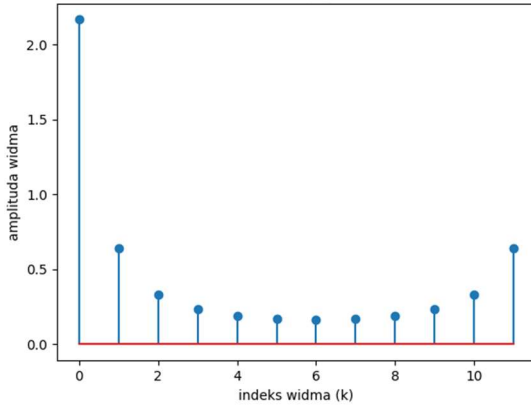
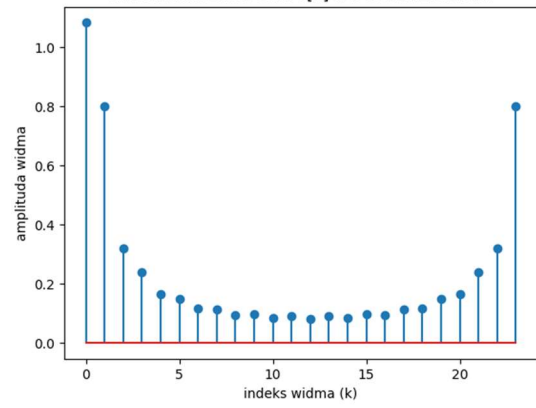
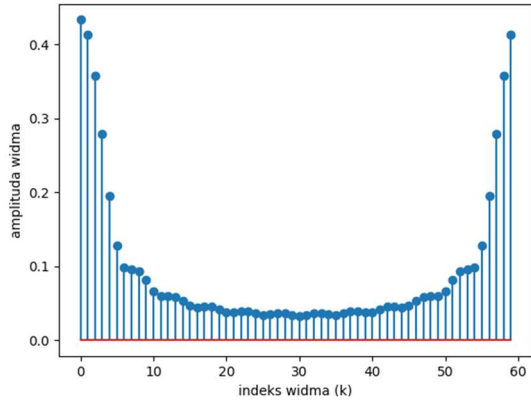
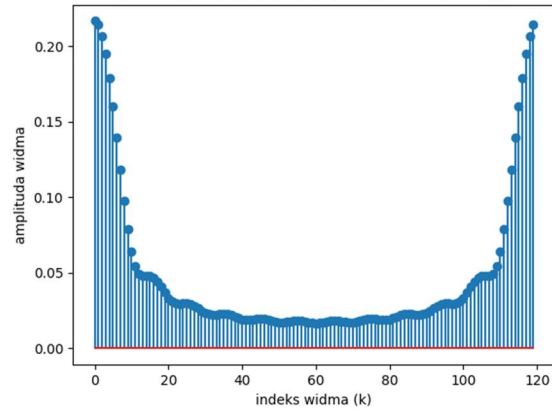
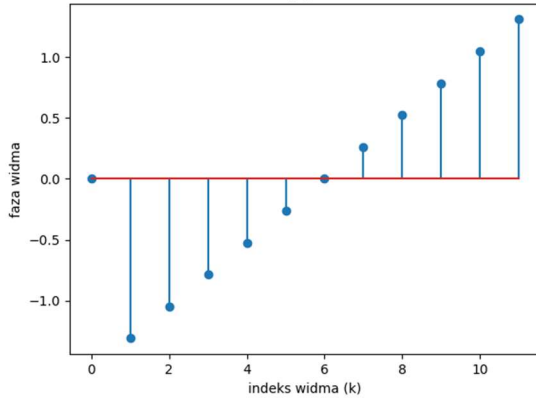
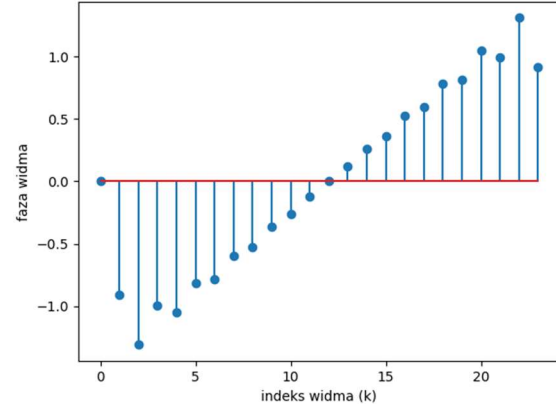
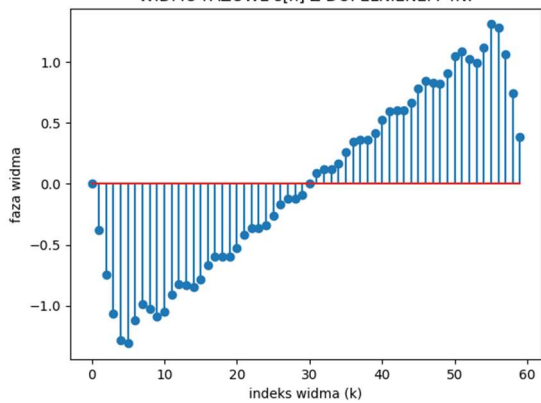
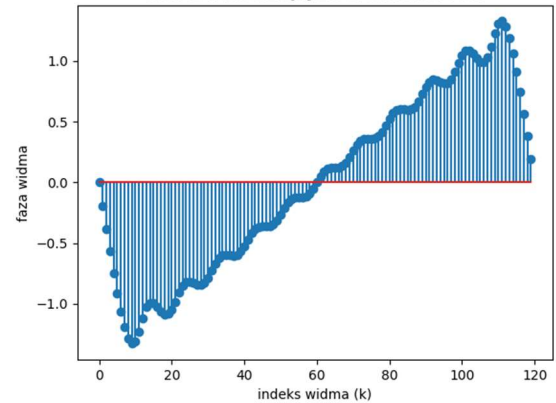
## III. Polecenie III.

Zbadać wpływ dopełnienia zerami na postać widma amplitudowego i widma fazowego dyskretnego sygnału  $s[n] = A(1 - \frac{n \bmod N}{N})$  o amplitudzie  $A = 4$  i okresie podstawowym  $N = 12$ . W tym celu dla każdej wartości  $N_0 \in \{0, 1N, 4N, 9N\}$  wykreślić widmo amplitudowe i fazowe sygnału  $s[n]$  dopełnionego  $N_0$  zerami. Skomentować otrzymane wyniki.

UWAGA: ROZWIĄZANIE POLECENIA ZNAJDUJE SIĘ W PLIKU **ex3.py**.

### 1. Implementacja:

W rozwiązaniu została zaimplementowana funkcja **signal()** wyliczająca wartość funkcji  $s[n] = A(1 - \frac{n \bmod N}{N})$  oraz funkcja **normalize()**, która „zeruje” każdy składnik wektora transformat, dla którego amplituda jest bardzo bliska 0 (w celu odszumienia widma fazowego). Funkcja **signal()** została wywołana dla 12 pierwszych próbek, tworząc wektor próbek, który następnie w kolejnych etapach jest dopełniany odpowiednią liczbą zer. Następnie została zastosowana transformata **fft()** z biblioteki **np.fft**, wyniki zostały podzielone przez nowe długości okresów, a wektory transformat znormalizowane wg amplitudy widma. Końcowym wynikiem jest zbiór wykresów opisujący widma amplitud i faz dla odpowiednio dopełnionych zerami sygnałów.

WIDMO AMPLITUDOWE  $s[n]$  BEZ DOPEŁNIENIA:WIDMO AMPLITUDOWE  $s[n]$  Z DOPEŁNIENIEM N:WIDMO AMPLITUDOWE  $s[n]$  Z DOPEŁNIENIEM 4N:WIDMO AMPLITUDOWE  $s[n]$  Z DOPEŁNIENIEM 9N:WIDMO FAZOWE  $s[n]$  BEZ DOPEŁNIENIA:WIDMO FAZOWE  $s[n]$  Z DOPEŁNIENIEM N:WIDMO FAZOWE  $s[n]$  Z DOPEŁNIENIEM 4N:WIDMO FAZOWE  $s[n]$  Z DOPEŁNIENIEM 9N:

## 2. Wnioski

- W tym przypadku zastosowanie dopełnienia zerami ma **znaczący wpływ** zarówno na postać **widma fazowego** jak i **widma amplitudowego** dyskretnego sygnału.
- Można zauważyć, że dopełnienie zerami znacząco **zwiększyło okresy** zarówno sygnału jak i jego widm (kolejno do  $N=12, 24, 60, 120$ ).
- Na podstawie wykresów można zauważyć, że uzupełnienie dodatkowymi zerami sygnału **zmniejsza różnicę pomiędzy kolejnymi wartościami widma** i **lepiej uwypukla szczegóły widma** (kształt widma się nie zmienia w okresie, rozszerza się on tylko po osi poziomej).
- Własność ta na pewno mogłaby być wykorzystana do rozszerzania liczby próbek sygnału dyskretnego do potęgi liczby 2, aby móc zastosować go do obliczeń algorytmu FFT.
- W tych przypadkach doszło również do zmniejszenia amplitud widma, jest to jednak związane ze zwiększeniem okresów kolejnych sygnałów i zastosowanie dzielenia przez ich długość przy transformacie.

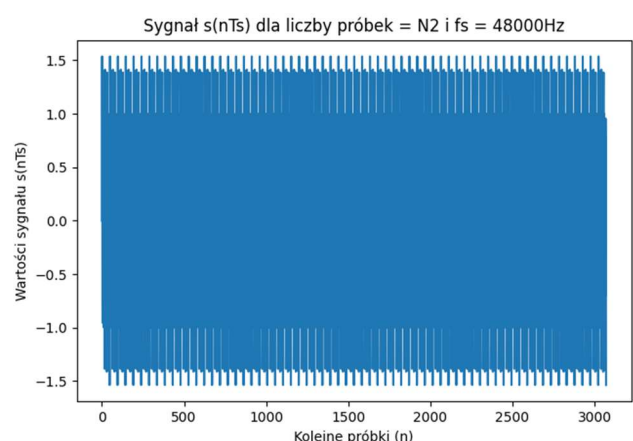
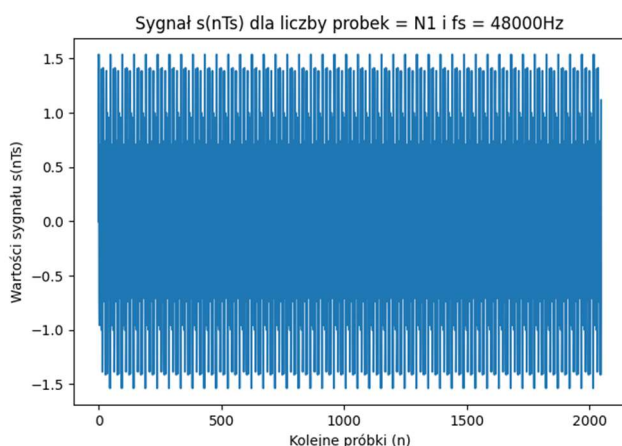
## IV. POLECENIE IV.

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  $A_1 = 0.1$ ,  $f_1 = 3000\text{Hz}$ ,  $A_2 = 0.7$ ,  $f_2 = 8000\text{Hz}$ ,  $A_3 = 0.9$ ,  $f_3 = 11000\text{Hz}$ . Przy założeniu, że 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.

UWAGA: rozwiązanie zadania znajduje się w pliku **ex4.py**.

### 1. Implementacja

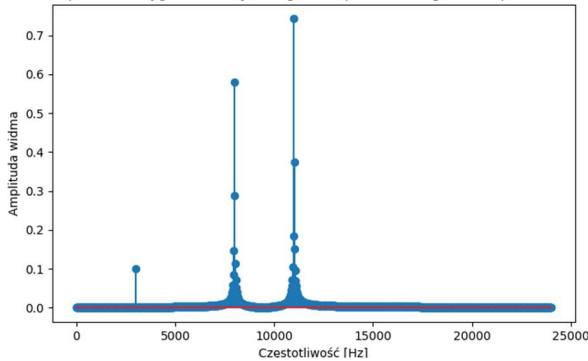
W rozwiązaniu została zaimplementowana funkcja **signal()** zwracająca wartość sygnału **s(t)**. Dodatkowo powstała funkcja **perform\_example()**, w której zostały wykonane niezbędne badania dla liczby próbek wynoszącej kolejno **N1** oraz **N2**. W celu wyznaczenia spróbkowanego sygnału **zbiór indeksów** próbek został **podzielony** przez **częstotliwość próbkowania**, a także wyznaczony został **wektor wartości sygnałów** dla kolejnych próbek. Następnie zostały narysowane **wykresy reprezentujące sygnały spróbkowane**:



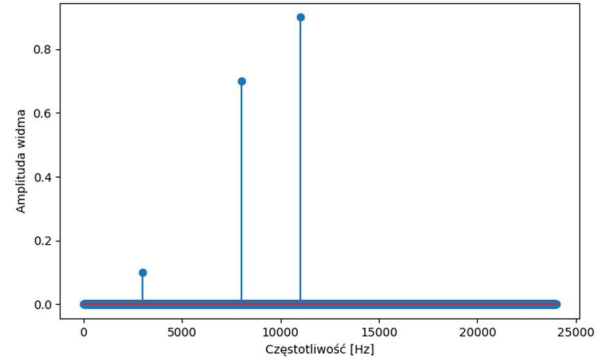


Następnie użyto funkcji `rfft()` oraz **podzielono wynik przez połowę liczby próbek** (widmo jest N-okresowe, jednak dla funkcji `rfft()` obserwowana jest tylko połowa widm, odrzucamy kopie będące lustrzanymi odbiciami) w celu wyznaczenia **widma rzeczywistego**. Następnie wyznaczone zostały **widma amplitudowe** sygnałów. Dodatkowo koniecznym było **przeskalowanie osi na oś częstotliwości**. Gwarantuje to funkcja `rfftfreq()` zwracająca zbiór częstotliwości dla zadanej liczby próbek i odległości w czasie między próbkami. Następnie zostały narysowane **wykresy widm amplitudowych** dla sygnałów:

Widmo amplitudowe sygnału rzeczywistego s(t) spróbkowanego na N1 próbek z fs=48kHz

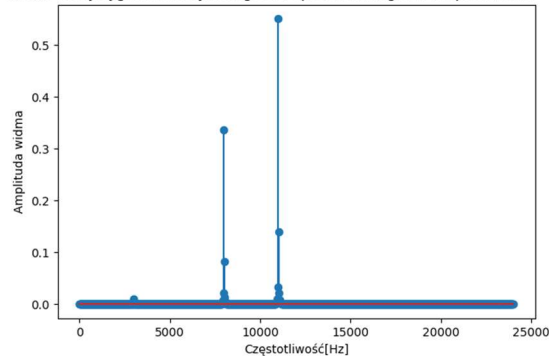


Widmo amplitudowe sygnału rzeczywistego s(t) spróbkowanego na N2 próbek z fs=48kHz

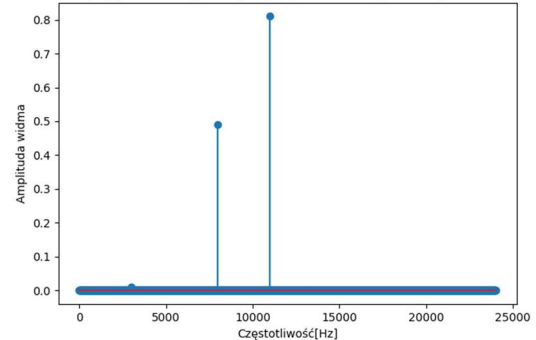


Dodatkowo zostały wykreślone **wykresy widm gęstości mocy** (poprzez podniesienie widm amplitudowych do kwadratu):

Widmo mocy sygnału rzeczywistego s(t) spróbkowanego na N1 próbek z fs=48kHz



Widmo mocy sygnału rzeczywistego s(t) spróbkowanego na N2 próbek z fs=48kHz



## 2. Wnioski:

- Na podstawie wykresów można zauważyć, że widma – w przypadku liczby próbek =  $N_2$  – zgadzają się z sygnałem rzeczywistym  $s(t)$  – na odpowiednich częstotliwościach, będących częstotliwościami sinusów składających się na sygnał, występują prążki widmowe, których amplitudy są równe amplitudom tych składowych sygnału, tzn. dla częstotliwości **3kHz, 8kHz** oraz **11kHz** wartości prążków widmowych, zgodnie z sygnałem  $s(t)$ , wynoszą kolejno **0.1, 0.7** oraz **0.9**.
- Wartości tych prążków nie są jednak idealne w przypadku liczby próbek =  $N_1$ . Można zauważyć, sugerując się widmem mocy, że moc zawarta w prążkach rzeczywistych dla sygnału „rozlała się” na prążki sąsiadujące z nimi. Efekt ten, zwany „**przeciekaniem widma**”, pojawił się tylko w przypadku próbkowania dla liczby próbek =  $N_1$ , co spowodowało „nieidealną” amplitudę tych prążków.
- Przeciek widma jest efektem ucięcia okresów składowych sygnału analogowego przez blok N próbek. Wystąpienie przecieku widma w przypadku 1. i jego brak w przypadku 2. można więc udowodnić odpowiednimi obliczeniami. Przeciek widma nie będzie występował w przypadku tego próbkowania, gdzie czas trwania próbkowanego sygnału będzie wielokrotnością czasu trwania okresów składowych tego sygnału. Tak więc wystarczy podzielić iloraz liczby próbek i częstotliwości próbkowania przez okresy sinusów składających się na sygnał, co realizuje funkcja `periods()`, jej wyniki wskazują na słuszność powyższych punktów (brak liczb całkowitych dla  $N_1$ ):

```
Liczba okresów sinusa o f=3000Hz dla 2048 probek wynosi: 128.0
Liczba okresów sinusa o f=8000Hz dla 2048 probek wynosi: 341.33
Liczba okresów sinusa o f=11000Hz dla 2048 probek wynosi: 469.3
Liczba okresów sinusa o f=3000Hz dla 3072 probek wynosi: 192.0
Liczba okresów sinusa o f=8000Hz dla 3072 probek wynosi: 512.0
Liczba okresów sinusa o f=11000Hz dla 3072 probek wynosi: 704.0
```