

# **WOJSKOWA AKADEMIA TECHNICZNA**

**im. Jarosława Dąbrowskiego**

## **Projekt**

**Laboratorium Kompresja Danych  
Multimedialnych**

2019/2020

***Zastosowanie kodowania LPC jako metody  
kompresji sygnału mowy***

***Prowadzący ćwiczenia:***

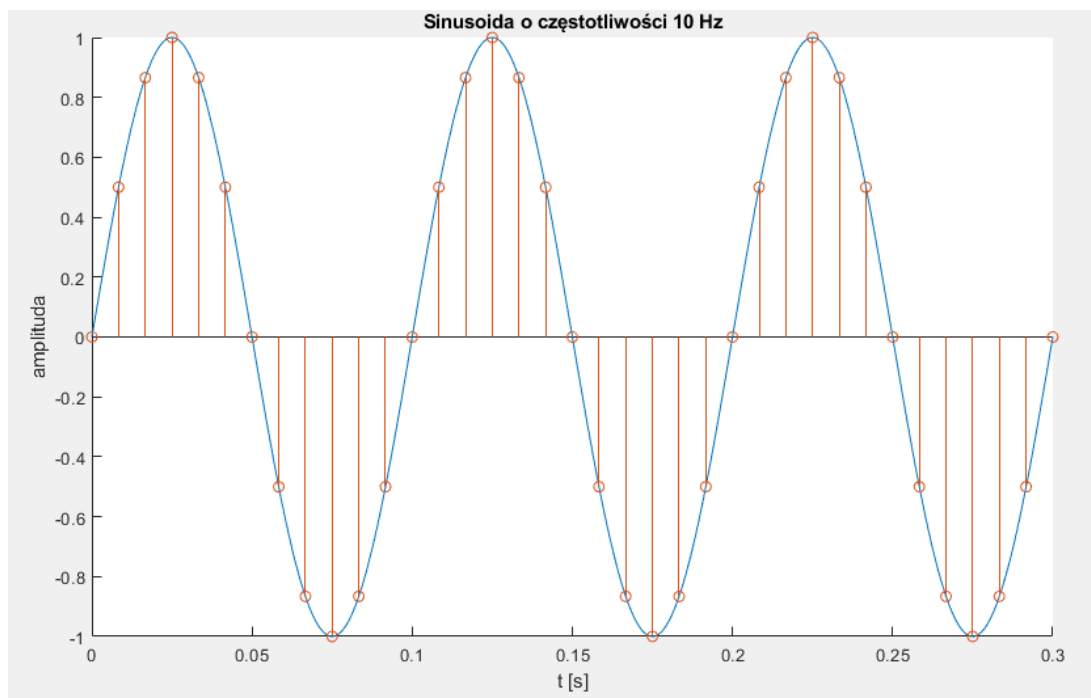
***dr inż. Leszek Grad***

***Paweł Kierski I7M1S1***

## Wstęp teoretyczny

Jak podaje definicja, dźwięk to wrażenie słuchowe, spowodowane falą akustyczną rozchodzącą się w ośrodku sprężystym. Częstotliwości fal, które są słyszalne dla człowieka, zawarte są w paśmie między wartościami granicznymi od ok. 16 Hz do ok. 20 kHz. Ponieważ dźwięk jest zjawiskiem wywodzącym się z natury to sygnał, reprezentujący falę dźwiękową jest analogowy. Oznacza to, że w dowolnym momencie  $t_0$  jesteśmy w stanie odczytać jego wartość. Sygnał w tej formie jest niemożliwy do zapisania na żadnym dostępnym obecnie nośniku, ponieważ okres próbkowania takiego sygnału dąży do 0. Dlatego w celu zapisania fali dźwiękowej sygnał analogowy musi zostać przekonwertowany na sygnał cyfrowy. Metoda cyfrowego zapisu polega na próbkowaniu sygnału w określonych odstępach czasu  $T_0$ , gdzie  $\frac{1}{T_0}$  – częstotliwość próbkowania. W celu zobrazowania procesu próbkowania umieszczono przykład funkcji sinusoidalnej.

$$y = \sin(2\pi f t)$$
$$f_0 = 10 \text{ Hz}$$

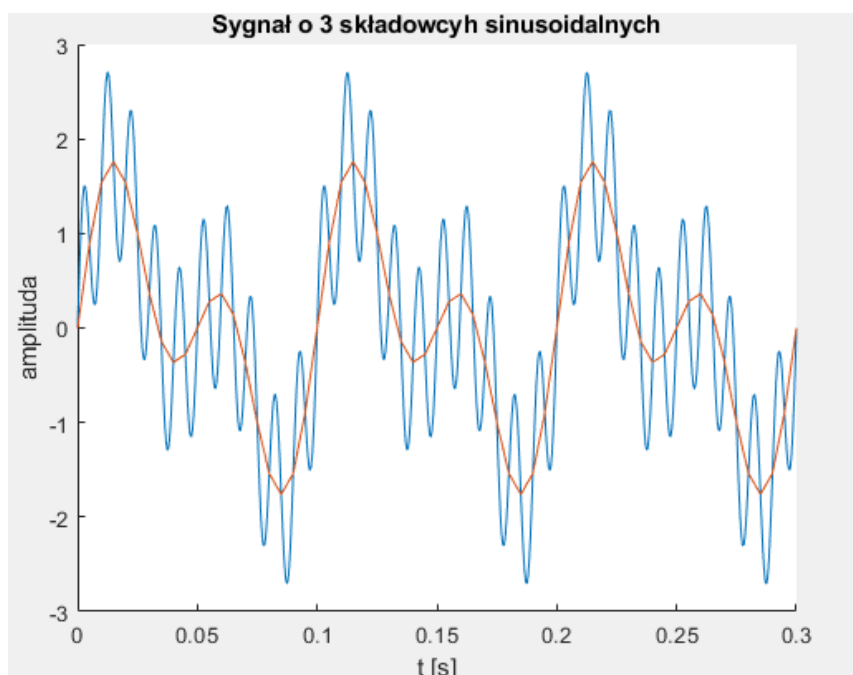


Częstotliwość próbkowania dla tego przypadku wynosi  $12f_0$ . Zapis chwilowych wartości dla tak próbkowanego sygnału jest o wiele łatwiejszy i pozwala na późniejsze odtworzenie faktycznego przebiegu sinusoidy.

Ponieważ sygnał dźwiękowy to suma składowych sinusoidalnych możemy sprowadzić funkcję opisującą ten sygnał do postaci sumy:

$$Y = \sum_{i=0}^n \sin(2\pi f_i t + \varphi)$$

$f_i$  to częstotliwość pojedynczej składowej fali dźwiękowej. Dobierając częstotliwość próbkowania musimy zatem liczyć się z tym iż składowe o częstotliwościach wyższych zostaną pominięte, a sygnał ciągły utworzony z próbek będzie „wygładzony”, czego efekt przedstawiony został poniżej.



Efekt taki jest niepożądany w sytuacji, gdy zależy nam na dokładnym odwzorowaniu. By uniknąć takiego wyniku częstotliwość próbkowania powinna zostać zwiększona, co oznacza więcej próbek, a co za tym idzie potrzeba więcej miejsca na zapis. W przechowywaniu danych, zależy nam zarówno na jak najlepszym (w zależności od potrzeb) odwzorowaniu rzeczywistości oraz jak najmniejszym wolumenie takiego zapisu. Dzięki procesowi jakim jest kompresja, możemy osiągnąć większą dokładność zapisu, przy zachowaniu lub nawet zmniejszeniu miejsca wymaganego na przechowanie danych. Dzięki kompresji każdy z nas może za pośrednictwem telefonu komórkowego usłyszeć znajomy głos z drugiego końca globu, czy odsłuchać ulubioną piosenkę w zatłoczonym autobusie miejskim.

## LPC – metoda kompresji

Jedną z metod wykorzystywaną do kompresji dźwięku jest metoda LPC (Linear predictive coding) . To potężne narzędzie w przetwarzaniu sygnału dźwiękowego, które opiera się na predykcji liniowej. LPC jest obecnie najczęściej stosowaną metodą w syntezie oraz kodowaniu mowy. LPC analizuje sygnał mowy poprzez oszacowanie formantów, usunięcie ich efektów z sygnału mowy oraz oszacowanie intensywności i częstotliwości pozostałych brzęczeń. Proces usuwania formantów nazywany jest filtracją odwrotną, a pozostały sygnał po odjęciu przefiltrowanego zamodelowanego sygnału nazywany jest pozostałością.

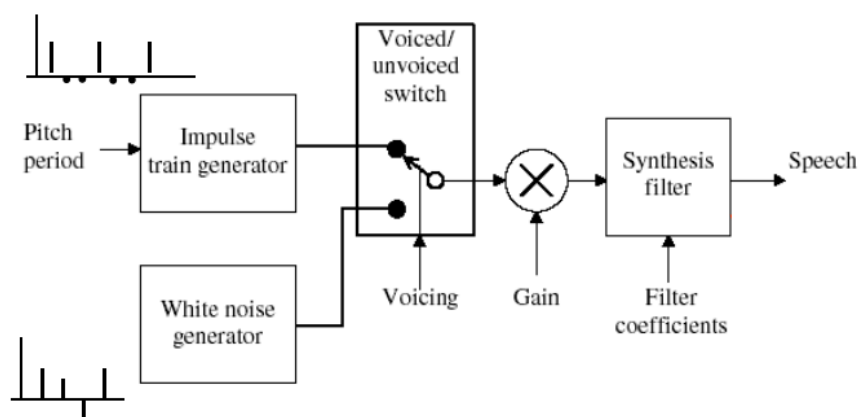
### Działanie filtra LPC

Predykcja liniowa polega na przewidywaniu kolejnej wartości bazując na poprzednich. Przy założeniu że  $B_i$  to jeden z trzech współczynników filtra LPC, a  $C_n$  to próbka sygnału z wektora próbek to;

$$C_n = B_{i-2}C_{n-3} + B_{i-1}C_{n-2} + B_iC_{n-1}$$

### Przykładowy model LPC

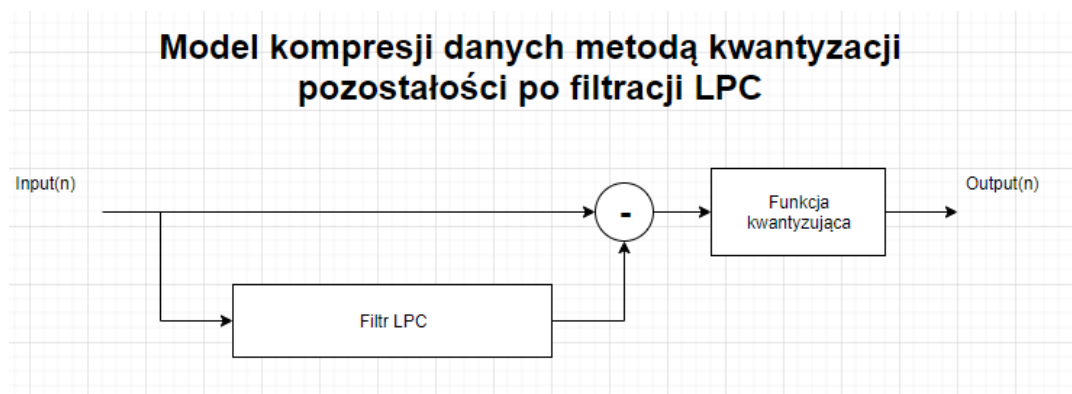
Poniżej przedstawiono przykład uproszczonego modelu wykorzystującego LPC do syntezy mowy:



## Badany model LPC

### Model

Do celów badań wykorzystany został znacznie uproszczony model kompresji z wykorzystaniem filtra LPC, przedstawiony na poniższym diagramie.



Proces kompresji zaproponowaną metodą polega na uproszczeniu wartości próbek różnicy sygnału przed i po filtracji. Uzyskany efekt pozwoli na reprezentację każdej próbki na określonej liczbie bitów. Efektywność wybranej metody w zależności od skali kwantyzacji zostanie przedstawiona w dalszej części badań.

### Przykład kwantyzacji

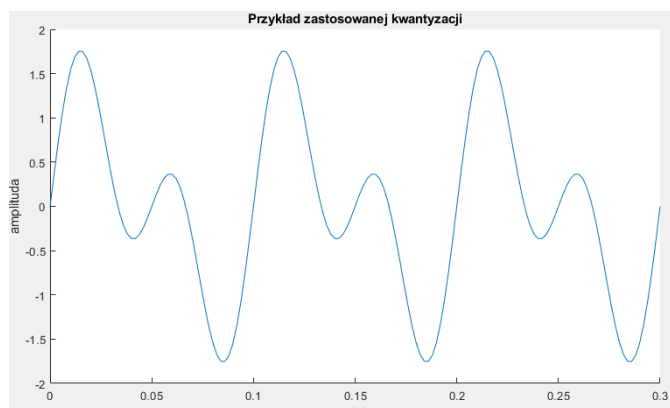
Dla wybranego wykresu sygnału należy policzyć wartość bezwzględną między maksymalnym wychyleniem górnym i dolnym. W tym przypadku:

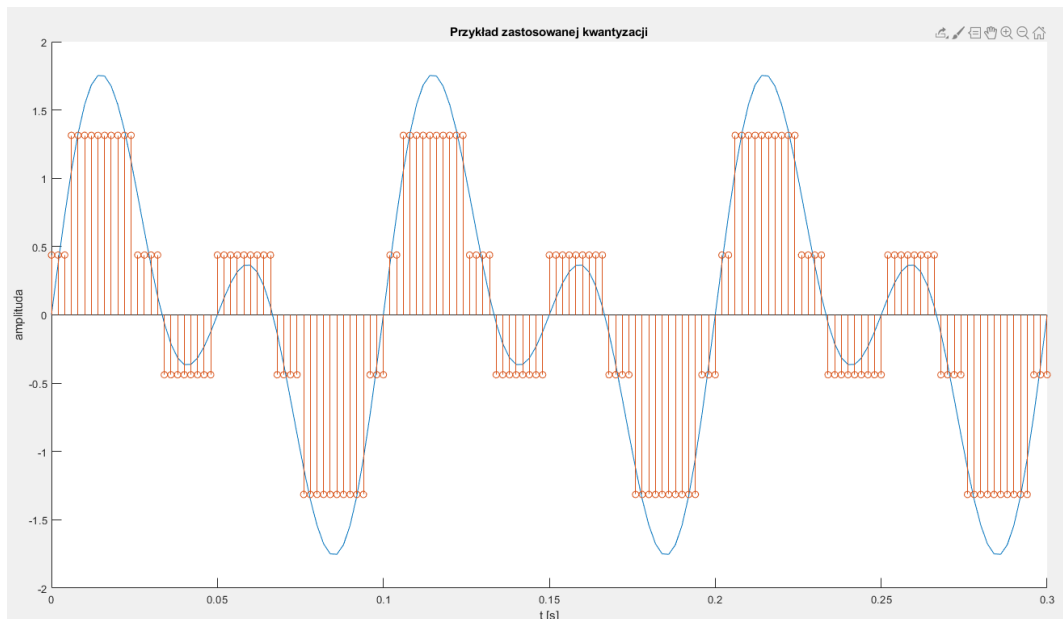
$$A = | \max - \min | = 4$$

By kwantyzacja była efektywna należy tak dobrać podziałkę, aby odległość pomiędzy punktem faktycznym, a skwantowanym był możliwie najmniejszy. Wartość podziałki określono ze wzoru, gdzie  $d$  - podziałka:

$$d = \frac{| \max - \min | - \frac{| \max - \min |}{\text{stopień kwantyzacji} * 2}}{\text{stopień kwantyzacji}} = 0,81$$

Na kolejnym wykresie przedstawione zostały skwantowane wartości sygnałów. Dla próbki rzeczywistej wyszukany został najbliższy poziom z wektora wartości kwantyzacji, po czym przypisano wartość.





## Listing skryptu dla prezentowanego modelu

Po przeanalizowaniu składowych postawionego problemu, można przystąpić do pisania skryptu w programie MatLab. Przygotowany skrypt zawiera prosty interfejs, który pozwala na wpisywanie głównych parametrów kompresji takich jak; nazwa pliku audio, liczba współczynników filtra LPC oraz liczba bitów na których ma zostać zapisany skompresowany sygnał.

```
% Kompresja Danych Multimedialnych 19/20
% Zadanie projektowe
%
% Prowadzący: dr Leszek Grad
% Wykonał: Paweł Kierski I7M1S1

clear all;

%Pobranie od użytkownika skryptu potrzebnych informacji
disp('Kompresja LPC');
fileName = input('Podaj nazwe pliku z rozszerzeniem .wav: ', 's');
coefAmount = input('Podaj ilość współczynników filtra LPC: ');
sampleBits = input('Podaj ilość bitów na próbkę: ');

%Odczytanie zawartości pliku i przygotowanie zawartości pliku
[input, fs] = audioread(fileName);
input = input(:,1);
input = input(2600:11150);
input = input';

%Wyznaczenie współczynników liniowych
N = length(input);
f = 0 : fs/N : fs - fs/N;
t = 0 : 1/fs : 1/fs * N - 1/fs;

%Przygotowanie filtra LPC
lpcSample = randn(100, 1);
lpcFilter = lpc(lpcSample, coefAmount);
```

```

%Wykonanie filtracji na sygnale wejsciowym i obliczenie roznicy
filteredInput = filter(lpcFilter, 1, input);
signalDiff = input - filteredInput;

%obliczenie wzmacnienia
force = max(input)/max(signalDiff);

%Kwantyzacja wartosci pomiedzy szczytami roznicy
quantum = abs(max(signalDiff) - min(signalDiff)) / 2.^sampleBits;
quantumVector = min(signalDiff) + quantum / 2 : quantum : max(signalDiff) - quantum / 2;

compSignal = zeros( [1 length(signalDiff)]); %Alokacja pamieci

%petla okreslajaca do ktorej wartosci zostanie "przydzielona probka"
for i = 1 : length(signalDiff)
    [val, idx] = min(abs(quantumVector - signalDiff(i)));
    compSignal(i) = idx-1;
end

%alokacja pamieci
uncompSignal = compSignal;

%dekompresja; przydzielenie wartosci dla konkretnego wspolczynnika
for i = 1 : length(uncompSignal)
    uncompSignal(i) = quantumVector(compSignal(i) + 1);
end

%Defiltracja sygnalu zdekompresowanego
outputSignal = filter(lpcFilter, 1, uncompSignal);

%Analiza 2
figure(2);
subplot(221);
title('Wykres slupkowy skwantowanego sygnalu');
stem(uncompSignal(3800:4500), 'MarkerSize', 0.001);
subplot(222);
stem(compSignal(300 : 600), 'MarkerSize', 0.001);
title('Wykres slupkowy indexow kwantyzacji');
subplot(223);
plot(outputSignal(3800:4500));
title('Frament sygnalu po dekompresji');
subplot(224);
plot(signalDiff(3800:4500));
title('Frament sygnalu przed kompresja');

%Wzmocnienie sygnalu po dekompresji
outputs = outputSignal * force;

%Analiza 2
figure(1);
subplot(221)
plot(t, input)
xlabel('Czas t[s]');
ylabel('Wartosc sygnalu');
title('Wykres przebiegu czasowego sygnalu po dekompresji');
subplot(222)
plot(f, abs(fft(input)));
xlabel('Czestotliwosc f[Hz]');
ylabel('Amplituda');
title('Wykres widma aplitudowego sygnalu po dekompresji');
subplot(223);
plot(t, outputs);
xlabel('Czas t[s]');
ylabel('Wartosc sygnalu');

```

```

title('Wykres przebiegu czasowego sygnału po dekompresji');
subplot(224)
plot(f ,abs(fft(outputS)));
xlabel('Częstotliwość f[Hz]');
ylabel('Amplituda');
title('Wykres widma amplitudowego sygnału po dekompresji');

```

## Opis działania skryptu

Skrypt do rozpoczęcia pracy potrzebuje trzech parametrów wejściowych; ścieżkę do pliku audio w formacie .wav, liczbę współczynników filtra LPC, ilość bitów (stopień kwantyzacji). Po uruchomieniu wyświetla dwa okna, każdy z czterema wykresami w celu dokonania analizy oraz wyciągnięcia wniosków. Poniżej przedstawiono przykładowe uruchomienie skryptu.

```

Nazwa pliku z rozszerzeniem .wav: audio.wav
Ilość współczynników filtra LPC: 10
Ilość bitów na próbkę: 2

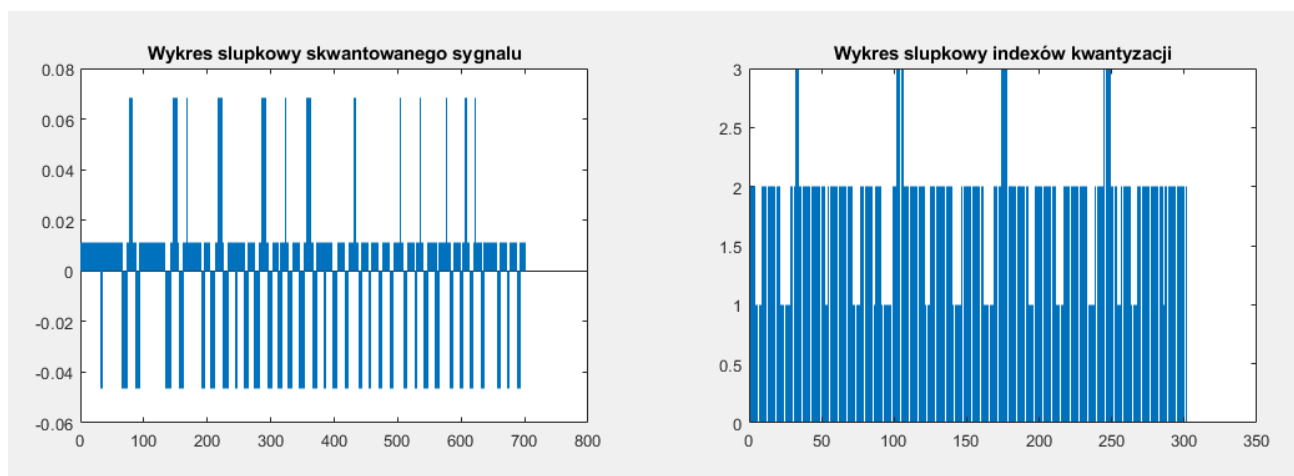
```

## Szczegółowa analiza modelu

W celu dokonania analizy zaproponowanego modelu, koniecznym będzie kilkakrotne uruchomienie wykonanego skryptu dla różnych parametrów wejściowych. Dzięki takiemu działaniu, skutecznie można manipulować efektami kompresji.

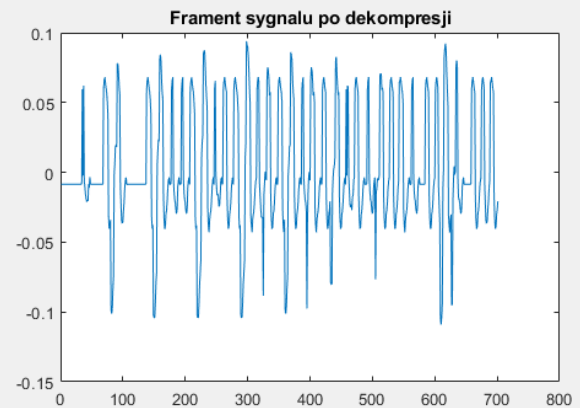
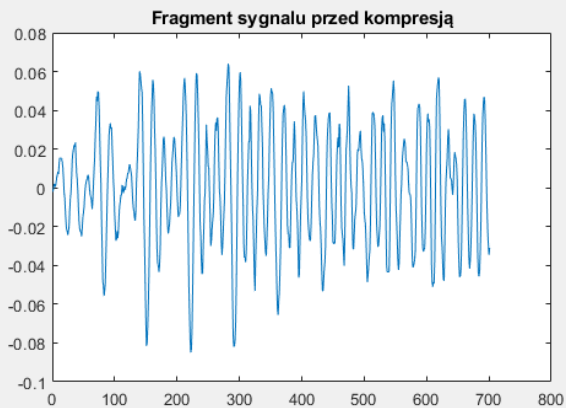
Badanie wpływu ilości bitów na skompresowaną próbkę sygnału:

1. Audio: `audio.wav`; Ilość bitów na dane: 2; Liczba współczynników LPC: 11;

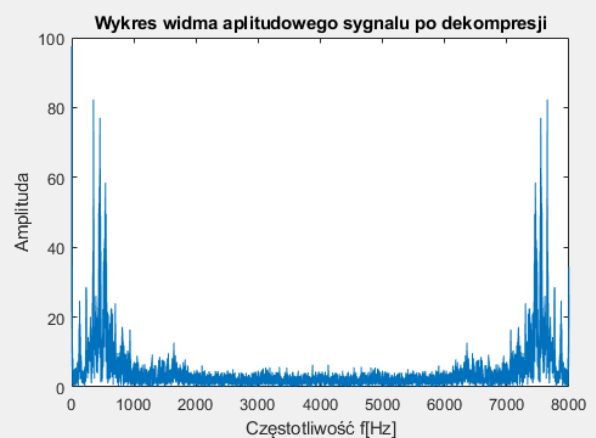
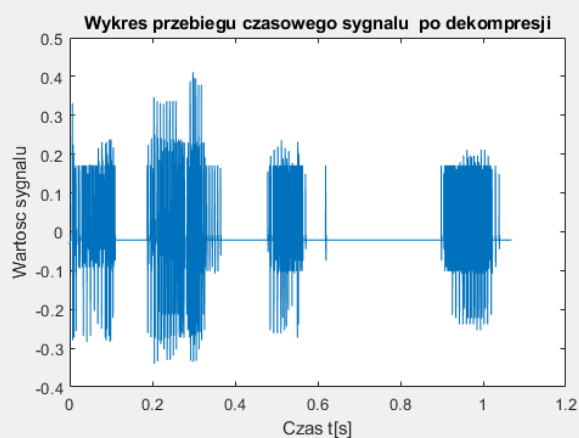
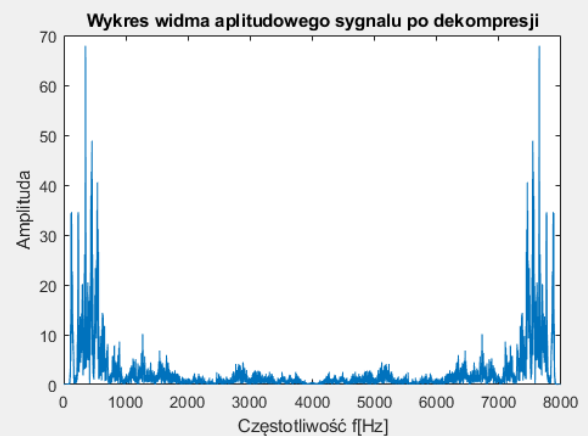
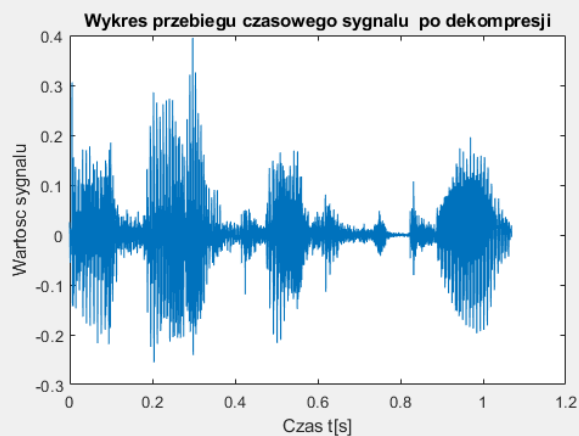




Stopień kwantyzacji dla 2 bitów jest bardzo duży, ponieważ rzeczywisty przebieg należy sprowadzić do 4 poziomów.



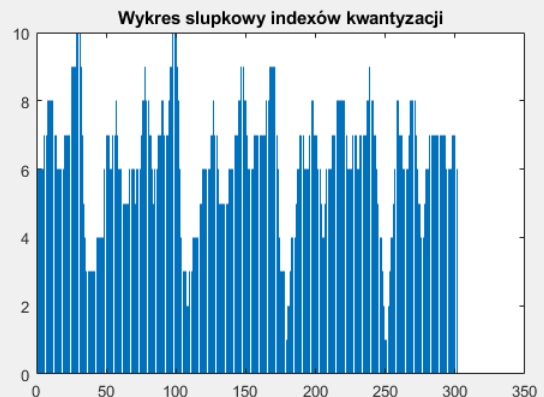
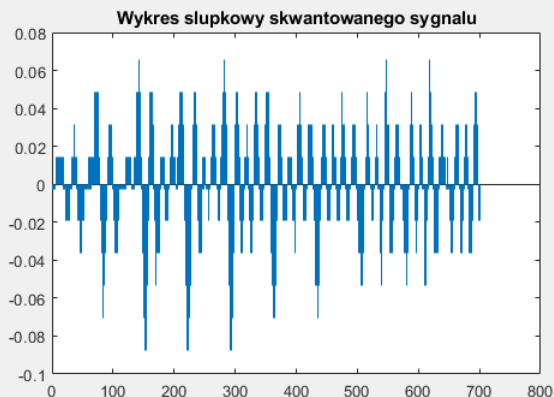
Jak można zauważyć na załączonych wykresach, sygnał po dekompresji został odtworzony w znacznym przybliżeniu. Obrys przebiegu czasowego został w pewien sposób zachowany. Dla zastosowanej metody o takich parametrach nie powinno spodziewać się czystego dźwięku.



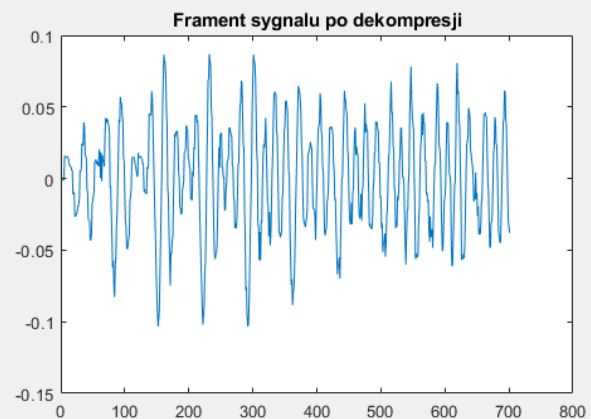
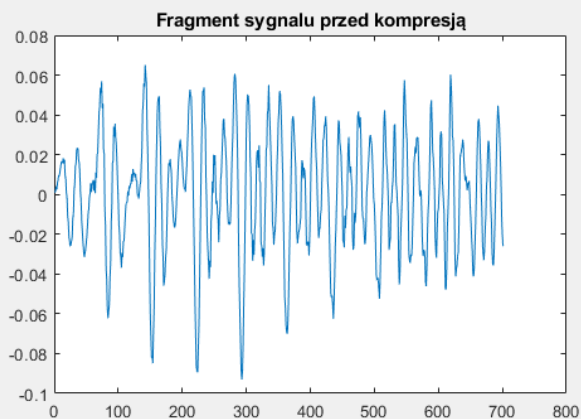
Na załączonych wykresach można dostrzec znaczną różnicę pomiędzy sygnałem

wejściowym, a wyjściowym po dekompresji. Walory słuchowe oceniam na słabe. O ile można usłyszeć treść przekazywanej informacji za pomocą mowy użytej w pliku audio, tak dźwięk wydaje się być „postrzępiony”, występuje też duża ilość szumu. Stopień kompresji jaki uzyskano w wyniku zabiegu to około 87%.

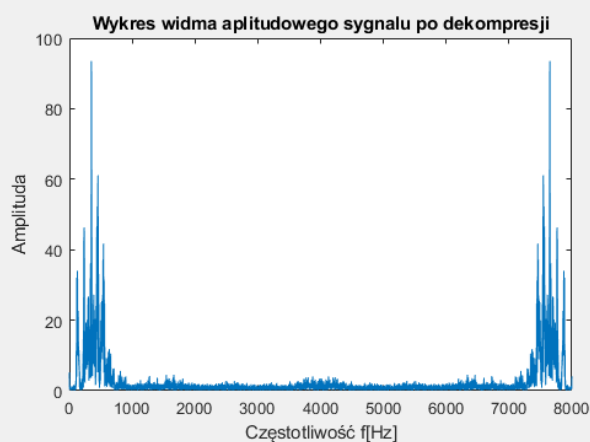
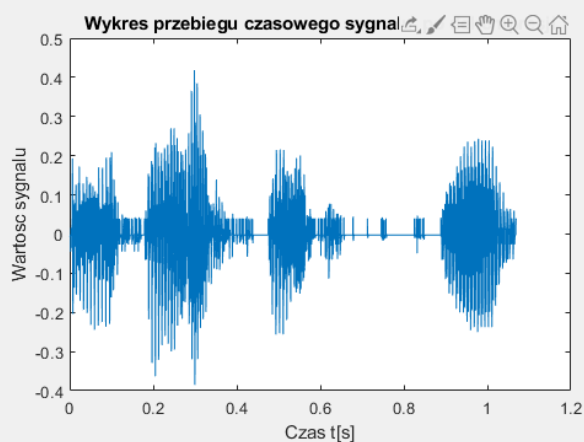
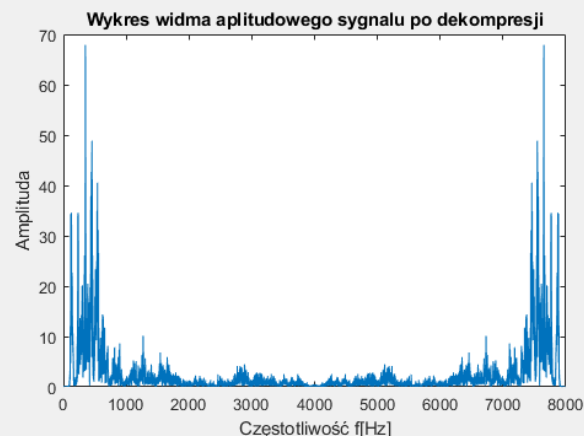
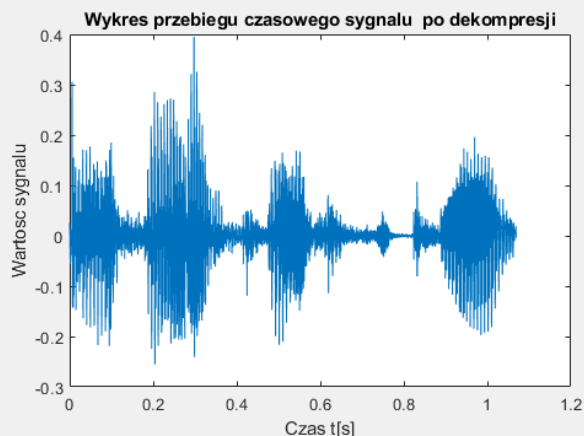
2. Audio: `audio.wav`; Liczba bitów na dane: 4; Liczba współczynników LPC: 11;



Dla 4 bitów jesteśmy w stanie zapisać już 16 różnych poziomów próbek, dlatego spodziewamy się znacznie lepszego wyniku w stosunku do próby poprzedniej. Maksymalna wartość na wykresie słupkowym indexów osiąga maksymalną wartość 10, ponieważ wyższe wartości zostały zarezerwowane dla większych pików sygnału, które dla przedstawionego fragmentu sygnału nie występują.

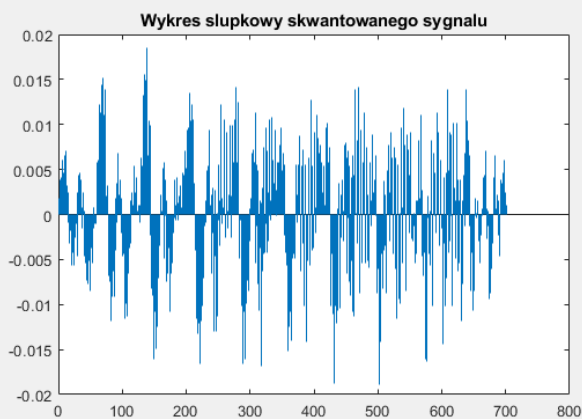


Przebieg czasowy sygnału przed i po kompresji jest dużo lepiej odwzorowany niż w przypadku zapisu na 2 bitach.

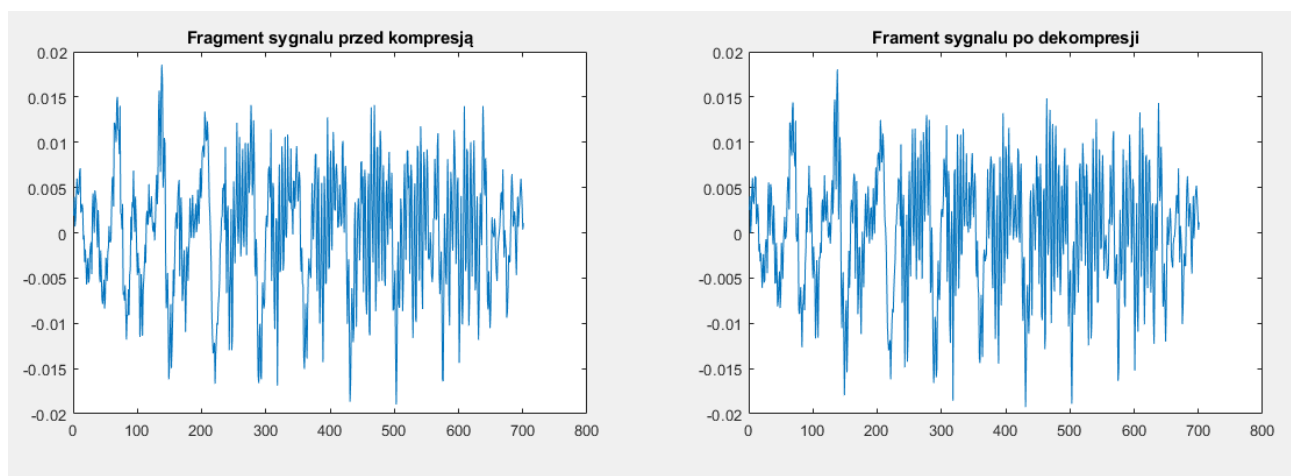


Sygnał przed i po kompresji zachowuje podobny kształt. Również walory słuchowe są o wiele lepsze niż w przypadku zapisu na 2 bitach. Pomimo obecności drobnych szumów można wyraźnie usłyszeć ton oraz barwę głosu z nagrania. W mojej opinii do potrzeb przekazania informacji kompresja taka jest zadowalająca. Stopień kompresji to 75%.

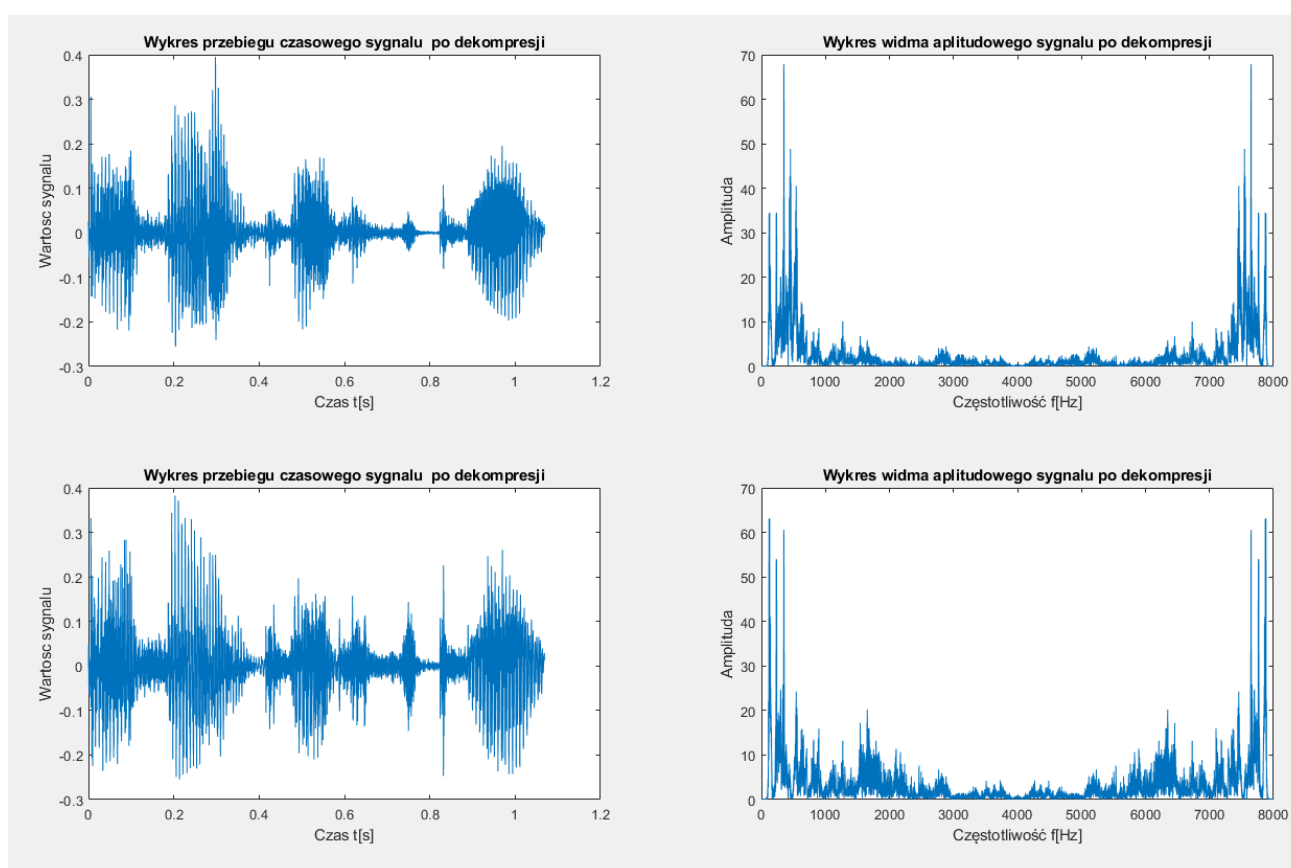
3. Audio: `audio.wav`; Liczba bitów na dane: 8; Liczba współczynników LPC: 11;



Dla zapisu na 8 bitach liczba możliwych poziomów sygnału do zapisania wynosi 256.



Fragment sygnału wejściowego i tego uzyskanego w wyniku dekompresji jest podobny na tyle, że w tej skali różnica jest niezauważalna.



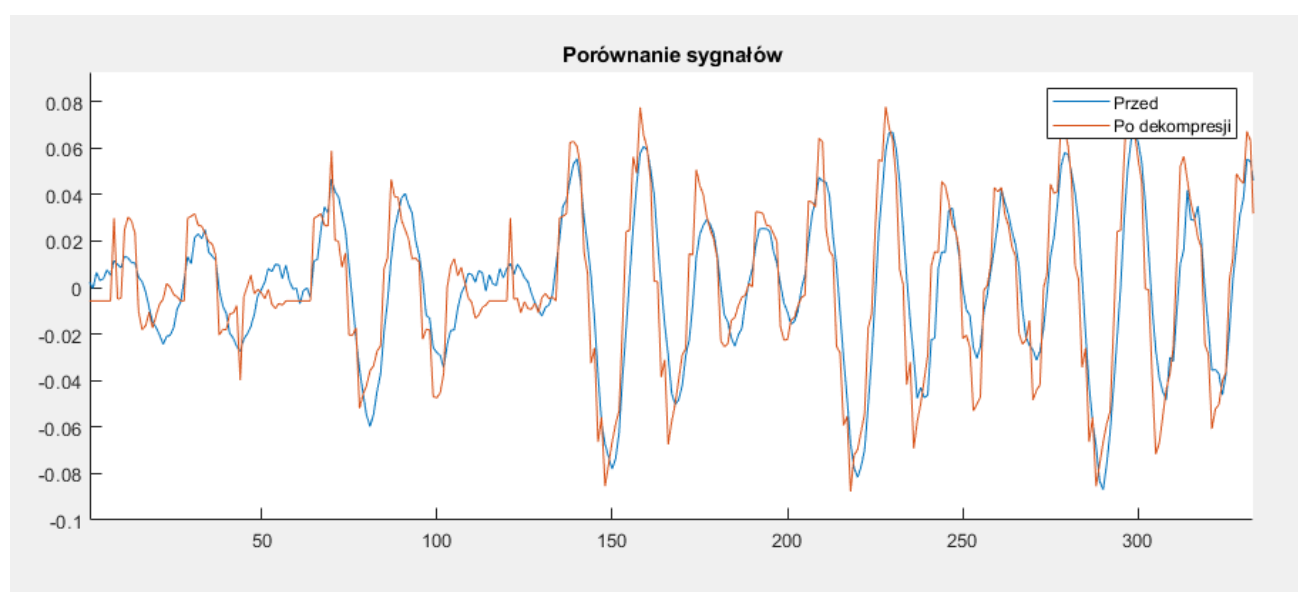
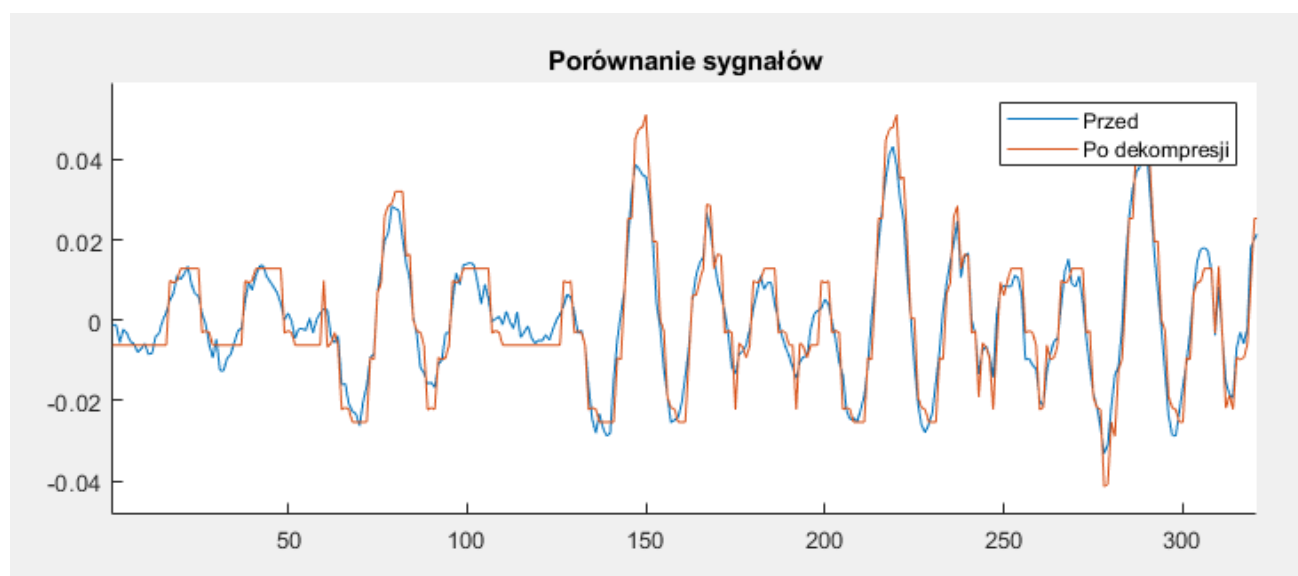
Wyraźna różnica jest zauważalna dopiero na pełnym przebiegu czasowym lub widmie sygnału. Różnica po odsłuchu jest zauważalna, natomiast odwzorowanie jest na bardzo wysokim poziomie. Występuje mało szumów, dźwięk jest czysty. Nastąpiła 50% kompresja.

## Podsumowanie wyników dla różnych ilości bitów zapisu skompresowanego sygnału

Manipulacja liczbą bitów zapisu jest bezpośrednim sposobem na regulowanie stopnia kompresji. Informacja zapisana na 16 bitach zajmuje 8 krotnie więcej miejsca niż informacja zapisana na 2 bitach. Parametr ten należy dostosować w zależności od preferencji co do jakości dźwięku po dekompresji.

### Dobór ilości współczynników LPC

W celu przeprowadzenia badania uruchomiono program dwa razy; dla stałej wartości ilości bitów na próbkę; 3 oraz dla liczby współczynników LPC 3 oraz 12.



Na załączonych wykresach można zauważyć, że dla mniejszej ilości współczynników występują tzw. „sygnały stałe”, które są bezdźwięczne na odsłuchu. Dla większej ilości współczynników LPC zjawisko to występuje rzadziej. Dla sygnału mowy zadowalająca liczba współczynników LPC to 11-13.

## Wnioski

Zaprezentowany model kompresji działa zgodnie z założeniami. Uzyskana kompresja na poziomie 91 % dla dwóch bitów na próbkę przy jednoczesnym zachowaniu przekazywanej za pomocą mowy informacji jest zadowalająca i niemożliwa do uzyskania np. poprzez przeprowadzenie decymacji czy filtracji. Z powyższych badań wynika, że LPC jest potężnym narzędziem, które w połączeniu z innymi dostępnymi metodami obróbki dźwięku umożliwia uzyskanie jeszcze lepszych efektów kompresji. Np. w telefonii komórkowej dochodzi ona do 95% przy jednoczesnym zachowaniu dobrej jakości odtwarzanego dźwięku.