

Raport - Dyskretna transformata Fouriera

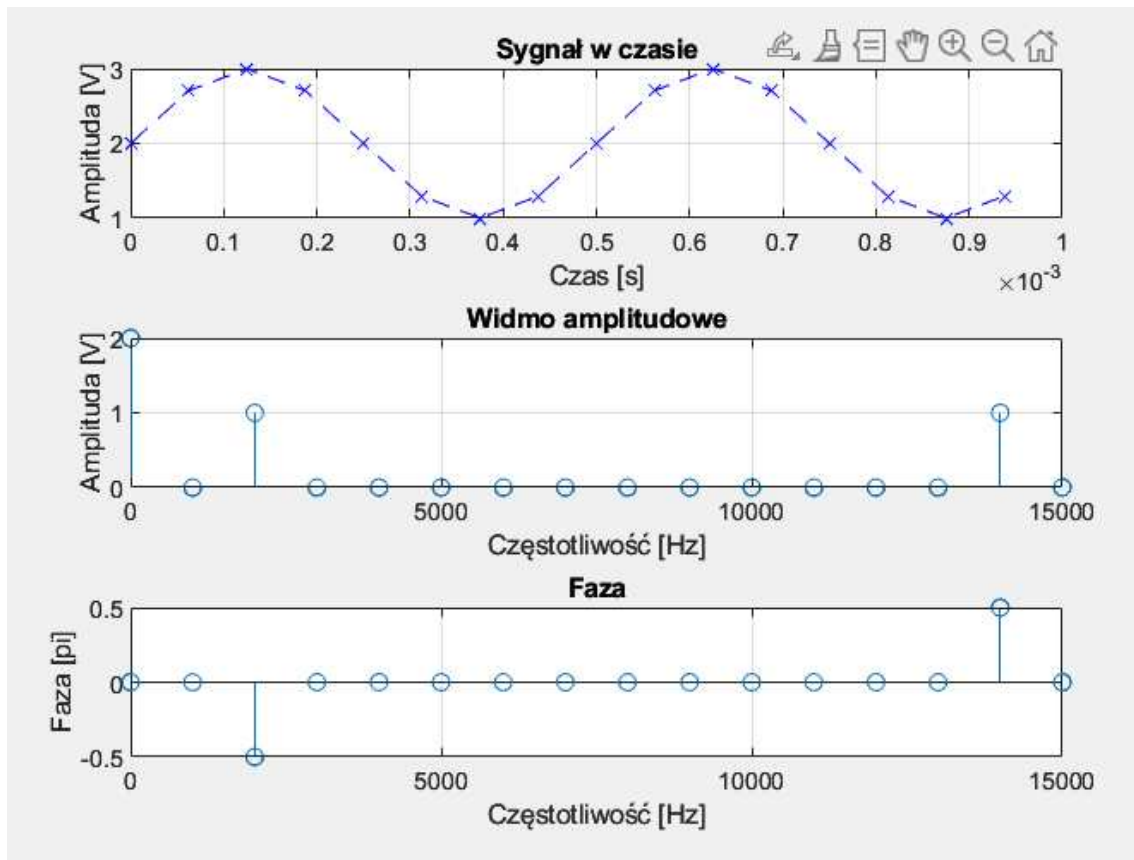
Cyfrowe przetwarzanie sygnałów

Mateusz Kulig

Semestr letni 2022/2023

1 Zadanie 1.1

1.1 1)



Rys. 1: Wykresy w przypadku sinusa ze składową stałą. Oś x została przedstawiona w skali częstotliwości wyrażonej w [Hz], a wykresy zostały unormowane.

1.2 2)

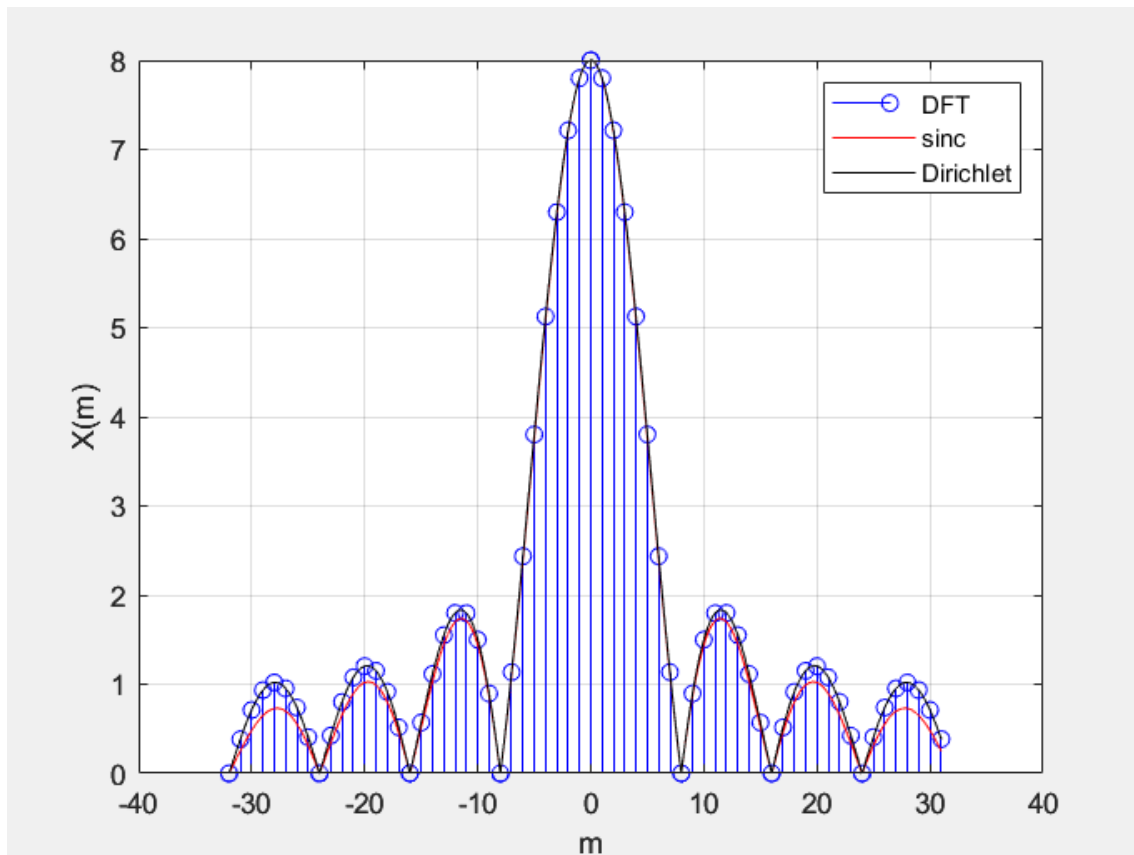
Wzór na częstotliwość bazową to $F_{base} = F_{sig}/2$, zatem częstotliwość bazowa użyta w ćwiczeniu to $\frac{2000}{2} = 1000$. Wzory na unormowaną amplitudę to $amplitudeNorm = amplitude/(N/2)$ oraz w punkcie 0 $amplitudeNorm = amplitude/N$.

1.3 3)

Przesunięcie fazy dla funkcji sinus wynosi $0,5\pi$, a dla kosinus wynosi $2.76653 \cdot 10^{-16}\pi$. Dzieje się tak ponieważ w postaci wykładniczej liczby zespolonej to funkcja kosinus ma zerową fazę. Wynika to ze wzoru $e^{jx} = \cos(x) + j\sin(x)$.

2 Zadanie 2.1

2.1 1)



Rys. 2: Wykres dla parametrów $K = 8$ oraz $n_0 = -5$.

Wynik DFT pokrywa się idealnie z jądrem Dirichlet'a na całym przedziale. Funkcja *sinc* pokrywa się idealnie na najwyższym pikie, ale na pikach bocznych zaczyna odstawać od DFT, im dalej tym bardziej.

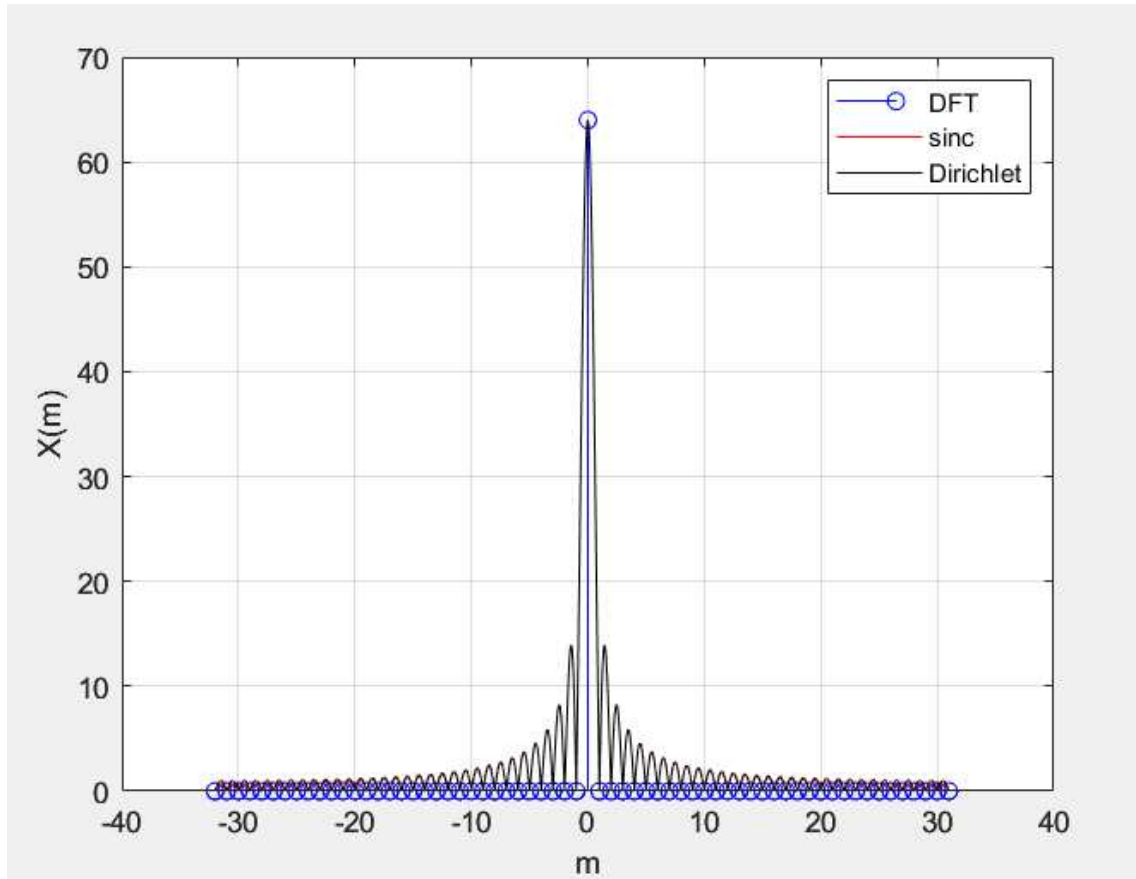
2.2 2)

Odczytane z wykresu położenie pierwszego minimum powyżej piku głównego wynosi $m = 8$ i zgadza się z wartością teoretyczną równą $m_0 = \frac{N}{K} = \frac{64}{8} = 8$.

2.3 3)

Współczynnik n_0 występuje tylko w członie eksponencjalnym jądra Dirichlet'a, a zatem wpływa tylko na fazę, widmo amplitudowe jest od niego niezależne.

2.4 4)

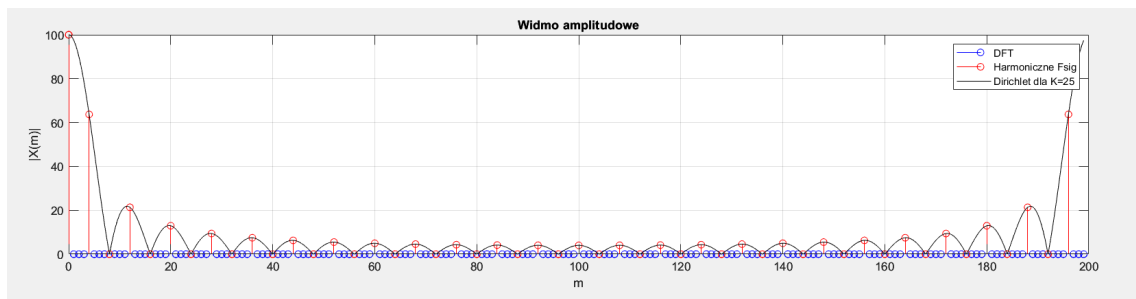


Rys. 3: Wykres dla parametrów $K = N$ oraz $n_0 = \frac{-(K-1)}{2}$.

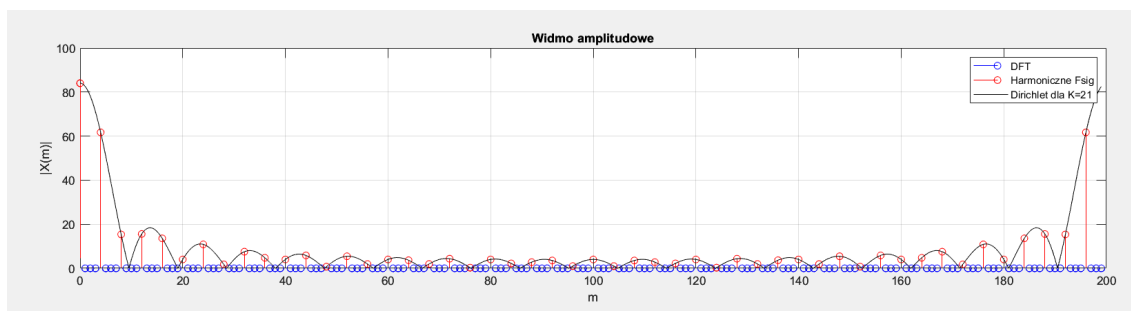
Położenie miejsc zerowych to $m = \pm 1, \pm 2, \pm 3, \pm 4, \dots$ i zgadza się to z teorią ponieważ, dla wartości współczynników $K = N$ oraz $n_0 = \frac{-(K-1)}{2}$ wzór na jądro Dirichlet'a przyjmuje postać $X(m) = \frac{\sin(m\pi)}{\sin(\pi m/N)}$, a funkcja ta ma miejsca zerowe równe $m_0 = \pm 1, \pm 2, \pm 3, \pm 4, \dots$.

3 Zadanie 2.3

3.1 1)



Rys. 4: Wykres dla parametrów $K = 25$.



Rys. 5: Wykres dla parametru $K = 21$.

3.2 2)

Wzór który pozwoli obliczyć położenie harmonicznych sygnału to $X(m) = \sin(\pi m K / N)$

3.3 3)

Dla $K = 25$ położenia wszystkich niezerowych prążków w widmie sygnału prostokątnego w zakresie indeksów $[0-40]$ to $m = 0, 4, 12, 20, 28, 36$, a dla $K = 21$ to $m = 0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40$

3.4 4)

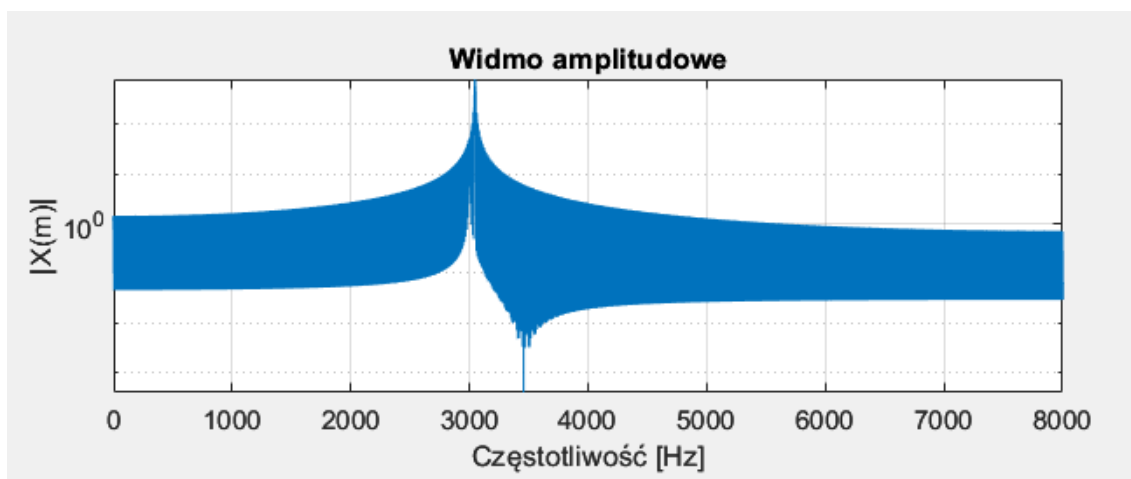
W obu przypadkach obserwujemy $m = 0, 4, 12, 20, 28, 36$ co zgadza się z wzorem i wartościami współczynników dla danego sygnału.

4 Zadanie 6.1

4.1 1)

Wartość zeroPadding dla której położenie maksimum widma zaczyna odpowiadać zadanej wartości $F_{sig2} = 3047[\text{Hz}]$ to 1000000. Wówczas maksimum przypada na 3047,002[Hz].

4.2 2)



Rys. 6: Wykres dla parametru $zeroPadding = 1000000$.

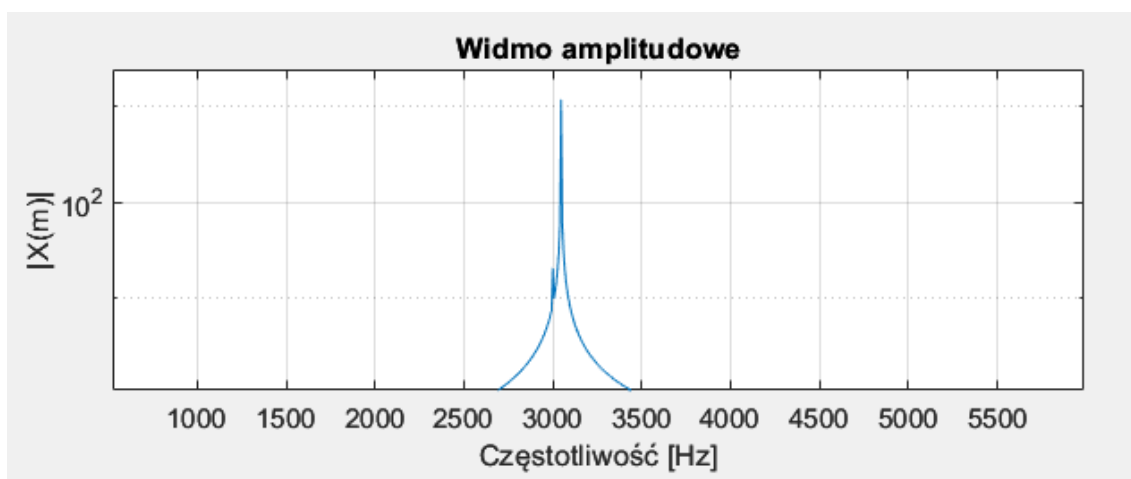
4.3 3)

Zero padding pomaga lepiej zlokalizować pik dla częstotliwości powodującej wyciek DFT ponieważ zwiększając ilość zer zaczynamy dokładnie widzieć pełne widmo sygnału obłożonego prostokątnym oknem jednostkowym.

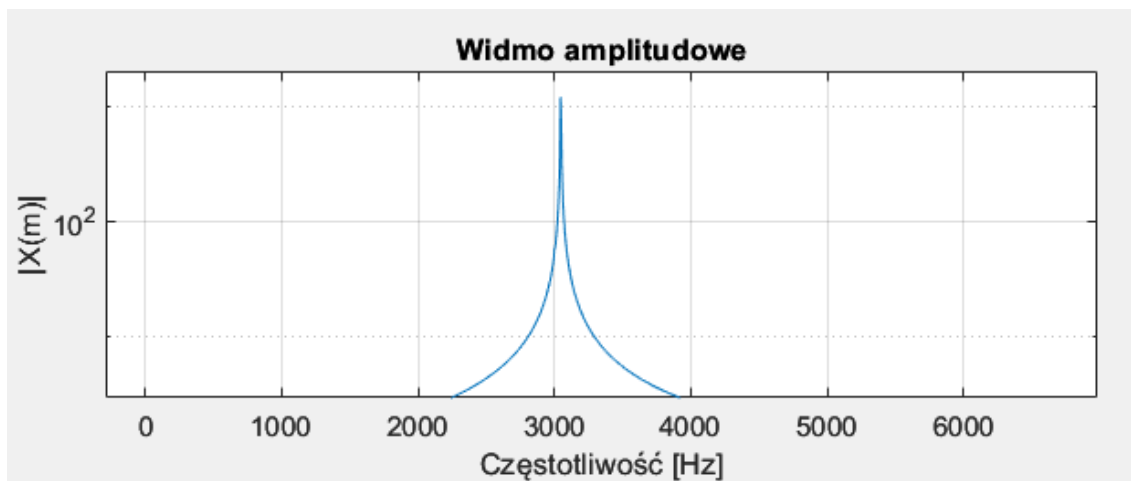
4.4 4)

Zero padding nie pomógł wyłuskać częstotliwość f_{sig1} z widma. Dzieje się tak dlatego, że zero padding umożliwia obserwację pełnego widma częstotliwościowego ale nie pomaga w zwiększeniu rozdzielczości częstotliwościowej w sensie odróżnienia od siebie blisko leżących składowych częstotliwościowych sygnału.

4.5 5)



Rys. 7: Wykres dla parametru $N = 2400$. W tym przypadku jesteśmy w stanie dostrzec pik 3 kHz.



Rys. 8: Wykres dla parametru $N = 3200$. W tym przypadku nie jesteśmy w stanie dostrzec piku 3 kHz.

W ogólności zwiększenie N powinno pomóc, bo sygnał danego tonu związany jest z jedną precyzyjnie określoną częstotliwością, więc amplituda DFT dla tej częstotliwości rośnie proporcjonalnie do ilości próbek N .

Dla $N=2400$ mamy $f_{sig2}/f_{base} = 3047/6,66666 = 457$ natomiast dla $N=3200$ $f_{sig2}/f_{base} = 3047/5 = 609,4$. Jeżeli jądro Dirichlet'a ma maksimum dla liczby całkowitej m to wszystkie kolejne wartości będą równe

zero. Jeśli nie to wyciek przykryje szukaną częstotliwość i dlatego nie możemy zaobserwować dodatkowego piksu dla wartości $N = 3200$

5 Zadanie 7.1

5.1 1)



Rys. 9: Wykres dla parametru $filterType = 1$.



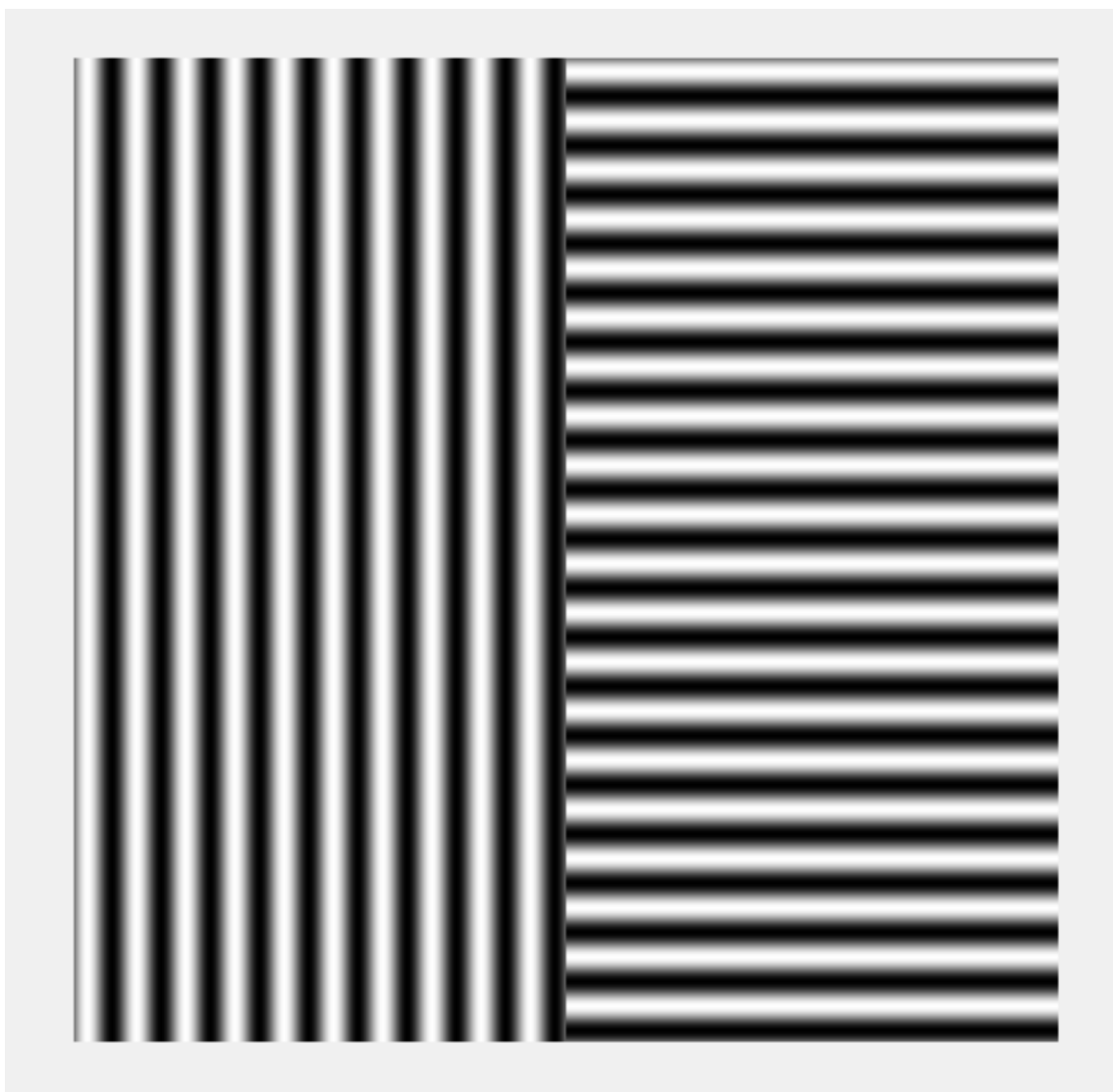
Rys. 10: Wykres dla parametru $filterType = 2$.

5.2 2)

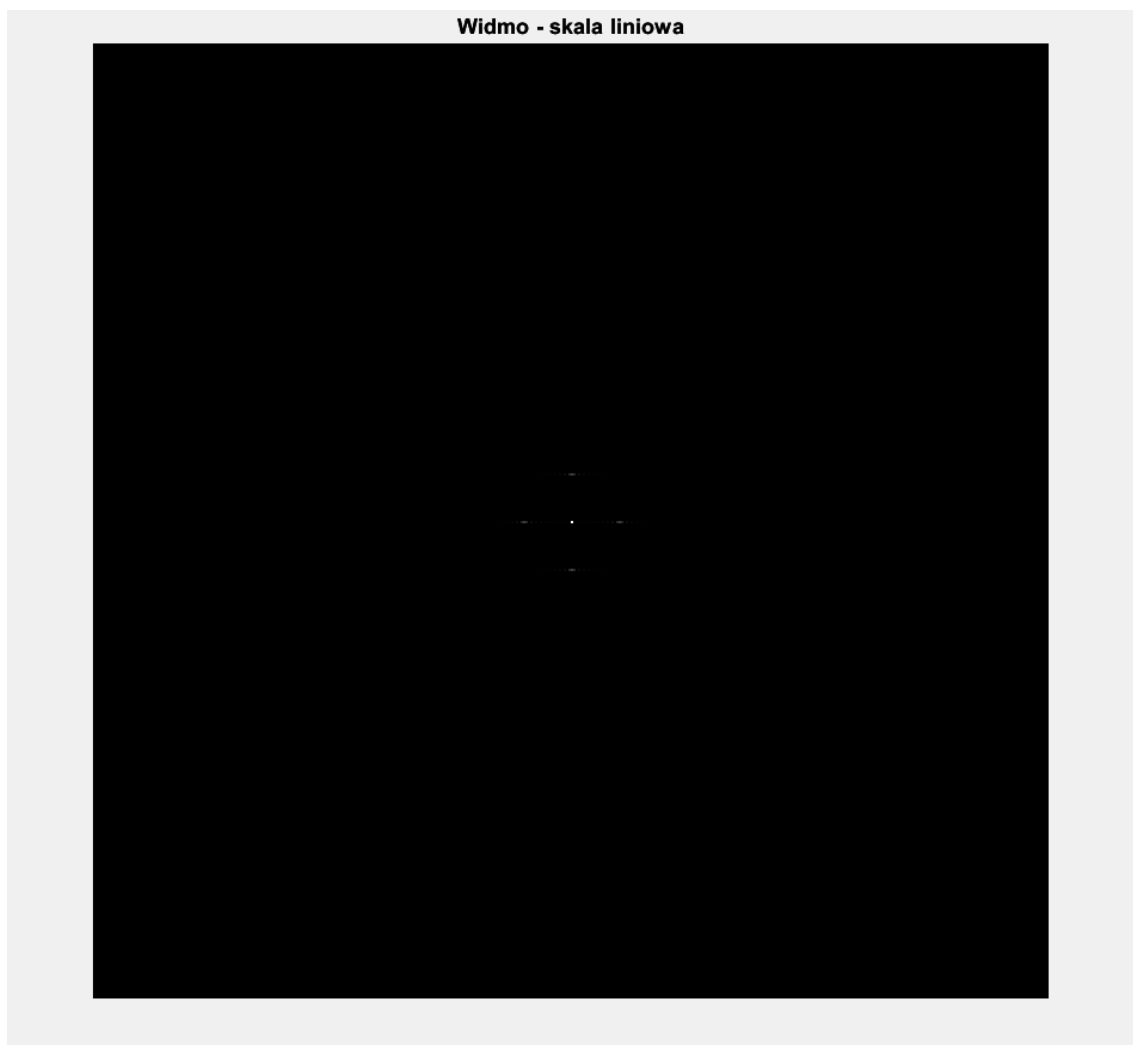
W pierwszym przypadku zaimplementowaliśmy filtr dolnoprzepustowy. Wycina on wysokie częstotliwości i przez to zrekonstruowany sygnał nie ma tak stromych krawędzi jak sygnał oryginalny. W drugim przypadku zaimplementowaliśmy filtr górnoprzepustowy. Wycina on niskie częstotliwości i przez to zrekonstruowany sygnał na pierwszy rzut oka pozbawiony jest swojej schodkowej natury.

6 Zadanie 8.3

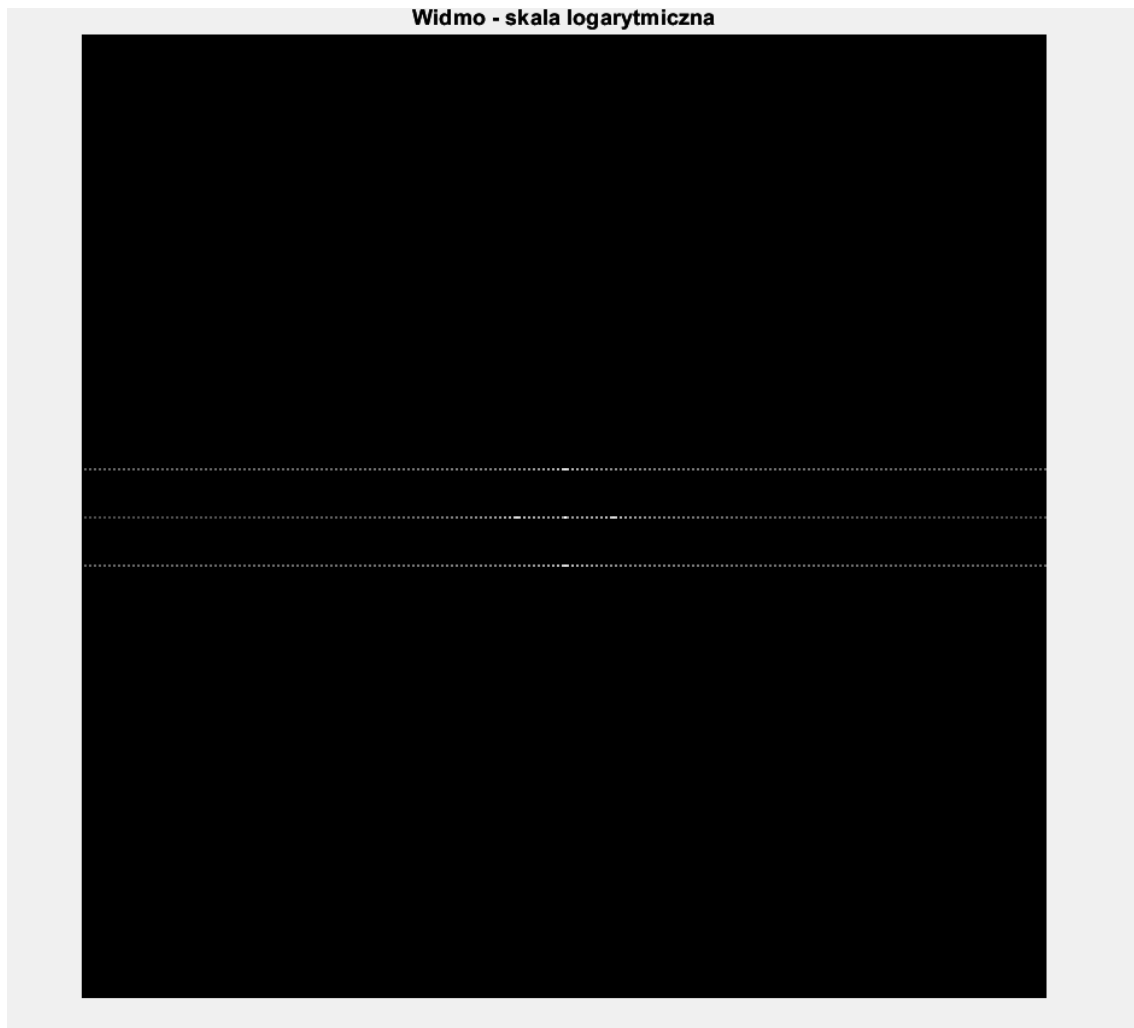
6.1 1)



Rys. 11: Rysunek przedstawiający pionowe i poziome linie w dwóch połówkach obrazka.



Rys. 12: Widmo powyższego obrazka z skali liniowej.



Rys. 13: Widmo powyższego obrazka z skali logarytmicznej.

Na widmie w skali logarytmicznej widzimy, że została złamana symetria pion-poziom w stosunku do skali liniowej. Dzieje się tak dlatego, że podczas wykonywania DFT musimy wybrać kierunek wykonywania. Zatem idąc od strony lewej do prawej najpierw będziemy obserwować szybkie zmiany koloru, a następnie praktycznie stały kolor. to właśnie powoduje, że na widmie logarytmicznym widzimy tylko poziome przerywane linie.

7 Zadanie 9.1

7.1 1)



Rys. 14: Zdjęcie dla parametru $filterType = 1$.



Rys. 15: Zdjęcie dla parametru $filterType = 3$.

7.2 2)

Zastosowanie filtru dolnoprzepustowego powoduje że zdjęcie jest rozmyte. Dzieje się tak ponieważ filtr dolnoprzepustowy nie przepuszcza wysokich częstotliwości, czyli szybkich zmian jasności na zdjęciu, co powoduje że widzimy obraz jako rozmyty.

7.3 3)



Rys. 16: Zdjęcie dla parametru $filterType = 1$ z zaznaczonym aliasingiem.

Na zdjęciu widać aliasing na niebie, a spowodowane jest to tym, że filtr typu 1 skokowo ucina wysokie częstotliwości.

7.4 4)

Tego efektu nie widać w przypadku filtru Butterwortha 2 rzędu, bo ten filtr ma gładkie brzegi. Efekt znów pojawia się dla filtru Butterwortha 20 rzędu, bo filtr Butterwortha 20 rzędu jest już na tyle stromy że praktycznie skokowy.

8 Zadanie 9.5

8.1 1)

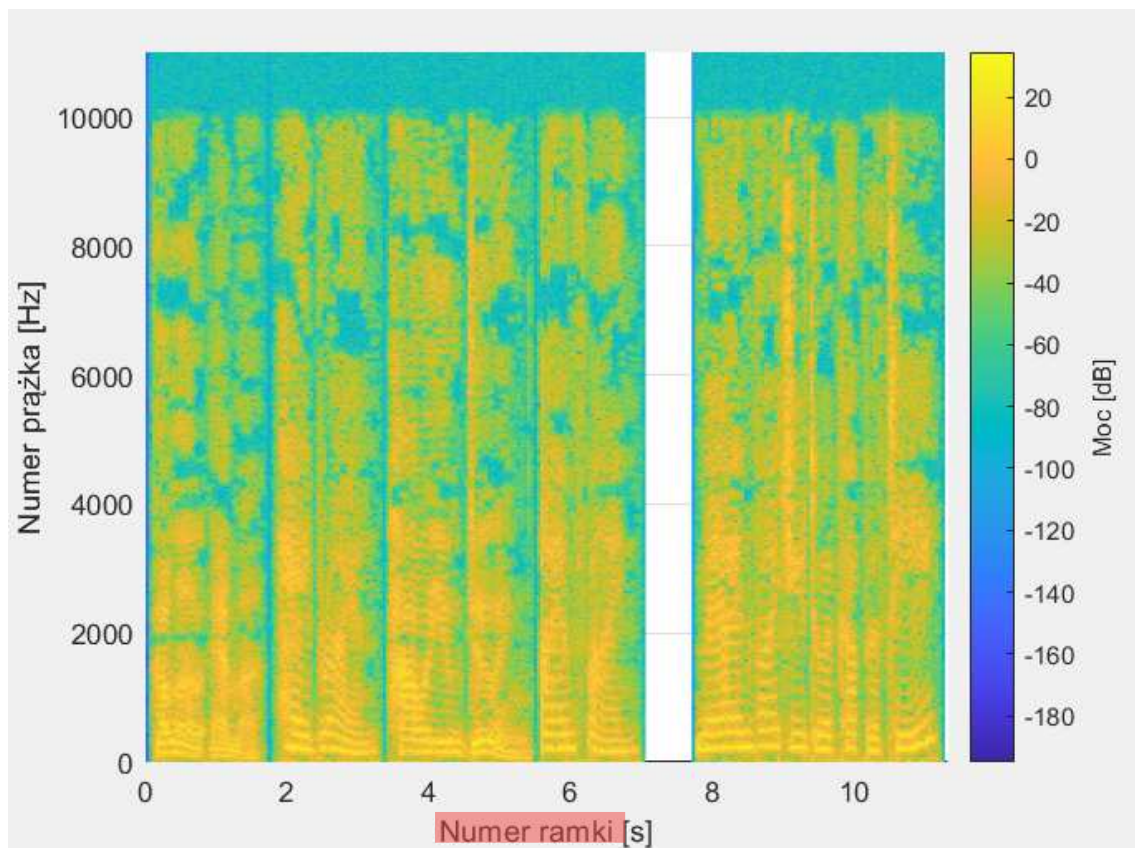
Moim zdaniem najlepsze efekty otrzymamy stosując parametry: $filterRatio = 0.5$, $butterFilterOrder = 2$, $sharpeningCoef = 0.8$



Rys. 17: Wyostrzone zdjęcie dla powyższego zestawu parametrów.

9 Zadanie 10.2

9.1 1)



Rys. 18: Spektrogram dla pliku dialog.mp3.

Powyższy spektrogram przedstawia jaki był wkład w sensie mocy od sygnałów o zadanej częstotliwości w danej chwili pliku dźwiękowego.

9.2 2)

Zwiększenie długości ramki powoduje, że po kompresji MDCT tworzą się bloki, ponieważ uśredniamy próbki z większego zakresu. **nie psuje to zbyt rozdzielczości częstotliwościowej natomiast bardziej psuje rozdzielczość czasową.**

10 Zadanie 10.4

10.1 1)

Wartość parametru quality dająca akceptowalny moim zdaniem rezultat to $quality = 20$. Wówczas współczynnik kompresji DCT = 99.34 % oraz współczynnik kompresji JPEG = 11.80 % .

10.2 2)



Rys. 19: Skompresowany obraz dla wartości quality=20.

Skompresowany obraz jest ogólnie wyblakły i lekko rozmazany, ale jest to akceptowalne w stosunku do oryginalnego obrazu.

10.3 3)

Tab. 1: Blok `dctImage(1:8, 1:8)`. Wszystkie wartości powinny być pomnożone przez 10^3 - wynika to z konwencji wypisywania bloków przez MATLAB.

1.3245	0.0220	0.0058	0.0002	-0.0003	0.0003	0.0003	-0.0001
0.0109	-0.0018	0.0003	-0.0000	0.0007	-0.0001	0.0000	0.0007
-0.0028	-0.0002	-0.0001	0.0002	0	0.0000	-0.0003	0.0004
-0.0007	-0.0003	0	0.0002	-0.0001	-0.0001	0.0005	-0.0004
0.0040	0.0004	-0.0002	-0.0001	0.0003	0.0006	0.0001	0.0003
-0.0004	-0.0003	0.0000	-0.0001	0.0004	0.0003	0.0004	0.0001
0.0004	-0.0001	-0.0001	0.0001	0	0.0000	-0.0001	0.0001
0.0001	-0.0003	0.0001	0.0002	-0.0002	-0.0003	-0.0000	-0.0002

Tab. 2: Blok dctQuantImage(1:8, 1:8).

33	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Kompresja JPEG działa następująco: najpierw dzielimy obraz na mniejsze bloki, potem wykonujemy DCT każdego bloku z osobna. Następnie bloki mnożymy przez odpowiednią macierz i wartości w komórkach bloków przyrównujemy do zera. Na końcu dokonujemy odwrotnej transformaty. Kompresja JPEG odpowiada w przybliżeniu filtracji dolnoprzepustowej.

10.4 4)

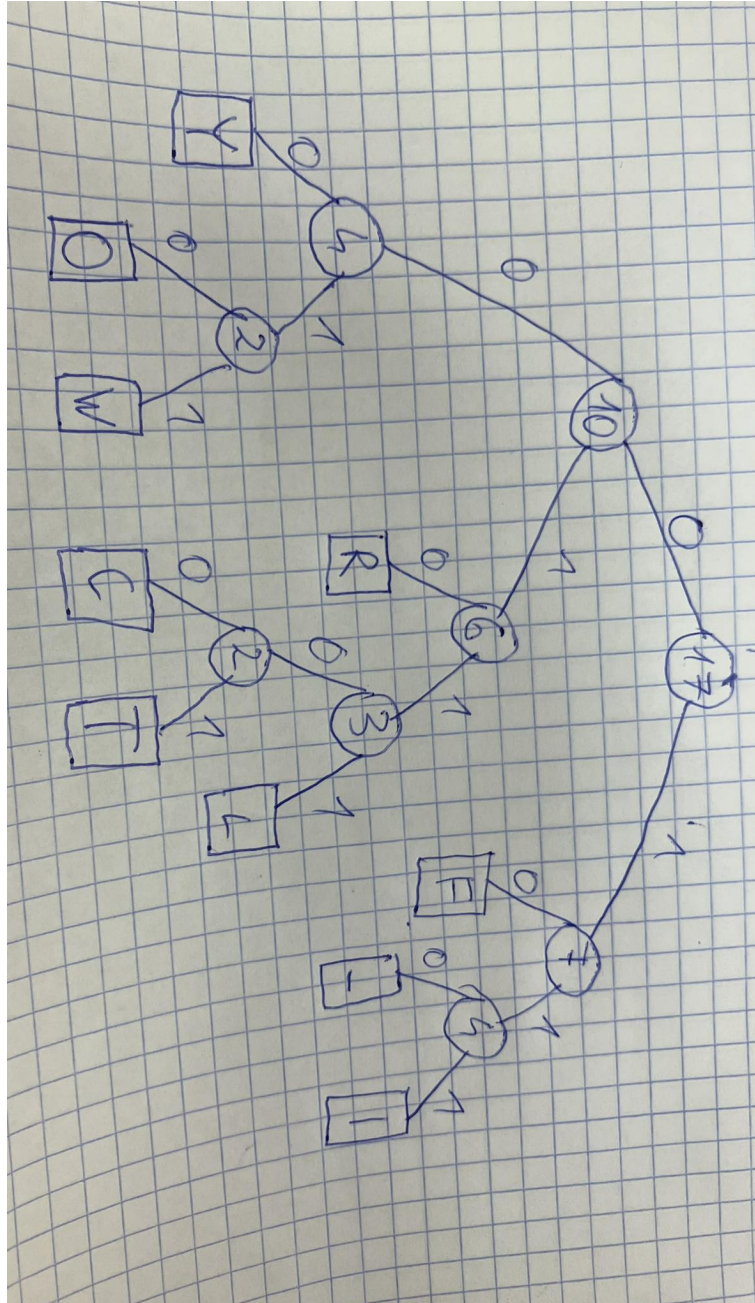
Obraz poddany kompresji jest gorszej jakości niż obraz oryginalny właśnie z tego powodu, że kompresja JPEG odpowiada w przybliżeniu filtracji dolnoprzepustowej - celowo tracimy niektóre informacje co powoduje pogorszenie jakości(rozmazanie, mniej żywe kolory), ale również i zmniejszenie rozmiaru.

11 Zadanie 10.5

Wybrane przeze mnie zdanie to „FILTR CYFROWY FIR”. Ma ono 17 znaków.

Tab. 3: Ilość wystąpień poszczególnych znaków w wybranym zdaniu.

Znak	Ilość wystąpień
F	3
R	3
I	2
—	2
Y	2
L	1
T	1
C	1
O	1
W	1



Rys. 20: Drzewo Huffman'a dla wybranego zdania

Zakodowane zdanie : 10 111 0111 01101 010 110 01100 000 10 010 0010 0011 000 110 10 111 010.
Średnia ilość bitów na znak to

$$H = \frac{3}{17} \cdot 2 + \frac{3}{17} \cdot 3 + \frac{2}{17} \cdot 3 + \frac{2}{17} \cdot 3 + \frac{2}{17} \cdot 3 + \frac{1}{17} \cdot 4 + \frac{1}{17} \cdot 5 + \frac{1}{17} \cdot 5 + \frac{1}{17} \cdot 4 + \frac{1}{17} \cdot 4 = \frac{55}{17} \approx 3,235 ,$$

natomiast idealna wartość to

$$H_{ideal} = - \left(2 \cdot \frac{3}{17} \cdot \log_2 \left(\frac{3}{17} \right) + 3 \cdot \frac{2}{17} \cdot \log_2 \left(\frac{2}{17} \right) + 5 \cdot \frac{1}{17} \cdot \log_2 \left(\frac{1}{17} \right) \right) \approx 3,175 .$$

Kodowanie Huffman'a daje prawie idealnie krótki ciąg kodowy ponieważ $\frac{3,235}{3,175} \approx 1,019$ czyli zaledwie 1,9% więcej, niż najkrótszy teoretycznie możliwy kod.