

Moduł 3

KOMPRESJA SYGNAŁÓW

PRZEMYSŁAW DYMARSKI

KWANTYZACJA, KWANTYZACJA WEKTOROWA, PREDYKCJA, PCM, ADPCM, CELP, KODOWANIE W DZIEDZINIE CZĘSTOTLIWOŚCI, KOMPRESJA OBRAZU, JPEG

W MODULE 3 PRZEDSTAWIONO ZAGADNIENIA KOMPRESJI MOWY, MUZYKI I OBRAZU.

OMÓWIONO METODY KODOWANIA ŹRÓDŁOWEGO I KOMPRESJI: KWANTYZACJA SKALARNA, WEKTOROWE, KODOWANIE PREDYKCYJNE I KODERY TRANSFORMATY. Z MODULEM 3 SĄ ZWIĄZANE DWIE SERIE EKSPERYMENTÓW ILUSTRUJĄCE WYKŁAD (KWANTYZACJA WEKTOROWA I KODOWANIE TRANSFORMARY SYGNAŁU AUDIO) ORAZ DWA ĆWICZENIA LABORATORYJNE (KWANTYZACJA SKALARNA I KOMPRESJA SYGNAŁU MOWY). SĄ ONE OPISANE W MODULE 4 .

Spis treści

1	Kompresja stratna i bezstratna	2
1.1	Kompresja bezstratna i jej granice	2
1.2	Kompresja stratna i jej granice	4
2	Kwantyzacja skalarna i kodery PCM	5
2.1	Przetworzenie analogowo-cyfrowe	5
2.2	Stosunek mocy sygnału do mocy błędu kwantyzacji.....	6
2.3	Kwantyzator równomierny	7
2.4	Kwantyzator nierównomierny i standard PCM G.711	9
2.5	Kwantyzator adaptacyjny	12
3	Predykcja i kodery ADPCM	14
3.1	Działanie kodera ADPCM	14
3.2	Obliczanie predyktora.....	15
3.3	Metody sekwencyjne (materiał dodatkowy).....	16
3.4	Koder ADPCM G.726	17
3.5	Interpretacja predykcji w dziedzinie częstotliwości	19
4	Kwantyzacja wektorowa.....	21
5	Kodery predykcyjno – wektorowe CELP i wokodery predykcyjne	25
6	Kodowanie w dziedzinie częstotliwości.....	32
7	Kodowanie obrazu nieruchomego i sekwencji wideo	37
7.1	Reprezentacje obrazu nieruchomego i ruchomego	37
7.2	Kwantyzacja skalarna i wektorowa obrazu.....	37
7.3	Transformaty dwuwymiarowe	39
7.4	Kompresja obrazu w dziedzinie transformaty	40
7.5	Kompresja sekwencji wideo	42
8	Zadania i testy.....	43
8.1	Przykładowe zadania z rozwiązaniami.....	43
8.2	Zadania do rozwiązania	46
9	Biblioteka.....	47
10	Bibliografia.....	48

1 Kompresja stratna i bezstratna

1.1 Kompresja bezstratna i jej granice

Transmisja cyfrowa wymaga zapisu przesyłanej informacji w postaci strumienia binarnego. Strumień binarny to ciąg wartości logicznych „0” i „1”. Celowo unikam tu słowa „bit”, gdyż nie zawsze liczba przesyłanych wartości logicznych jest równa liczbie **bitów informacji**. Zetknęliśmy się już z tym problemem w p.3.3 Moduł2. Transmitując serię sygnałów binarnych pojawiających się z niejednakowym prawdopodobieństwem (sygnał $s_1(t)$ reprezentował wartość logiczną „1” i pojawiał się z prawdopodobieństwem P_1 , a $s_0(t)$ reprezentował wartość logiczną „0” i pojawiał się z prawdopodobieństwem $P_0=1-P_1$) przekonaliśmy się, że przesyłamy mniej niż 1 bit informacji na sygnał. W istocie, gdy $P_1=1$, $P_0=0$ (lub odwrotnie) nie przesyłamy żadnej informacji, gdyż z góry wiemy, że otrzymamy zawsze ten sam sygnał. Gdy oba sygnały pojawiają się z prawdopodobieństwem $P_1=P_0=1/2$, wówczas dopiero przesyłamy jeden bit informacji na sygnał.

Jeśli ilość informacji zawarta w strumieniu binarnym jest mniejsza niż liczba wartości logicznych, wówczas istnieje możliwość przeprowadzenia **kompresji bezstratnej**, tzn. przetworzenia ciągu wartości logicznych na inny ciąg zawierający mniej elementów. Operacja taka jest bezstratna, tzn. możemy zawsze wrócić do pierwotnego ciągu wartości logicznych.

Jaki jest cel stosowania kompresji? Chodzi o oszczędne wykorzystanie kanału. Im mniej symboli transmisji danych transmitujemy w jednostce czasu, tym węższe pasmo częstotliwości zajmujemy i możemy podzielić się kanałem z innymi użytkownikami. Jeżeli informacja ma być zapisana w pamięci, to dzięki kompresji zużywamy mniej pamięci dla przechowania tych samych danych.

Nie wszystkie rodzaje danych mogą być poddane bezstratnej kompresji, jedynie takie, które dadzą się zapisać jako ciąg wartości logicznych. Dane tekstowe są najlepszym przykładem. Dźwięk, nawet spróbkowany – już nie, gdyż każda próbka jest liczbą rzeczywistą, więc nie może być przedstawiona jako ciąg binarny o skończonej długości. Tym niemniej mówi się o „bezstratnych” koderach dźwięku czy obrazu. Należy to rozumieć w ten sposób, że wstępnie dokonano przetworzenia analogowo-cyfrowego (A/C), otrzymując strumień binarny o dużej liczbie elementów. To przetworzenie A/C jest operacją stratną. Następnie przetwarzamy ten strumień binarny na inny strumień, mający mniejszą liczbę elementów. To już jest operacja bezstratna. W ten sposób postępujemy kodując obraz w systemie JPEG.

Algorytmy bezstratnej kompresji nie są w zasadzie algorytmami przetwarzania sygnałów, nie poświęcimy im więc wiele miejsca. Warto jednak poznać granicę stosowania tych algorytmów.

Założmy zatem, że źródło informacji może przekazać do transmisji M różnych wiadomości (np. liter alfabetu). Te wiadomości pojawiają się z prawdopodobieństwem odpowiednio p_1, p_2, \dots, p_M (np. litera „e” pojawia się częściej niż „q”). Jeśli pojawi się i -ta wiadomość, to ilość informacji, jaką otrzymamy, będzie odwrotnie proporcjonalna do p_i . Co więcej, będzie ona proporcjonalna do $\log_2\left(\frac{1}{p_i}\right)$. Dlaczego wykorzystano tu logarytm? Gdy otrzymamy dwie wiadomości (np. słowo złożone z dwóch liter alfabetu), to prawdopodobieństwo takiego zdarzenia jest równe iloczynowi prawdopodobieństw dla poszczególnych liter. Pozyskana w ten sposób ilość informacji jest sumą ilości informacji obu wiadomości. Przetworzenie iloczynu w sumę zapewnia funkcja logarymiczna: $\log\left(\frac{1}{p_1 p_2}\right) = \log\left(\frac{1}{p_1}\right) + \log\left(\frac{1}{p_2}\right)$. Podstawa logarytmu równa 2 daje wynik w bitach informacji.

Podstawowe znaczenie ma **średnia ilość informacji, zwana entropią**:

$$H = \sum_{i=1}^M p_i \log_2\left(\frac{1}{p_i}\right) \quad (1)$$

C. Shannon udowodnił, że kodując informacje dostarczane przez źródło o entropii H trzeba przeznaczyć (średnio) co najmniej H wartości logicznych („0” i „1”) na wiadomość. Dokładniej, **średnia długość słowa kodowego nie może być mniejsza niż entropia**: $L_{min} = H$. Przeznaczając mniej niż H wartości logicznych na

wiadomość, nie zdekodujemy poprawnie każdej z M możliwych wiadomości. Możemy jednak konstruować kody o średniej długości słowa L dowolnie bliskiej entropii:

$$H \leq L < H + \varepsilon, \quad \varepsilon > 0 \quad (2)$$

Jak stworzyć taki kod? Optymalny algorytm podał D.Huffman [3]. Zapoznamy się z nim na przykładzie: Mamy 8 zdarzeń, pojawiających się z różnym prawdopodobieństwem (Tabela 1).

Tabela 1 Kod Huffmana

Zdarzenie nr...	Prawdopodobieństwa 8 zdarzeń	Połączone prawdopodobieństwa		Słowa kodowe
1	0.3	0.6	1	11
2	0.3			10
3	0.1	0.2	0.4	011
4	0.1			010
5	0.05	0.1	0.2	0011
6	0.05			0010
7	0.05	0.1	0.2	0001
8	0.05			0000

Stosując zwykły kod binarny, mielibyśmy 8 słów o długości 3: 000, 001, 010,..., 111. Zastosujemy jednak kod o zmiennej długości słowa. Algorytm Huffmana składa się z kolejnych kroków i polega na łączeniu dwóch zdarzeń o najmniejszym prawdopodobieństwie.

W pierwszym kroku łączymy zdarzenia 7,8 i dodajemy ich prawdopodobieństwa (moglibyśmy zacząć od dowolnej pary wybranej z czterech ostatnich zdarzeń). W drugim kroku łączymy 5,6. Mamy teraz 6 zdarzeń, z których 4 mają najmniejsze prawdopodobieństwa równe 0.1. Łączymy dwa z nich, a potem dwa pozostałe. Mamy teraz 4 zdarzenia o prawdopodobieństwach 0.3, 0.3, 0.2, 0.2. Łączymy 2 ostatnie, potem 2 pierwsze. Ostatnim krokiem będzie połączenie dwóch zagregowanych zdarzeń i otrzymanie prawdopodobieństwa równego 1. Najczęściej zamiast tabeli tworzy się drzewo i właśnie dotarliśmy do jego korzenia.

Teraz wychodząc z korzenia docieramy do kolejnych "liści", czyli pierwotnych zdarzeń. Idąc w górę wpisujemy "1" do słowa kodowego, a idąc w dół wpisujemy "0" (można postąpić odwrotnie). Tak więc pierwszemu zdarzeniu przypisujemy kod 11, a ostatniemu 0000.

Średnią długość słowa kodowego otrzymamy, mnożąc długości słów przez prawdopodobieństwa ich pojawienia się. W naszym przypadku

$$L = 0.3 \times 2 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 3 + 4 \times (0.05 \times 4) = 2.6 \text{ bit} < 3 \text{ bit}$$

Otrzymaliśmy zatem bardziej efektywny kod niż zwykły kod binarny o długości słów równej 3.

Zauważmy, że kod Huffmana może być dekodowany „na bieżąco”, gdyż żadne słowo kodowe nie jest początkiem innego słowa. Taki kod nazywa się **kodem prefikсовym**.

Obliczając entropię analizowanego źródła informacji wg. wzoru (1), otrzymujemy 2.57 bita informacji. Kod Huffmana ma średnią długość słowa równą 2.6. Zgodnie ze wzorem (2) istnieje możliwość bardziej efektywnego kodowania. Możemy to osiągnąć, kodując pary następujących po sobie zdarzeń. Dla 8 zdarzeń takich par będzie 64.

Rozpatrzmy jednak prostszy przykład. Mamy 2 zdarzenia: E1 o prawdopodobieństwie $p_1 = 0.1$ i E2 o prawdopodobieństwie $p_2 = 0.9$. Entropia wynosi $H = 0.469$, ale kodując pojedyncze zdarzenia nie mamy wyboru: jednemu przypisujemy kod "1" a drugiemu "0". Długość kodu wynosi $L = 1$.

Kodując 4 pary zdarzeń i stosując algorytm Huffmana mamy:

E1, E1 - prawdopodobieństwo	0.01	kod	111
E1, E2 - prawdopodobieństwo	0.09	kod	110
E2, E1 - prawdopodobieństwo	0.09	kod	10
E2, E2 - prawdopodobieństwo	0.81	kod	0

Średnia długość słowa wynosi $0.01 \times 3 + 0.09 \times 3 + 0.09 \times 2 + 0.81 \times 1 = 1.29$ na parę zdarzeń, a więc 0.645 wartości logicznych na zdarzenie.

Kodując trójki sąsiednich zdarzeń, możemy jeszcze bardziej zbliżyć się do entropii. Takie postępowanie nazywa się **rozszerzaniem źródła informacji**.

Istnieje wiele efektywnych algorytmów kompresji bezstratnej, zainteresowanych odsyłam do [3].

1.2 Kompresja stratna i jej granice

Występujące w naszym otoczeniu dźwięki mowy, muzyki czy obrazy są źródłami informacji o nieskończonej entropii, nie można więc ich bezstratnie zakodować w postaci strumienia binarnego o skończonej długości. Próbkki sygnału akustycznego czy piksele obrazu są to liczby rzeczywiste, mogące przyjmować nieskończenie wiele różnych wartości. Nie można ich zapisać w słowie kodowym o skończonej długości.

Nieskończona entropia wynika z następującego rozumowania. Niech próbka sygnału lub piksel obrazu przyjmuje M wartości z jednakowym prawdopodobieństwem $p_i = \frac{1}{M}$. Entropia obliczona wg wzoru (1) wynosi:

$$H = \sum_{i=1}^M p_i \log_2\left(\frac{1}{p_i}\right) = \sum_{i=1}^M \frac{1}{M} \log_2(M) = M \frac{1}{M} \log_2(M) = \log_2(M)$$
. Gdy M dąży do nieskończoności, wówczas i entropia staje się nieograniczona.

Gdy ograniczymy liczbę wartości logicznych reprezentujących sygnał o nieskończonej entropii, powstanie różnica między sygnałem oryginalnym i sygnałem zdekodowanym ze strumienia binarnego:

$$e = x - x^* \tag{3}$$

W przypadku kodowania sygnałów akustycznych ta różnica jest postrzegana jako szum, mówimy o szumie kwantyzacji. Miarą jakości odtworzonego sygnału akustycznego jest stosunek mocy sygnału do mocy szumu:

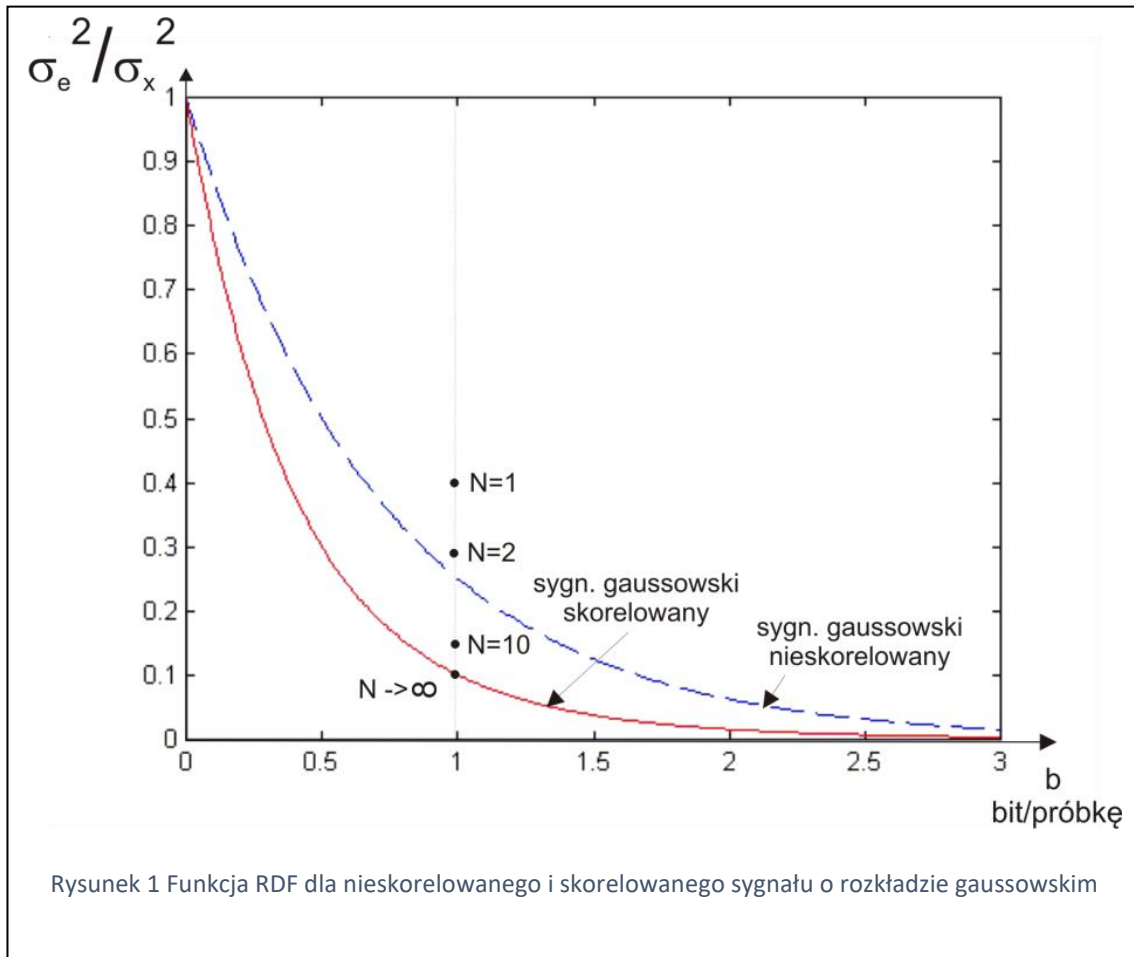
$$SNR_q = \frac{\sigma_x^2}{\sigma_e^2} \tag{4}$$

Konstruując kodery, dążymy do maksimum SNR_q , przy danej średniej długości słowa kodowego b elementów binarnych (wartości logicznych) na próbkę sygnału czy piksel obrazu (ogólnie mówimy „na odczyt źródła informacji”). Starając się zoptymalizować koder dźwięku czy obrazu, powinniśmy pamiętać o tym, że wartość SNR_q jest ograniczona właściwościami samego sygnału i istnieje nieprzekraczalna wartość maksymalna SNR_q dla każdej wartości b . To ograniczenie zazwyczaj przedstawia się w postaci **funkcji RDF (funkcja przepływność – zniekształcenie, ang. rate - distortion function)**. Na osi x odkładamy wartości b (długość słowa kodowego na próbkę lub piksel, w przypadku dźwięku staje się przepływnością binarną po pomnożeniu przez częstotliwość próbkowania). Na osi y odkłada się odwrotność SNR_q , czyli stosunek mocy zniekształceń powstających w procesie stratnej kompresji do mocy sygnału. Przykładowy przebieg funkcji RDF dla sygnału, którego wartości chwilowe mają rozkład gaussowski, pokazano na rys.1. Jeżeli próbki sygnału nie są skorelowane, wówczas RDF jest opisana wzorem

$$\frac{\sigma_e^2}{\sigma_x^2} = 2^{-2b} \quad (5)$$

Jeśli próbki sygnału są skorelowane, to oznacza że istnieje możliwość zmniejszenia zniekształceń (szumu) dzięki stosowaniu predykcji. Oznacza to obniżenie wartości RDF dla tego sygnału. RDF jest funkcja malejącą, ale wartość zerową (czyli bezstratne kodowanie) osiąga tylko dla sygnałów o skończonej entropii. Wartość b , przy której RDF osiąga wartość zerową, jest właśnie entropią sygnału.

Wszystkie kodery osiągają wartości $\frac{\sigma_e^2}{\sigma_x^2}$ powyżej entropii. Na rys.1 pokazano wyniki dla kwantyzatora wektorowego kodującego N-wymiarowe wektory skorelowanych próbek o rozkładzie gaussowskim.



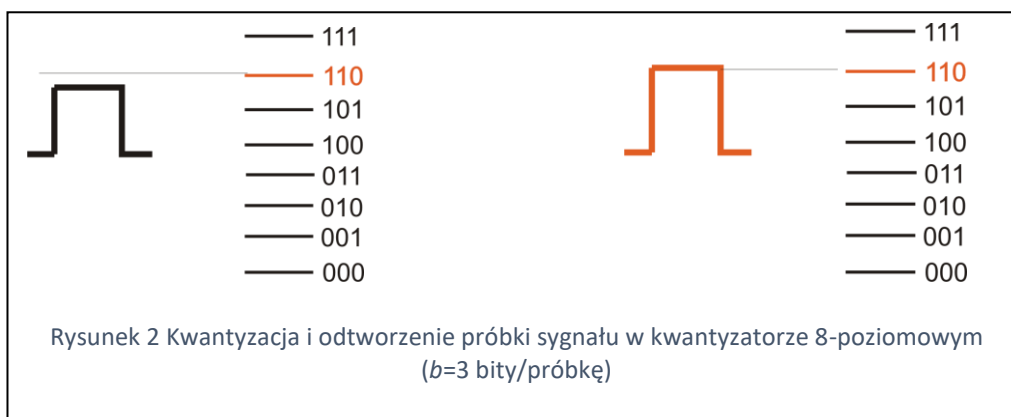
2 Kwantyzacja skalarna i kodery PCM

2.1 Przetworzenie analogowo-cyfrowe

Aby przesłać sygnał analogowy kanałem cyfrowym, należy zapisać go w strumieniu binarnym, jako ciąg wartości logicznych „0” i „1”. Potocznie nazywa się je bitami, choć niekoniecznie są to bity informacji - była o tym mowa w poprzednim punkcie. My też będziemy używać tego potocznego określenia. **Przetworzenie analogowo-cyfrowe składa się z dwóch procesów: próbkowania i kwantyzacji.** O próbkowaniu była mowa w p.4 Modułu 1. Próbkowanie sygnałów o ograniczonym paśmie częstotliwości B [Hz] może być operacją bezstratną, jeśli postępujemy zgodnie z twierdzeniem o próbkowaniu, pobierając nie mniej niż $2B$ próbek na sekundę.

Kwantyzacja jest obciążona błędem, gdyż próbki sygnału analogowego są liczbami rzeczywistymi, a mając b bitów na zakodowanie każdej próbki, jesteśmy w stanie rozróżnić jedynie $L = 2^b$ wartości. W tym celu

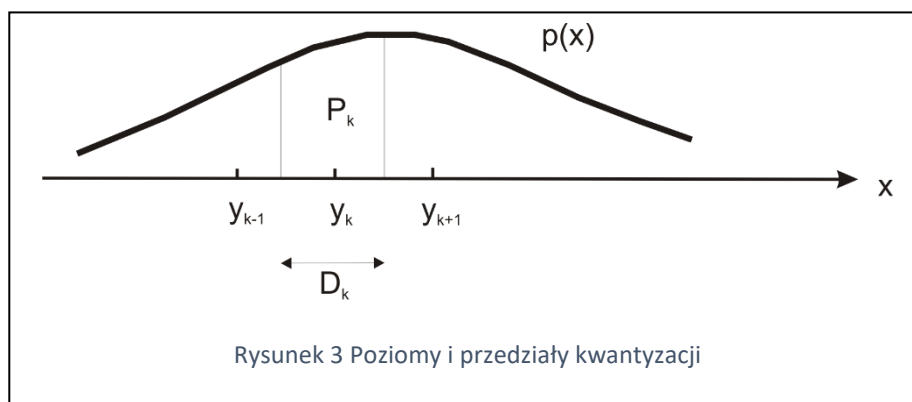
wyberamy L wartości zwanych poziomami kwantyzacji i zaokrąglamy wartość próbki do najbliższego poziomu. Proces kwantowania i odtworzenia próbki z wykorzystaniem kwantyzatora 8-poziomowego pokazano na rys.2.



Każdemu poziomowi kwantyzacji jest przypisane słowo kodowe, w omawianym przykładzie 3-bitowe. Ze względu na fakt, że niektóre poziomy są wybierane częściej, a inne rzadziej, można tu wykorzystać kod o zmiennej długości słowa (patrz punkt 1). W ten sposób łączymy kwantyzację z kompresją sygnału. Trzeba jednak pamiętać, że nie ma sposobu, aby moc sygnału błędu spadła poniżej wartości funkcji RDF (rys.1).

2.2 Stosunek mocy sygnału do mocy błędu kwantyzacji

Po próbkowaniu mamy M próbek sygnału: x_1, x_2, \dots, x_M . Ich wartości chwilowe są opisane gęstością prawdopodobieństwa $p(x)$ – rys.3.



Kwantyzator jest opisany L poziomami kwantyzacji, nazwijmy je y_1, y_2, \dots, y_L . Kwantowanie próbki x_n polega na jej zaokrągleniu do najbliższego poziomu kwantyzacji. Wartości próbek, które zostaną zaokrąglone do tego samego poziomu y_k , tworzą przedział kwantyzacji D_k . Granice przedziałów leżą w równej odległości od najbliższych poziomów, choć poziom kwantyzacji niekoniecznie musi leżeć w środku przedziału.

W wyniku kwantowania próbka x_n zostanie zastąpiona wartością x_n^* , najbliższym jej poziomem kwantyzacji. Błąd zaokrąglenia (błąd kwantyzacji) wynosi $e_n = x_n - x_n^*$, jego moc chwilowa $e_n^2 = (x_n - x_n^*)^2$, a moc średnia

$$\sigma_e^2 = \frac{1}{M} \sum_{n=1}^M (x_n - x_n^*)^2 \quad (6)$$

Wzór (6) należałoby przepisać w takiej postaci, aby wystąpiły w nim wartości poziomów kwantyzacji. Można to uczynić, jeśli sumę po wszystkich próbkach sygnału rozdzielimy na L sum, każda z nich zawierająca próbki należące do tego samego przedziału kwantyzacji:

$$\sigma_e^2 = \frac{1}{M} \sum_{n=1}^M (x_n - x_n^*)^2 = \frac{1}{M} \sum_{k=1}^L \sum_{x_n \in D_k} (x_n - y_k)^2 \quad (7)$$

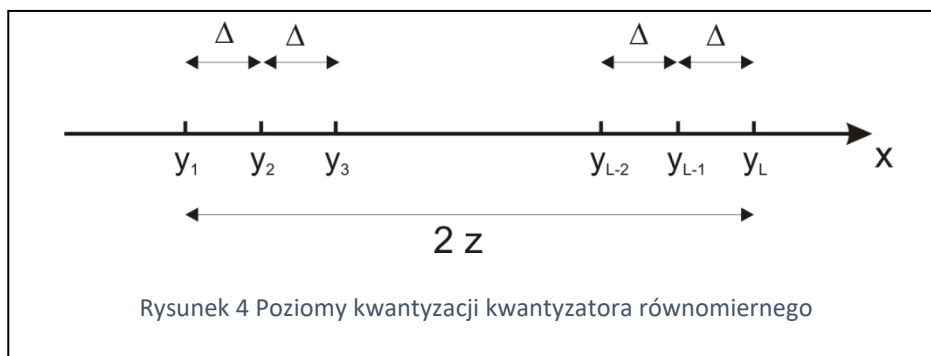
Mając dane poziomy kwantyzacji i próbki sygnału, można obliczyć średnią moc błędu (szumu) kwantyzacji ze wzoru (7). Można też szukać poziomów kwantyzacji minimalizujących moc błędu kwantyzacji. To zadanie nie jest już tak proste, gdyż poziomy kwantyzacji wpływają na przedziały kwantyzacji, wykorzystywane również we wzorze (7). Można jedynie uruchomić algorytm iteracyjny, obliczając na przemian poziomy i przedziały. Nosi on nazwę algorytmu Lloyd'a, Lloyd'a-Maxa, K-średnich. Zapoznamy się z nim omawiając kwantyzację wektorową.

Minimalizacja średniej mocy błędu kwantowania oznacza maksymalizację $SNR_q = \frac{\sigma_x^2}{\sigma_e^2}$ – stosunku mocy sygnału do mocy błędu kwantowania (wzór 4). Moc sygnału obliczymy ze wzoru:

$$\sigma_x^2 = \frac{1}{M} \sum_{n=1}^M (x_n)^2 \quad (8)$$

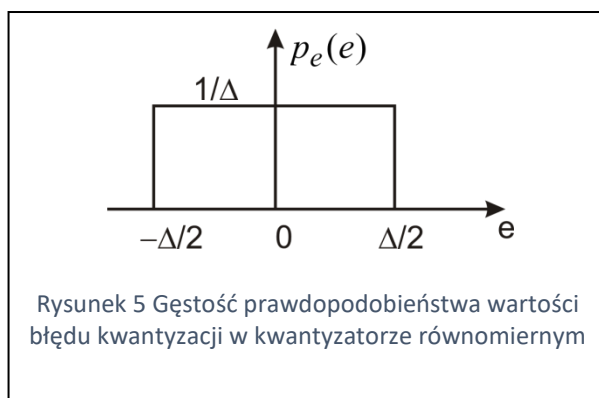
2.3 Kwantyzator równomierny

Popularne przetworniki analogowo-cyfrowe wykorzystują najczęściej kwantyzację równomierną. W kwantyzatorze równomiernym odległości sąsiednich poziomów są równe (rys.4). W przypadku sygnałów audio są one symetrycznie rozmieszczone od wartości (-z) do (+z). Jest to zakres pracy kwantyzatora, próbki większe od z lub mniejsze od (-z) są zaokrąglane do poziomów krańcowych (y_1 lub y_L), najczęściej z dużym błędem wynikającym z przesterowania.



Jeżeli nie występuje przesterowanie, wówczas błąd kwantyzacji nie przekracza połowy odległości sąsiednich poziomów: $|e| \leq \frac{\Delta}{2}$. Przyjmując równomierny rozkład wartości błędu w przedziale $-\frac{\Delta}{2} < e \leq \frac{\Delta}{2}$ (rys.5), można obliczyć moc błędu kwantyzacji:

$$\sigma_e^2 = \int_{-\infty}^{\infty} p_e(e) e^2 de = \int_{-\Delta/2}^{\Delta/2} \frac{1}{\Delta} e^2 de = \frac{2}{\Delta} \int_0^{\Delta/2} e^2 de = \frac{2}{\Delta} \left[\frac{1}{3} e^3 \right]_0^{\Delta/2} = \frac{2}{\Delta} \left(\frac{1}{3} \frac{\Delta^3}{8} \right) = \frac{1}{12} \Delta^2 \quad (9)$$



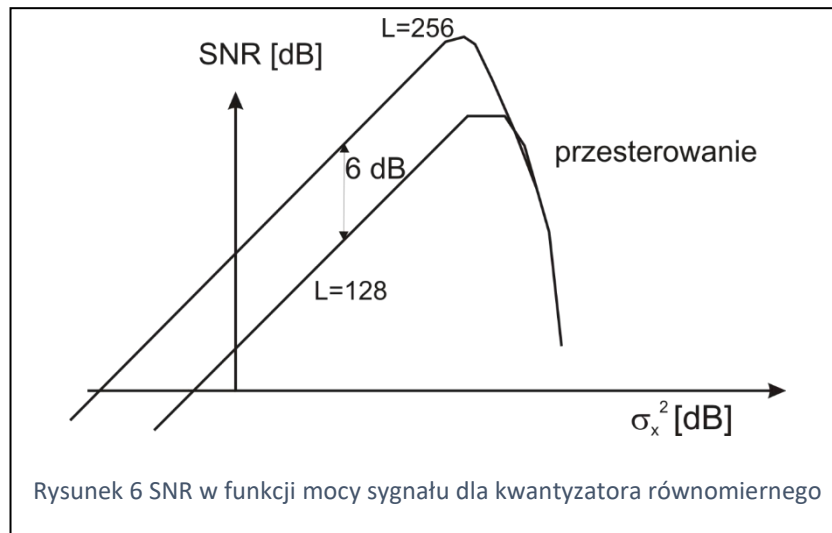
Podstawiając $\Delta = 2z/(L - 1) \approx 2z/L$ obliczymy SNR_q kwantyzatora równomiernego w decybelach:

$$SNR_q [dB] = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} = 10 \log_{10}(\sigma_x^2) - 10 \log_{10} \left(\frac{1}{12} \Delta^2 \right) =$$

$$= \sigma_x^2 [dB] - 10 \log_{10} \left(\frac{4z^2}{12L^2} \right) = \sigma_x^2 [dB] - 20 \log_{10}(z) + 20 \log_{10}(L) + 4.77$$

(10)

Ze wzoru (10) wynika, że SNR podąża za mocą sygnału. Ponieważ moc szumu kwantyzacji nie zależy od mocy sygnału (zależy jedynie od wartości Δ), wobec tego $SNR_q = \frac{\sigma_x^2}{\sigma_e^2}$ jest proporcjonalne do σ_x^2 . Należy jednak pamiętać, że wzór (10) opisuje działanie kwantyzatora równomiernego bez przesterowania. Gdy pojawi się przesterowanie, błąd kwantyzacji szybko rośnie i SNR spada (rys.6).



Jakość sygnału (reprezentowana przez SNR) oczywiście rośnie gdy rośnie liczba poziomów kwantyzacji. Podstawiając $L = 2^b$, gdzie b to liczba bitów przeznaczona na zakodowanie jednej próbki, do wzoru (10) otrzymuje się:

$$\begin{aligned} SNR_q [dB] &= \sigma_x^2 [dB] - 20 \log_{10}(z) + 20 \log_{10}(2^b) + 4.77 = \\ &= \sigma_x^2 [dB] - 20 \log_{10}(z) + 20b \log_{10}(2) + 4.77 = \\ &= \sigma_x^2 [dB] - 20 \log_{10}(z) + 6.02b + 4.77 \end{aligned}$$

(11)

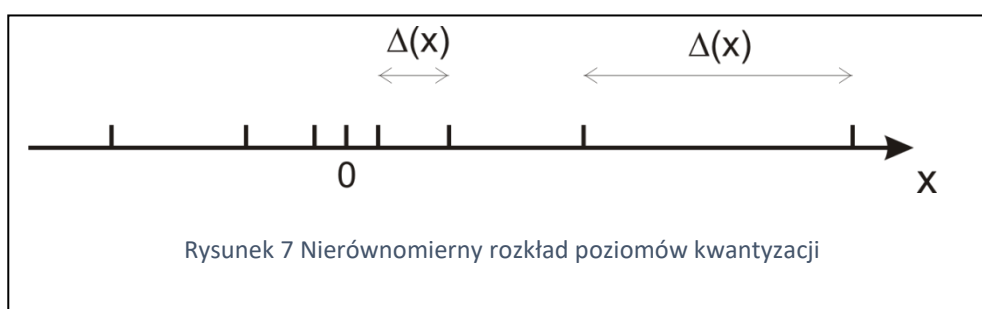
Zwiększenie przepływności binarnej o 1 bit na próbkę pociąga za sobą wzrost SNR o około 6 dB. Nazwijmy to zasadą „**6 dB na bit**”. Aby ją zrozumieć, nie trzeba formalnych obliczeń: Dodanie jednego bitu prowadzi do dwukrotnego zwiększenia liczby poziomów kwantyzacji. Odległości między nimi zmniejszą się dwukrotnie, a więc dwukrotnie zmniejszą wartości błędów kwantyzacji. Dwukrotne zmniejszenie amplitudy to czterokrotne zmniejszenie mocy szumu kwantyzacji. SNR wzrośnie więc 4-krotnie. Ponieważ $10 \log_{10}(4) \approx 6$, to mamy przyrost SNR o 6 dB. Zaznaczono to na rys.6, porównując SNR dla przetwornika 7-bitowego (128 poziomów kwantyzacji) i 8-bitowego (256 poziomów).

Największą wadą kwantyzatora równomiernego jest szybki spadek SNR dla sygnałów o małej amplitudzie. W telefonii oznacza to niską jakość sygnałów cichych lub wytłumionych w łączach analogowych doprowadzających sygnał do przetwornika analogowo-cyfrowego.

W przypadku gdy dysponujemy większą przepływnością binarną, spadek jakości nie będzie zauważalny, gdyż wystąpi dla sygnałów o bardzo niskiej amplitudzie, wręcz niesłyszalnych. Jest to przypadek CD-Audio, gdzie $b=16$ bitów na próbkę i przy pełnym wysterowaniu kwantyzatora można spodziewać się SNR o wartości ponad 90 dB.

2.4 Kwantyzator nierównomierny i standard PCM G.711

Nierównomierny rozkład poziomów kwantyzacji pozwala na osiągnięcie lepszych parametrów przy tej samej liczbie poziomów kwantyzacji. W szczególności, można tak rozłożyć poziomy, aby osiągnąć maksymalną wartość SNR_q dla danego sygnału. W telefonii ważniejszy jest jednak inny cel: utrzymać akceptowalną wartość SNR_q niezależnie od mocy sygnału mowy. Ten cel można osiągnąć zagęszczając siatkę poziomów dla próbek o małej amplitudzie i rozszerzając ją dla próbek o dużej amplitudzie (rys.7). W ten sposób przy kwantowaniu sygnałów cichych powstaje szum kwantyzacji o małej mocy a przy kwantowaniu sygnałów głośniejszych – szum o większej mocy, zaś stosunek mocy sygnału do szumu kwantyzacji jest niemal stały: $SNR_q = \frac{\sigma_x^2}{\sigma_e^2} \approx const.$ Taki rozkład poziomów nazywamy rozkładem logarytmicznym. Zasada logarytmiczna nie może być utrzymana dla bardzo cichych sygnałów, gdyż nie jest możliwe ciągłe zagęszczanie siatki poziomów z uwagi na ich ograniczoną liczbę.



Kwantyzator nierównomierny może być opisany przez podanie wartości wszystkich poziomów. Innym sposobem opisu jest podanie **charakterystyki kompresji** – funkcji przekształcającej nierównomierny rozkład poziomów na rozkład równomierny. Dla kwantyzatora o logarytmicznym rozkładzie poziomów kwantyzacji jest to zmodyfikowana funkcja logarytmiczna, która dla wartości argumentu bliskiego zeru staje się funkcją liniową.

Kwantyzator logarytmiczny zastosowano w najstarszym lecz ciągle popularnym standardzie kodowania sygnału telefonicznego Pulse Code Modulation (PCM) G.711 [4]. Warto zapoznać się z podstawowymi parametrami tego kodera:

- Pasmo sygnału mowy: 300-3400 Hz
- Częstotliwość próbkowania 8 kHz
- Liczba poziomów kwantyzacji: 256
- Liczba bitów na próbkę: 8
- Przepływność binarna: 8000 próbek/s razy 8 bitów/próbkę= 64000 bit/s
- Rozkład poziomów kwantyzacji: logarytmiczny typu A lub typu μ .

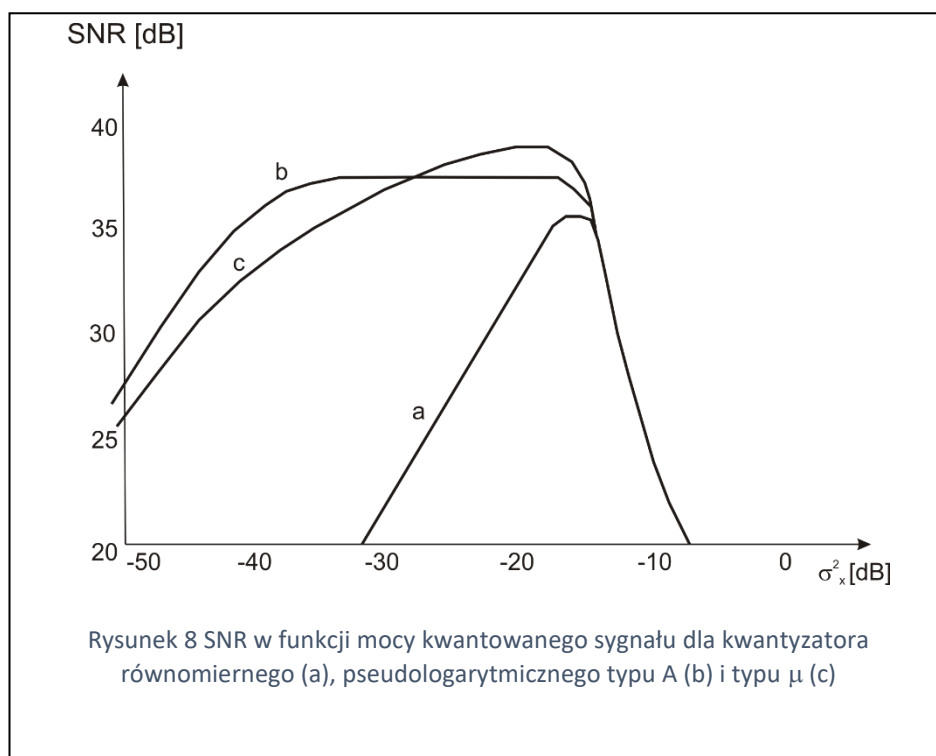
Istnieją dwa warianty standardu: z charakterystyką kompresji typu A (obowiązuje np. w Europie) i typu μ (obowiązuje np. w USA). Jeśli założymy, że zakres pracy kwantyzatora rozciąga się od (-1) do (+1), to odwzorowanie nierównomiernej siatki poziomów na siatkę równomierną przedstawia się następująco:

$$y = \operatorname{sgn}(x) \frac{\ln(1 + \mu|x|)}{\ln(1 + \mu)} \quad (12)$$

lub

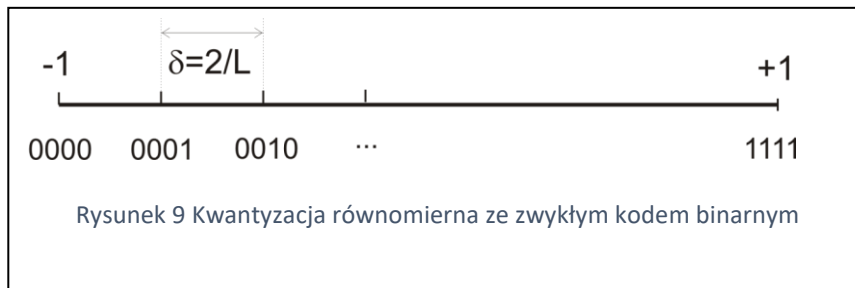
$$y = \begin{cases} \frac{Ax}{1 + \ln(A)}, & \text{gdy } 0 < x < 1/A \\ \operatorname{sgn}(x) \frac{1 + \ln(A|x|)}{1 + \ln(A)}, & \text{gdy } 1/A < x < 1 \end{cases} \quad (13)$$

Wzór (12) opisuje charakterystykę kompresji typu μ (wartość $\mu=255$), a (13) typu A (wartość $A=87.6$). Dla wartości $|x| \ll 1$ oba odwzorowania stają się liniowe co oznacza, że rozkład poziomów kwantyzacji staje się równomierny. Na rys.8 pokazano zależność SNR_q od mocy sygnału wejściowego dla kwantyzatora równomiernego, oraz kwantyzatora nierównomiernego z charakterystyką typu A i typu μ . Wyniki otrzymano dla sztucznego sygnału o eksponencjalnym rozkładzie wartości próbek, dobrze „dopasowanym” do pseudologarytmicznego rozkładu poziomów kwantyzacji. Z tego względu kwantyzatory nierównomierne wykazały większą wartość SNR niż kwantyzator równomierny przy pełnym wystrojeniu. Wykonując symulacje dla sygnału sinusoidalnego (ćwiczenie „Kwantyzacja skalarna” opisane w Module 4) zauważymy, że lepsze wyniki zapewnia kwantyzator równomierny. Najważniejsze jest jednak to, że kwantyzator pseudologarytmiczny zapewnia stosunkowo wysoką wartość SNR zarówno dla sygnału głośnego jak i cichego. Kwantyzator równomierny działa dobrze jedynie przy pełnym wystrojeniu, niemal na granicy przesterowania.



Wartości SNR pokazane na rys.8 odnoszą się jedynie do szumu kwantyzacji, nie uwzględnia się tu szumu w kanale transmisyjnym. Transmituje się b bitów na każdą próbkę sygnału (w standardzie G.711 8 bitów). Szum w kanale powoduje przekłamania bitów (stopa błędów P_e). W przypadku przekłamania w odbiorniku zostaje odtworzona próbka o błędnej wartości amplitudy, co może zostać odebrane jako trzask i wpłynąć na jakość odebranego sygnału mowy. Ze względu na nierównomierny rozkład poziomów kwantyzacji, analiza wpływu szumu w kanale na sygnał mowy przesyłany wg standardu G.711 nie jest prosta. Dokonamy zatem analizy uproszczonego problemu. Założmy, że:

- Rozkład poziomów kwantyzacji jest równomierny
- Zakres pracy kwantyzatora rozciąga się od (-1) do (+1).
- Poziomy kwantyzacji są zakodowane z wykorzystaniem zwykłego kodu binarnego (słowa kodowe od 000...0 do 111...1 - Rys.9.
- Stopa błędów binarnych wynosi P_e .



Załóżmy, że znamy moc sygnału $\sigma_x^2 = \int x^2 p(x) dx$. Moc szumu kwantyzacji podano we wzorze (9): $\sigma_e^2 = \frac{\delta^2}{12} = \frac{1}{3L^2}$.

Słowo kodowe opisujące poziom kwantyzacji składa się z b bitów:

$bit_{b-1} \dots bit_2 \ bit_1 \ bit_0$

Przekłamanie najmniej znaczącego bitu (bit_0) oznacza przejście do sąsiedniego poziomu kwantyzacji (błąd δ), przekłamanie kolejnych bitów jest przyczyną następujących błędów:

$\pm 2^{(b-1)}\delta \quad \dots \quad \pm 4\delta \quad \pm 2\delta \quad \pm \delta$

Związane z tym moce sygnału błędu otrzymamy, podnosząc wartości błędu do kwadratu:

$2^{2(b-1)}\delta^2 \quad \dots \quad 16\delta^2 \quad 4\delta^2 \quad \delta^2$

Każdy z tych błędów może wystąpić niezależnie z prawdopodobieństwem P_e . Średnią moc błędu wywołanego przekłamaniami bitów obliczymy, dodając moce poszczególnych błędów, pomnożone przez prawdopodobieństwo ich wystąpienia (P_e):

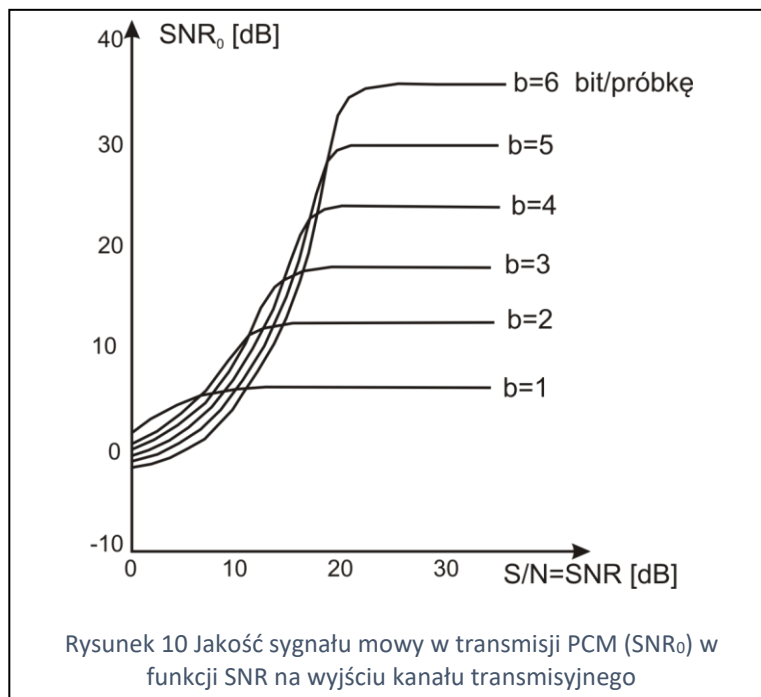
$$\begin{aligned} \sigma_n^2 &= P_e \delta^2 [1 + 4 + 16 + \dots + 2^{2(b-1)}] = P_e \delta^2 \frac{4^b - 1}{4 - 1} \cong \\ &\cong P_e \delta^2 \frac{4^b}{3} = P_e \delta^2 \frac{L^2}{3} = P_e \frac{4}{L^2} \frac{L^2}{3} = \frac{4}{3} P_e \end{aligned} \tag{14}$$

Wykorzystaliśmy tu wzór na sumę wyrazów ciągu geometrycznego. Wynik nie jest zaskakujący: moc błędu wynikającego z przekłamań bitów jest proporcjonalna do prawdopodobieństwa przekłamania. Ostatecznie na jakość sygnału wpływa zarówno szum kwantyzacji jak i błąd wynikający z przekłamań.

$$SNR_0 = \frac{\sigma_x^2}{\sigma_e^2 + \sigma_n^2} = \frac{\sigma_x^2}{\frac{1}{3L^2} + \frac{4}{3}P_e} \tag{15}$$

Prawdopodobieństwo przekłamań maleje wraz ze wzrostem SNR na wyjściu kanału. Gdybyśmy bity transmitowali w kodzie bipolarnym, to stopę błędów możemy odczytać ze wzoru (45), Moduł 2. Mając dobry kanał transmisyjny (wysoka wartość SNR na jego wyjściu), błędy związane z przekłamaniami możemy pominąć. Jakość sygnału mowy będzie związana wyłącznie z szumem kwantyzacji, a ten zależy od liczby bitów na próbkę b (zasada 6 dB/bit). Pokazano to na rys. 10.

Jeśli kanał jest niskiej jakości (niska wartość SNR na jego wyjściu), wówczas dominującym zakłóceniem jest błąd wynikający z przekłamań transmitowanych bitów. Wówczas jakość mowy (SNR_0) maleje wraz ze zmniejszaniem się SNR. Co więcej, stosując wysokiej jakości kwantyzator (duża liczba bitów na próbkę) narażamy się bardziej na wpływ szumu w kanale. Ma to związek z szybkością transmisji – szybka transmisja jest bardziej wrażliwa na szum w kanale. Np. stosując wspomniany kod bipolarny, szybsza transmisja (większa wartość b) oznacza krótszy czas trwania symbolu transmisyjnego (T), co zgodnie ze wzorem (45), Moduł 2, oznacza większe prawdopodobieństwo przekłamania i większy spadek jakości mowy (SNR_0) – rys.10.

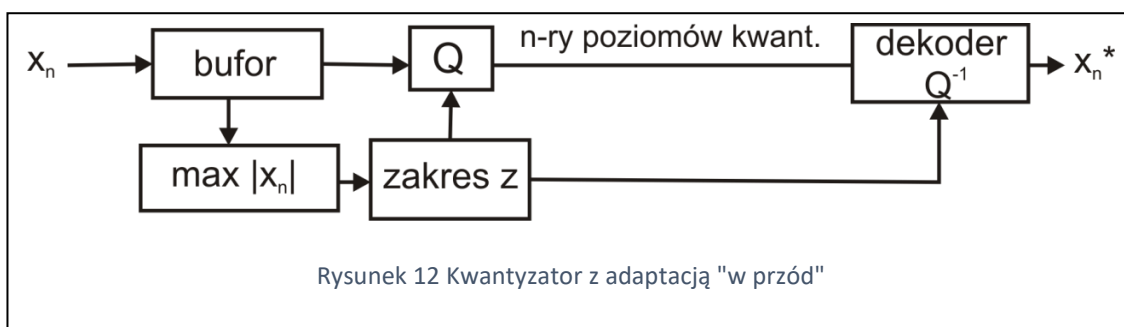


2.5 Kwantyzator adaptacyjny

Stosowanie kwantyzatora pseudologarytmicznego w pewnym stopniu rozwiązuje problem zależności SNR_q od mocy sygnału. Trzeba jednak podkreślić, że sygnały o małej mocy są kwantowane z wykorzystaniem tylko części z całkowitej liczby L poziomów. Z tego względu lepszym rozwiązaniem jest zagęszczanie siatki poziomów kwantyzacji dla sygnałów o małej amplitudzie i rozciąganie dla sygnałów o większej amplitudzie, czyli adaptacja zakresu pracy kwantyzatora do amplitudy kwantowanego sygnału – rys 11.



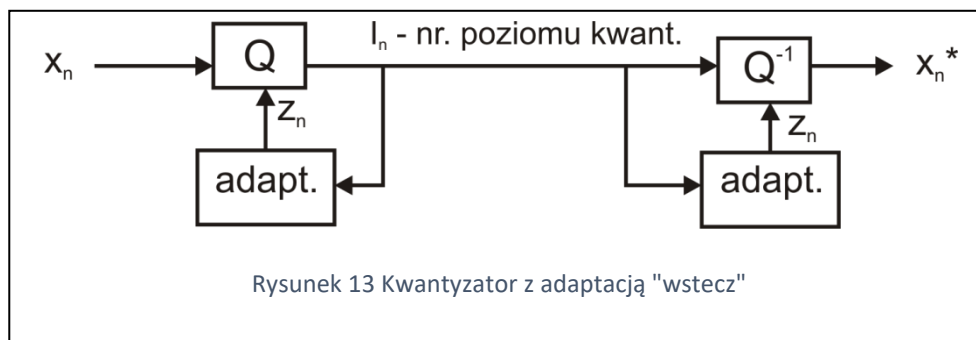
Są dwie grupy algorytmów adaptacji kwantyzatora: adaptacja „w przód” (forward) i „wstecz” (backward). Metodę adaptacji „w przód” pokazano na rys.12.



Po zgromadzeniu pewnej liczby próbek w buforze, znajduje się próbkę o największej wartości bezwzględnej. Następnie dobiera się zakres pracy kwantyzatora z tak, aby nie był mniejszy od tej próbki. Istnieje ograniczona liczba dopuszczalnych zakresów pracy, gdyż informacja o wybranym zakresie musi zostać wysłana do dekodera. Dopiero w tym momencie można rozpocząć kwantowanie, tzn. przypisywanie każdej próbce

odpowiedniego poziomu kwantowania i wysyłanie b bitów do dekodera. Dekodowanie wymaga znajomości zakresu pracy i numerów poziomów kwantyzacji. Adaptację przeprowadza się co kilka-kilkanaście próbek. Zaletą adaptacji „w przód” jest brak przesterowań, wadą jest opóźnienie proporcjonalne do pojemności bufora. Nie jest też wygodne przesyłanie informacji dodatkowej, tzn. aktualnego zakresu pracy. Adaptacja „w przód” jest stosowana w koderach audio wyższej jakości, np. w koderze MP3, ze względu na brak przesterowań.

Do transmisji mowy telefonicznej stosuje się najczęściej adaptację „wstecz”. Nie przesyła się tu informacji dodatkowej, a zakres pracy kwantyzatora jest synchronicznie aktualizowany w koderze i dekodrze – rys.13.



Zakres pracy zmienia się co próbkę -najczęściej jest on mnożony przez współczynnik adaptacji M przypisany wybranemu poziomowi kwantyzacji:

$$z_{n+1} = z_n M(I_n) \quad (16)$$

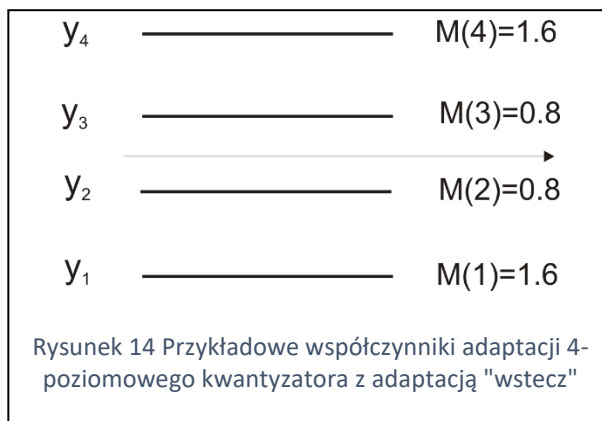
Gdzie: z_n – zakres pracy dla n -tej próbki kwantowanego sygnału,

I_n – numer poziomu kwantyzacji wybrany dla n -tej próbki,

$M(I_n)$ – współczynnik adaptacji (mnożnik) przypisany poziomowi kwantyzacji o numerze I_n

Na rys.14 pokazano przykład 4-poziomowego kwantyzatora z adaptacją „wstecz”. Wewnętrznym poziomom kwantyzacji przypisane są mnożniki mniejsze od 1, a zewnętrznym – większe od 1. Dzięki temu algorytm działa prawidłowo: zawęża siatkę poziomów (mnoży przez 0.8) gdy próbki sygnału mają niewielką amplitudę i są zaokrąglane do poziomów kwantyzacji bliskich zeru i rozszerza siatkę poziomów (mnoży przez 1.6) gdy próbki mają większą amplitudę i są zaokrąglane do zewnętrznych poziomów. W ćwiczeniu laboratoryjnym „Kwantyzacja skalarna” można zmieniać współczynniki adaptacji. Wybierając np. wartości 1.01, 0.99, 0.99, 1.01 mamy bardzo powolną adaptację. Wybierając np. 2, 0.5, 0.5, 2 otrzymujemy szybką adaptację.

Gdy mamy np. 16 poziomów musimy zadeklarować 8 par mnożników (z reguły ich wartości są symetryczne, bo próbki dodatnie i ujemne sygnału mowy mają podobne wartości bezwzględne).



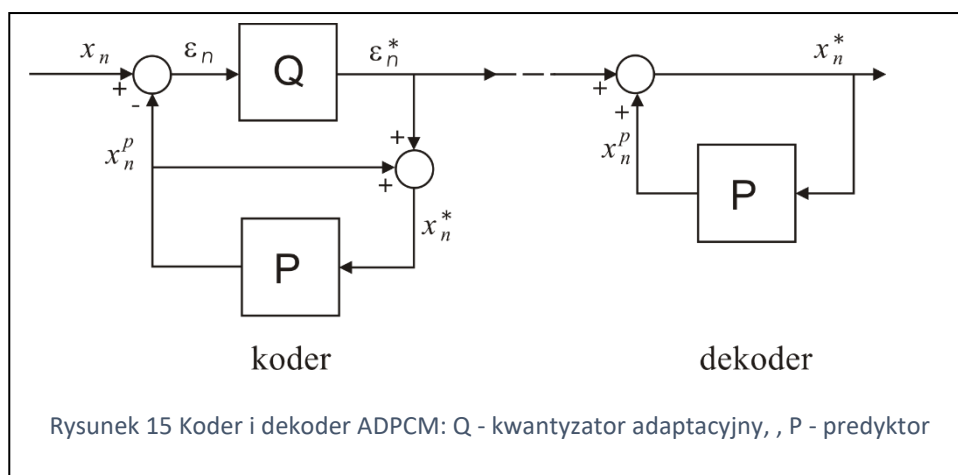
Zaletą adaptacji „wstecz” jest znikome opóźnienie (tak jak w PCM, wynosi ono 1 próbkę), nie przesyła się też zakresu pracy. Wadą jest brak zabezpieczenia przed przesterowaniem. Po gwałtownym zwiększeniu amplitudy sygnału, następuje seria przesterowanych próbek, zanim kwantyzator powiększy odpowiednio swój zakres

pracy. Przy przesyłaniu mowy telefonicznej te krótkie okresy przesterowania nie wpływają decydująco na jakość sygnału, w związku z tym w koderach ADPCM stosuje się niemal wyłącznie kwantyzatory adaptacyjne.

3 Predykcja i kodery ADPCM

3.1 Działanie kodera ADPCM

Predykcja to przewidywanie – w tym przypadku przewidywanie wartości bieżącej próbki sygnału na podstawie obserwacji próbek poprzednich. Przewidywalnej części sygnału nie trzeba transmitować, można ją odtworzyć w odbiorniku, używając do tego celu poprzednich próbek. Transmitować należy jedynie nieprzewidywalną część sygnału. Tę ideę wykorzystano m.in. w koderze ADPCM (**Adaptive Differential Pulse Code Modulation**) – rys.15.



Kwantowaniu (z wykorzystaniem adaptacji kwantyzatora) podlega sygnał różnicowy $\varepsilon_n = x_n - x_n^p$, gdzie x_n^p – predykcja bieżącej próbki x_n . Po stronie odbiorczej ta sama predykcja jest dodawana do skwantowanego sygnału różnicowego. Predykcja w koderze i w dekodery musi być obliczona w identyczny sposób. W dekodery do przewidywania próbki bieżącej możemy wykorzystać poprzednie próbki sygnału wyjściowego x_n^* , więc ten sam sygnał musimy wytworzyć po stronie nadawczej. Stąd częścią nadajnika jest kopia odbiornika, tzw. dekodery lokalny.

Ponieważ sygnał predykcji jest odejmowany od sygnału mowy, a potem jest dodawany, to sygnał wyjściowy dekodera x_n^* różni się od sygnału wejściowego kodera x_n jedynie błędem kwantyzacji powstającym w kwantyzatorze adaptacyjnym:

$$x_n^* = x_n + e_n \quad (17)$$

gdzie $e_n = \varepsilon_n^* - \varepsilon_n$ - błąd kwantyzacji, powstający przy kwantowaniu sygnału różnicowego (zwanego sygnałem błędu predykcji). Jeśli predykcja jest skuteczna, czyli sygnał błędu predykcji $\varepsilon_n = x_n - x_n^p$ ma mniejszą amplitudę niż sygnał wejściowy x_n , wówczas kwantyzator adaptacyjny „zagęszcza” siatkę poziomów kwantyzacji i błąd kwantyzacji maleje. Tym samym poprawia się jakość sygnału na wyjściu dekodera. Ta właściwość ADPCM wynika także ze wzoru (18):

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\sigma_x^2}{\sigma_\varepsilon^2} \cdot \frac{\sigma_\varepsilon^2}{\sigma_e^2} = G_p \cdot SNR_q \quad (18)$$

gdzie $\sigma_x^2, \sigma_\varepsilon^2, \sigma_e^2$ - moce sygnału wejściowego, różnicowego i szumu kwantyzacji, $SNR_q = \frac{\sigma_\varepsilon^2}{\sigma_e^2}$ - stosunek mocy sygnału wejściowego kwantyzatora do mocy szumu kwantyzacji (zależy głównie od liczby bitów na próbkę), $G_p = \frac{\sigma_x^2}{\sigma_\varepsilon^2}$ - stosunek mocy sygnału wejściowego do mocy błędu predykcji, zwany **zyskiem predykcji**.

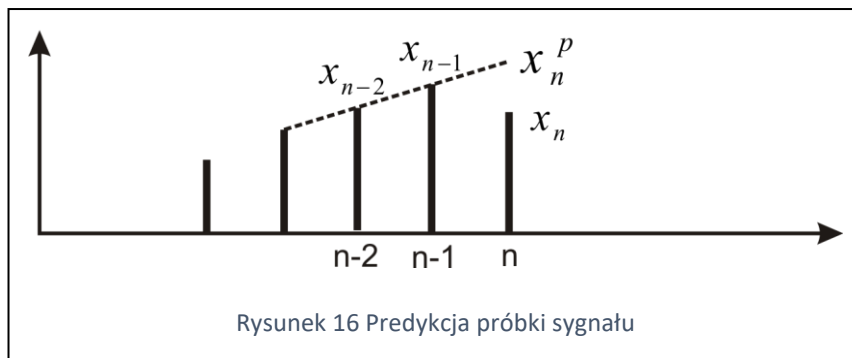
Logarytmując obie strony wzoru (18) i mnożąc przez 10, otrzymujemy wartości w decybelach:

$$SNR[dB] = SNR_q[dB] + G_p[dB] \quad (19)$$

Jeśli błąd predykcji jest mniejszy od sygnału wejściowego, wówczas $G_p > 1$ i $G_p[dB] > 0$. Następuje wówczas poprawa jakości sygnału skutkiem zastosowania predyktora.

3.2 Obliczanie predyktora

Jak skonstruować predyktor o wysokim zysku predykcji? Aby rozpatrzyć to zagadnienie, oddzielimy predyktor od układu ADPCM, przyjmując, że do przewidywania próbki sygnału wejściowego x_n możemy wykorzystać poprzednie próbki tego samego sygnału $x_{n-1}, x_{n-2}, \dots, x_{n-p}$ – rys.16.



Predyktory wykorzystywane w koderach mowy należą do **predyktorów liniowych**, tzn. predykcja x_n^p jest kombinacją liniową poprzednich próbek sygnału.

Najprostszy predyktor wykorzystuje wprost wartość poprzedniej próbki $x_n^p = x_{n-1}$. Lepszym rozwiązaniem jest użycie poprzedniej próbki pomnożonej przez współczynnik predykcji $x_n^p = a_1 x_{n-1}$. Ogólnie predyktor liniowy wyraża się wzorem:

$$x_n^p = \sum_{i=1}^p a_i x_{n-i} \quad (20)$$

Współczynniki predykcji a_1, a_2, \dots, a_p obliczamy na minimum mocy błędu predykcji:

$$\min \sigma_\varepsilon^2, \quad \sigma_\varepsilon^2 = E(x_n - x_n^p)^2 \quad (21)$$

gdzie E – operator uśredniania, który w praktyce jest zastępowany obliczaniem wartości średniej w bloku próbek.

Przyjmijmy, że znamy współczynniki autokorelacji sygnału:

$$R_i = E(x_n x_{n-i}) \quad (22)$$

Współczynnik R_0 jest równy mocy sygnału: $R_0 = E(x_n^2) = \sigma_x^2$.

Na początku obliczmy zysk predykcji najprostszego predyktora opisanego równaniem $x_n^p = x_{n-1}$.

$$\sigma_\varepsilon^2 = E((x_n - x_{n-1})^2) = E(x_n^2) - 2E(x_n x_{n-1}) + E(x_{n-1}^2) = 2R_0 - 2R_1$$

$$G_p = \frac{\sigma_x^2}{\sigma_\varepsilon^2} = \frac{R_0}{2R_0 - 2R_1} = \frac{1}{2(1 - \frac{R_1}{R_0})} \quad (23)$$

Jeżeli $\frac{R_1}{R_0} > 0.5$, wówczas $G_p > 1$ i stosowanie tego predyktora jest korzystne. Jeżeli $\frac{R_1}{R_0} < 0.5$, wówczas jakość sygnału na wyjściu układu ADPCM jest niższa niż ta, jaką oferuje sam kwantyzator.

Zajmijmy się teraz predyktorem ze współczynnikiem predykcji: $x_n^p = a_1 x_{n-1}$. Próbka sygnału błędu predykcji wynosi: $\varepsilon_n = x_n - x_n^p = x_n - a_1 x_{n-1}$, a moc tego sygnału

$$\sigma_\varepsilon^2 = E((x_n - a_1 x_{n-1})^2) = E(x_n^2) - 2a_1 E(x_n x_{n-1}) + (a_1)^2 E(x_{n-1}^2) = R_0 - 2a_1 R_1 + (a_1)^2 R_0$$

Minimum mocy błędu predykcji obliczymy, różniczkując σ_ε^2 względem a_1 i przyrównując do zera:

$$\frac{\partial \sigma_\varepsilon^2}{\partial a_1} = -2R_1 + 2a_1 R_0 = 0 \quad \Rightarrow \quad a_1 = \frac{R_1}{R_0}$$

(24)

Zysk dla optymalnego współczynnika predykcji $a_1 = \frac{R_1}{R_0}$ wynosi

$$G_p = \frac{\sigma_x^2}{\sigma_\varepsilon^2} = \frac{R_0}{R_0 - 2\frac{R_1}{R_0}R_1 + (\frac{R_1}{R_0})^2 R_0} = \frac{1}{1 - (\frac{R_1}{R_0})^2} \geq 1$$

(25)

Jest on większy od 1, gdyż współczynniki autokorelacji spełniają nierówność $|R_1| \leq R_0$. Wartość równą 1 osiąga dla nieskorelowanego sygnału, dla którego $R_1 = 0$. Wówczas również $a_1 = 0$. Predyktor nie jest w stanie przewidzieć próbki nieskorelowanego sygnału, w związku z tym wyłącza się.

Podobnie postępujemy w ogólnym przypadku, gdy predyktor ma p współczynników i jest opisany wzorem (20). Do wyznaczenia współczynników predykcji będą potrzebne współczynniki autokorelacji R_0, R_1, \dots, R_p . Współczynniki predykcji są rozwiązaniem układu równań liniowych [1]. Np., dla $p=3$ jest to następujący układ równań:

$$\begin{matrix} R_0 & R_1 & R_2 & a_1 & R_1 \\ R_1 & R_0 & R_1 & a_2 & R_2 \\ R_2 & R_1 & R_0 & a_3 & R_3 \end{matrix} \cdot \begin{matrix} a_1 \\ a_2 \\ a_3 \end{matrix} = \begin{matrix} R_1 \\ R_2 \\ R_3 \end{matrix} \quad (26)$$

Wykonując symulacje w ramach ćwiczenia laboratoryjnego „Kwantyzacja skalarna” można będzie porównać działanie kodera ADPCM z działaniem samego kwantyzatora. Różnica wartości SNR[dB] to zysk predykcji wyrażony w decybelach (wzór 19). Zysk predykcji zależy od predyktora (jest tym większy im więcej poprzedzających próbek wykorzystamy do przewidywania próbki bieżącej). W głównej jednak mierze zależy od samego sygnału. W przypadku sygnału nieskorelowanego predykcja nie jest możliwa. Podstawiając zerowe wartości współczynników autokorelacji (poza wartością R_0 , która jest mocą sygnału) do równania (26) otrzymamy zerowy wektor prawej strony i zerowe wartości współczynników predykcji. Predykcja $x_n^p = 0$ i zysk wynosi 1 (w decybelach 0). Z kolei sygnał o wartości stałej jest bezbłędnie przewidywalny z wykorzystaniem predykcji $x_n^p = x_{n-1}$. Tu zysk jest nieskończony. Bezbłędnie przewidywalny jest też sygnał harmoniczny, wystarczy znać 2 poprzednie próbki, aby obliczyć kolejną próbkę jako kombinację liniową tych 2 poprzednich. Będzie się o tym można przekonać, symulując koder ADPCM z predyktorem o co najmniej 2 współczynnikach predykcji. Należy się wówczas spodziewać wysokich wartości SNR.

3.3 Metody sekwencyjne (materiał dodatkowy)

Współczynniki predykcji można obliczyć ze współczynników autokorelacji sygnału (wzory 24, 26). Aby oszacować wartości współczynników autokorelacji, trzeba wpięrw zgromadzić pewną liczbę próbek sygnału. Oznacza to wprowadzenie opóźnienia, w kodowaniu mowy rzędu 10-30ms. Obliczone współczynniki trzeba ponadto przesać do odbiornika, żeby tam uruchomić identyczny predyktor. Aby tego uniknąć, w koderach ADPCM oblicza się współczynniki predykcji z poprzednich, do tego obarczonych błędem kwantyzacji próbek sygnału (x_n^* na rys.15). Na tej samej zasadzie działa algorytm adaptacji kwantyzatora metodą „wstecz”.

Algorytm sekwencyjny aktualizuje, co próbkę n , wektor współczynników predykcji przechowywany w pamięci. $\mathbf{a}(n) = [a_1(n), a_2(n), \dots, a_p(n)]^T$, gdzie T oznacza transpozycję. Predykcja jest obliczona jak we wzorze (20): $x_n^p = \sum_{i=1}^p a_i x_{n-i}^* = \mathbf{a}^T(n) \mathbf{X}_n^*$, gdzie $\mathbf{X}_n^* = [x_{n-1}^*, x_{n-2}^*, \dots, x_{n-p}^*]^T$ jest wektorem złożonym z p próbek sygnału poprzedzających próbkę bieżącą.

Aktualizacja współczynników predykcji odbywa się przez dodanie niewielkiej poprawki:

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \Delta \mathbf{a}(n)$$

(27)

Poprawka winna zmierzać w kierunku zmniejszenia mocy błędu predykcji $[\varepsilon_n^*]^2$. Na rys.17 pokazano proces aktualizacji predyktora o dwóch współczynnikach. W chwili n dysponujemy wektorem współczynników $\mathbf{a}(n)$. Kierunek wzrostu błędu predykcji pokazuje gradient $grad[\varepsilon_n^*]^2 = \nabla[\varepsilon_n^*]^2$. Poprawka $\Delta\mathbf{a}(n)$ winna zmierzać w przeciwną stronę: $\Delta\mathbf{a}(n) \propto -grad[\varepsilon_n^*]^2$.

Ponieważ $\varepsilon_n^* = x_n^* - x_n^p = x_n^* - \sum_{i=1}^p a_i x_{n-i}^* = x_n^* - \mathbf{a}^T(n)\mathbf{X}_n^*$, to gradient (wektor pochodnych) wynosi

$$grad[\varepsilon_n^*]^2 = \frac{\partial}{\partial \mathbf{a}(n)} [x_n^* - \mathbf{a}^T(n)\mathbf{X}_n^*]^2 = -2\varepsilon_n^*\mathbf{X}_n^*$$

Poprawka $\Delta\mathbf{a}(n)$ powinna mieć przeciwny znak, natomiast długość skoku niech będzie regulowana współczynnikiem szybkości adaptacji β .

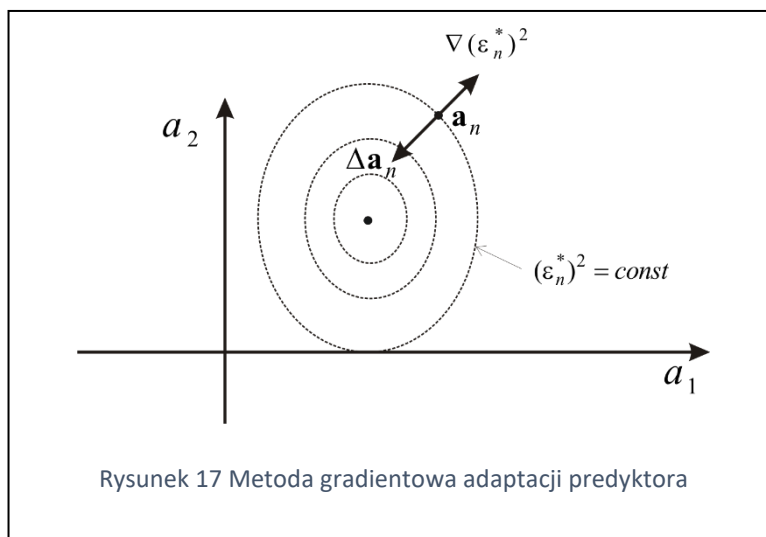
$$\Delta\mathbf{a}(n) = \beta\varepsilon_n^*\mathbf{X}_n^*$$

Ostatecznie otrzymuje się dość prosty w stosowaniu wzór

$$\mathbf{a}(n+1) = \mathbf{a}(n) + \beta\varepsilon_n^*\mathbf{X}_n^*$$

(28)

Ta metoda adaptacji nazywana jest **metodą stochastycznego gradientu**. Według wzoru (28) aktualizuje się współczynniki predykcji – synchronicznie w nadajniku i odbiorniku ADPCM. Adaptacja sekwencyjna nie wymaga przesyłania współczynników predykcji do dekodera.



Dobór szybkości adaptacji β nie jest sprawą łatwą. Za duża wartość współczynnika β prowadzi do niestabilności procesu adaptacji (zbyt duże kroki $\Delta\mathbf{a}(n)$). Wzór (28) pokazuje, że wielkość poprawki zależy ponadto od mocy sygnału. Rozwiązaniem tego problemu jest metoda stochastycznego gradientu z normalizacją, w której współczynnik β jest odwrotnie proporcjonalny do estymaty mocy sygnału:

$$\beta_n = \frac{L_\beta}{\hat{\sigma}_x^2(n) + M_\beta}$$

(29)

gdzie L_β reguluje szybkość adaptacji a M_β zapobiega dzieleniu przez zero.

W ćwiczeniu „Kwantyzacja skalarna” zaimplementowano metodę stochastycznego gradientu bez normalizacji i z normalizacją.

3.4 Koder ADPCM G.726

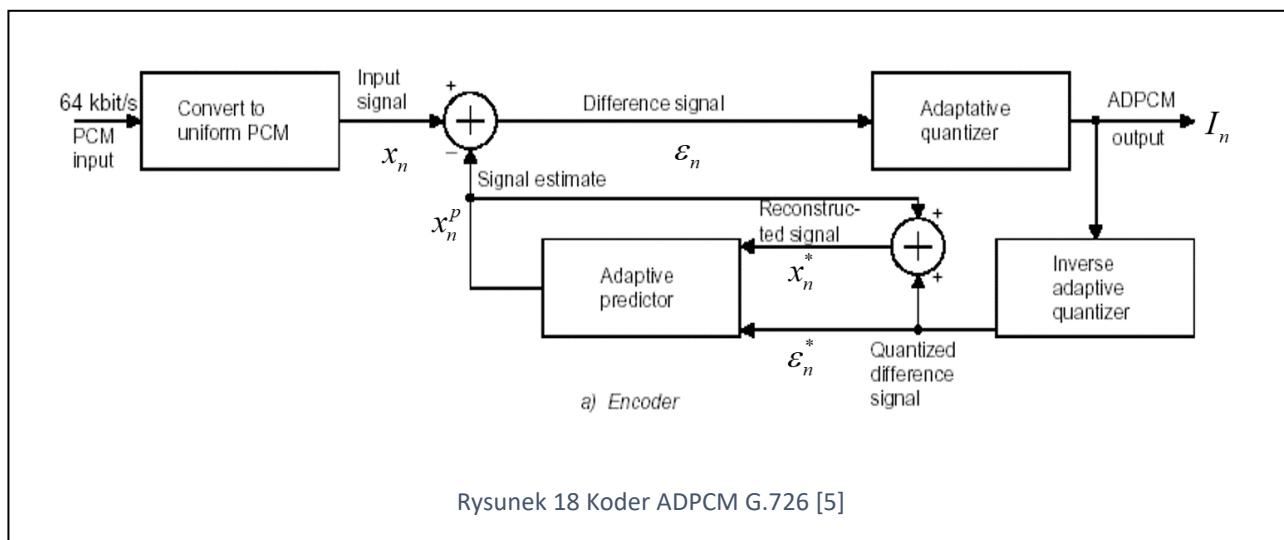
Jako przykład kodera ADPCM weźmy koder G.726, który (podobnie jak inne kodery, w tym znany już nam G.711) jest standardem ITU - Unii Telekomunikacyjnej z siedzibą w Genewie [5]. Podstawową funkcją tego kodera jest zamiana strumienia binarnego PCM o przepływności 64 kbit/s na strumień o 2 razy mniejszej

przeptywności. Poza przepływnością 32 kbit/s standard G.726 oferuje kilka innych przepływności. Podstawowe parametry kodera są następujące:

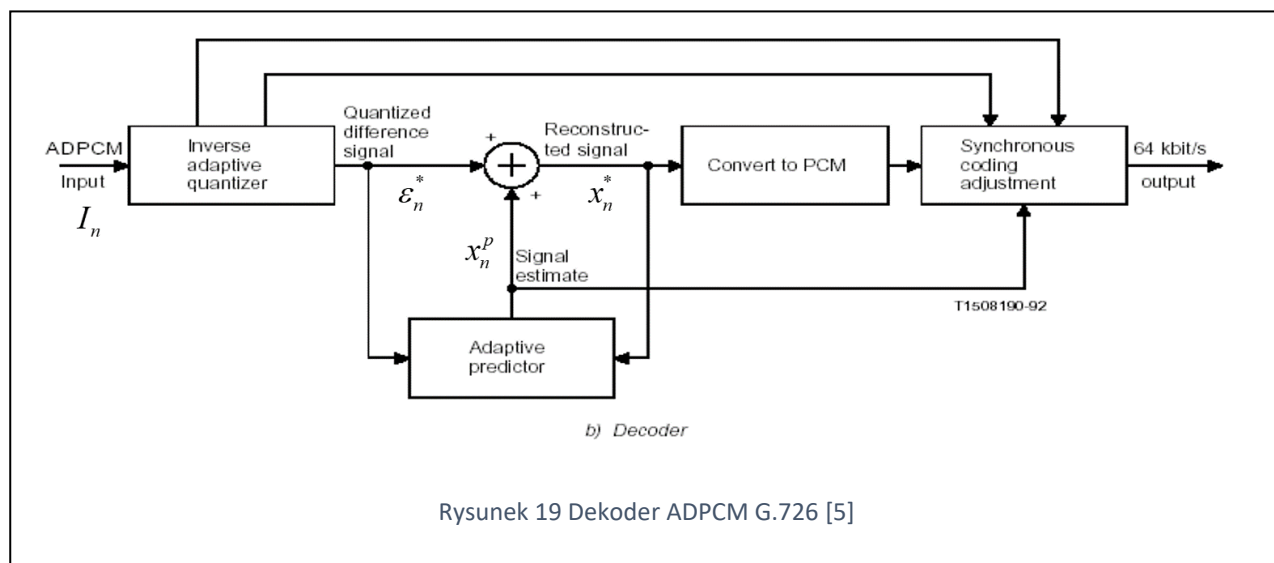
- Częstotliwość próbkowania 8 kHz
- Przepływności binarne 40, 32, 24, 16 kbit/s
- Zabezpieczenie przed kumulacją zniekształceń (przy wielokrotnym kodowaniu/dekodowaniu).

Odbiornik ADPCM generuje strumień binarny PCM o przepływności 64 kbit/s w taki sposób, aby w kolejnym koderze G.726 wybierane były te same poziomy kwantyzacji.

Schemat kodera, wg normy ITU [5], pokazano na rys.18, a dekodera – na rys.19.



Rysunek 18 Koder ADPCM G.726 [5]



Rysunek 19 Dekoder ADPCM G.726 [5]

Kwantyzator adaptacyjny ma następujące parametry:

- Liczba poziomów 31, 15, 7 lub 4 (dla przepływności 40, 32, 24 lub 16 kbit/s).
- Adaptacja „wstecz” wg wzoru (16): $z(n + 1) = z(n)M(|I_n|)$
- Adaptacja szybka dla mowy i wolna dla sygnałów modemowych, które są transmitowane w paśmie akustycznym
- Zabezpieczenie przed kumulacją błędów. Chodzi o to, aby odebrany z błędem numer poziomu kwantyzacji (I_n) nie spowodował trwałego rozejścia się zakresu pracy kwantyzatora i dekodera.

Predyktor różni się od opisanego wzorem (20), gdyż do przewidywania wykorzystuje nie tylko próbki sygnału mowy ale także próbki sygnału błędu predykcji:

$$x_n^p = \sum_{i=1}^2 a_i(n)x_{n-i}^* + \sum_{i=1}^6 b_i(n)\varepsilon_{n-i}^* \quad (30)$$

Współczynniki predykcji $a_1, a_2, b_1, b_2, \dots, b_6$ są adaptowane metodą sekwencyjną. Aby nie dopuścić do trwałego rozejścia się wartości współczynników predykcji w nadajniku i w odbiorniku, wprowadza się tłumienie o wartości $1 - \alpha = \frac{255}{256}$. Gdy, skutkiem przekłamania bitów, pojawi się błąd, jest on mnożony przez $\frac{255}{256}$ co próbkę i szybko zanika:

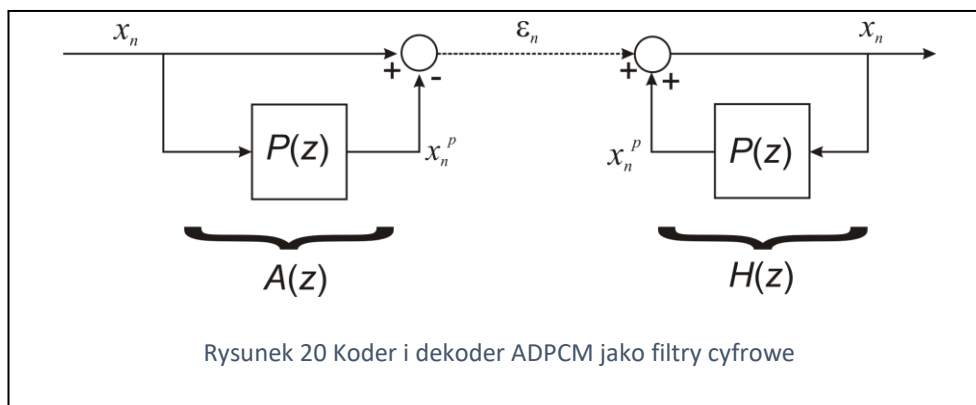
$$a_i(n+1) = a_i(n)(1 - \alpha) + \alpha \Delta a_i(n),$$

$$b_i(n+1) = b_i(n)(1 - \alpha) + \alpha \Delta b_i(n), \quad \alpha = \frac{1}{256}$$

Przy szybkości transmisji z przepływnością 32 kbit/s koder G.726 utrzymuje tę samą jakość sygnału co koder PCM o przepływności 64kbit/s.

3.5 Interpretacja predykcji w dziedzinie częstotliwości

Jeśli w schemacie kodera ADPCM (rys.15) pominiemy szum kwantyzacji (podstawiając $\varepsilon_n^* = \varepsilon_n$), wówczas na wejściu predyktora pojawi się sygnał x_n i schemat nadajnika będzie można przekształcić do postaci pokazanej na rys.20. Predyktor, opisany wzorem (20) jest w istocie filtrem cyfrowym o skończonej odpowiedzi impulsowej $\{0, a_1, a_2, \dots, a_p\}$ – patrz p.7.4, Moduł1. Jeżeli współczynniki predykcji są przez jakiś czas stałe (tak jest w koderach ze „skokową” adaptacją predyktora, np. co 10-30 ms), to do opisu filtru można wykorzystać transformatę Zet. Transmitancja filtru predyktora wówczas wynosi $P(z) = \sum_{i=1}^p a_i z^{-i}$.



Koder ADPCM, przetwarzający sygnał wejściowy na sygnał błędu predykcji, ma transmitancję

$$A(z) = 1 - P(z) = 1 - \sum_{i=1}^p a_i z^{-i} \quad (31)$$

Filtr $A(z)$ jest nazywany filtrem inwersyjnym.

Dekoder też można opisać w dziedzinie transformaty Zet: $X(z) = E(z) + X(z)P(z)$. Po przekształceniu $X(z)[1 - P(z)] = E(z)$ otrzymujemy transmitancję filtru reprezentującego dekodek ADPCM:

$$H(z) = \frac{X(z)}{E(z)} = \frac{1}{1 - P(z)} = \frac{1}{A(z)} \quad (32)$$

Filtr $H(z)$ jest nazywany filtrem syntezy predykcyjnej. Przetwarza on sygnał błędu predykcji na sygnał mowy.

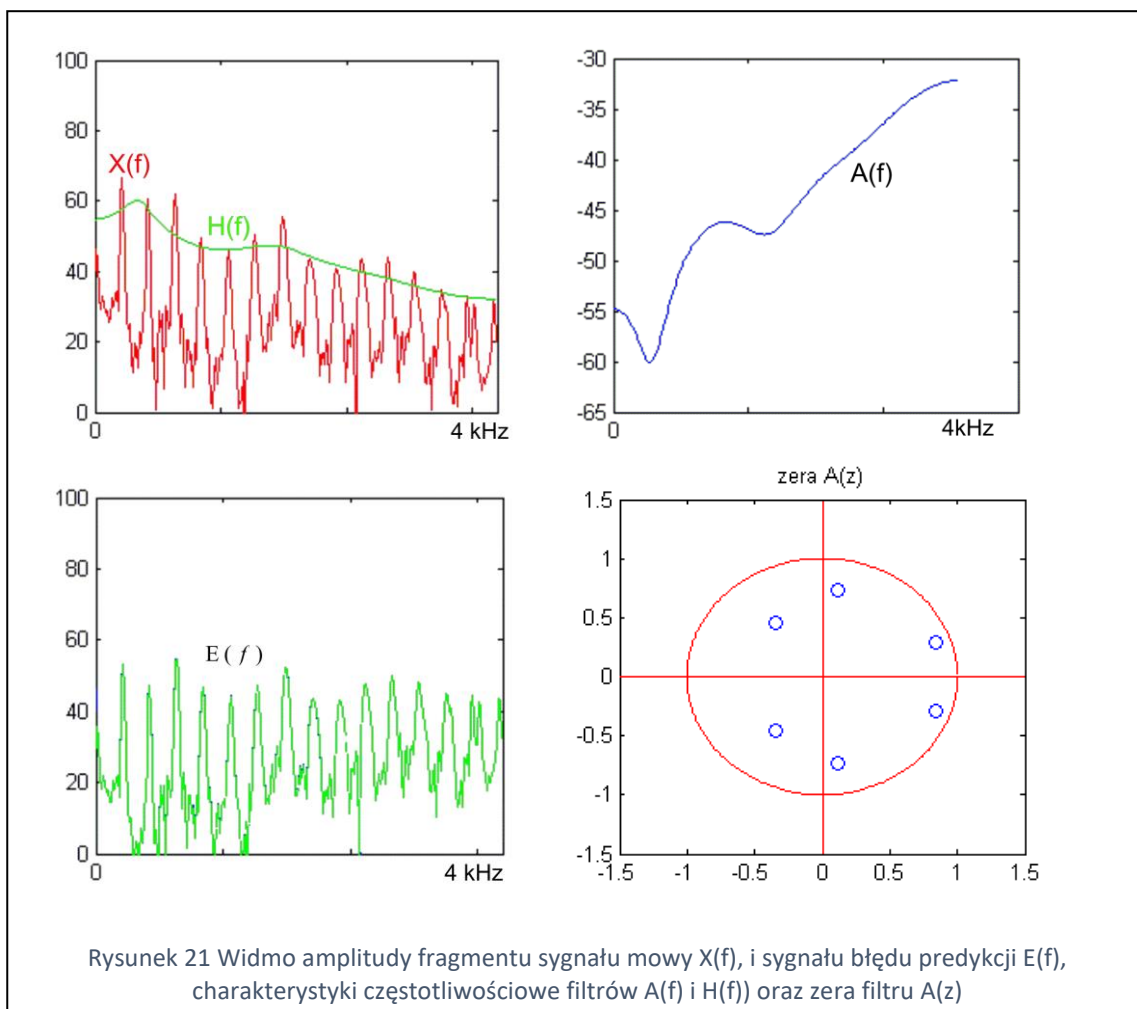
Filtr $A(z)$ jest filtrem o skończonej odpowiedzi impulsowej, posiadającym p zer. Filtr $H(z)$ jest filtrem o nieskończonej odpowiedzi impulsowej, posiadającym p biegunów w tych samych miejscach, gdzie filtr $A(z)$ ma zera.

Podstawiając $z = e^{j2\pi fT}$ ($\frac{1}{T} = f_s$ jest częstotliwością próbkowania) otrzymamy charakterystyki częstotliwościowe obu filtrów. Na rys.21 pokazano widma amplitudy sygnałów $X(f)$ i $E(f)$ i charakterystyki częstotliwościowe filtrów $A(f)$ i $H(f)$ otrzymane dla fragmentu mowy przetwarzanego w układzie z rys.20. Liczba współczynników predykcji wynosi $p=6$ i tyle jest zer filtru $A(z)$ i biegunów $H(z)$. Charakterystyki częstotliwościowe podano w decybelach, w zakresie częstotliwości od zera do połowy częstotliwości próbkowania. W związku z tym obserwujemy wpływ jedynie 3 zer na charakterystykę filtru $A(z)$ – pozostałe 3 zera tworzą niewidoczne na wykresach lustrzane odbicie. Bieguny $H(z)$ leżą w tych samych miejscach co zera

$A(z)$, ich położenie w obrębie koła o promieniu jednostkowym gwarantuje stabilność filtra $H(z)$. Minimom charakterystyki częstotliwościowej $A(f)$ odpowiadają maksima charakterystyki $H(f)$. Porównanie widma sygnału $X(f)$ i charakterystyki filtra $H(f)$ prowadzi do wniosku, że filtr syntezy predykcyjnej $H(f)$ jest dobrym modelem widma mowy, tym lepszym, im więcej mamy współczynników predykcji. Zauważmy, że charakterystyka $H(f)$ może mieć co najwyżej $p/2$ maksimów w zakresie częstotliwości od zera do połowy częstotliwości próbkowania. Im więcej współczynników predykcji tym lepsze dopasowanie $H(f)$ do widma sygnału. Najczęściej wystarcza $p=10$ współczynników predykcji.

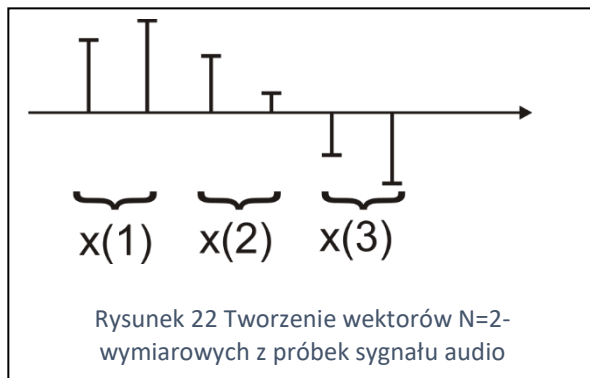
Jak wytłumaczyć to dopasowanie charakterystyki $H(f)$ do widma mowy? Zauważmy, że sygnał mowy jest przepuszczany przez filtr inwersyjny $A(z)$, na wyjściu którego powstaje sygnał błędu predykcji. Współczynniki predykcji obliczane są w taki sposób, aby zminimalizować energię sygnału błędu predykcji. W związku z tym minima charakterystyki $A(f)$ muszą pojawić się na tych częstotliwościach, na których sygnał mowy ma najwięcej energii (czyli na częstotliwościach formantowych). Minima $A(f)$ to maksima $H(f)$, stąd podobieństwo charakterystyki $H(f)$ do widma mowy. Po wytłumieniu rezonansów formantowych otrzymuje się sygnał błędu predykcji o płaskiej obwiedni widma $E(f)$. Prążkowy charakter tego widma wynika z quasiperiodycznego charakteru mowy dźwięcznej. Odległość prążków widmowych jest równa częstotliwości drgania strun głosowych.

Interpretacja częstotliwościowa liniowej predykcji przydaje się w koderach typu CELP i w tzw. wokoderach. Wykorzystuje się tam filtr $H(z)$ jako model widma mowy, a pobudzenie tego filtra wytwarza się w generatorze (wokodery) lub poddaje silnej kompresji (CELP). Filtr $H(z)$ odtwarza rezonanse formantowe, to on odpowiada za to, czy słyszymy głoskę „a” czy np. „e”.

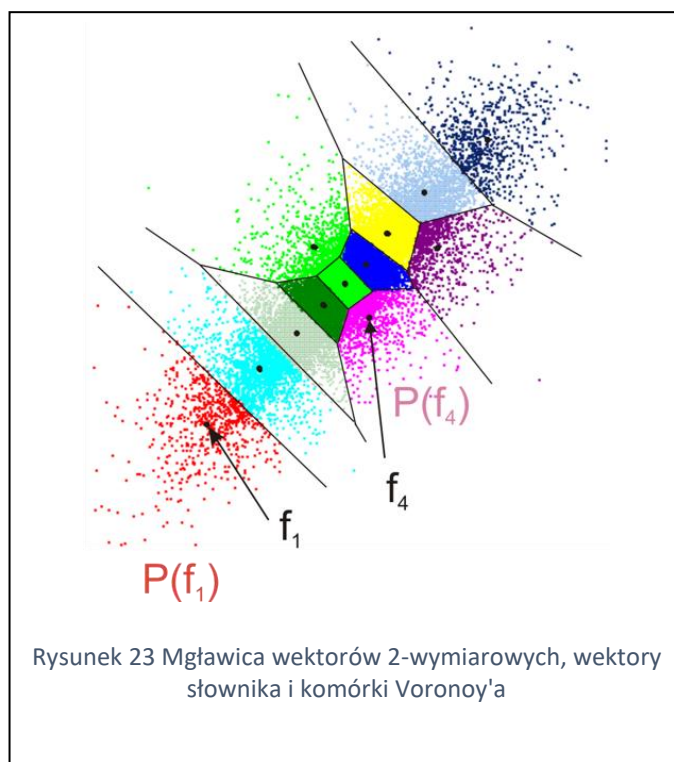


4 Kwantyzacja wektorowa

Kwantyzator skalarny przetwarza każdą próbkę sygnału dźwiękowego (lub każdy piksel obrazu) osobno, z tego względu nie może wykorzystać związków łączących sąsiednie próbki lub piksele. Cechę tę posiada **kwantyzator wektorowy (VQ – vector quantizer)**, przetwarzający N-wymiarowe wektory złożone z sąsiednich próbek (lub blisko siebie leżących pikseli). Na rys. 22 pokazano sposób tworzenia 2-wymiarowych wektorów z próbek sygnału audio.



Mając zbiór wektorów pobranych np. z frazy sygnału mowy możemy je przedstawić jako mgławicę w przestrzeni N-wymiarowej - rys.23.



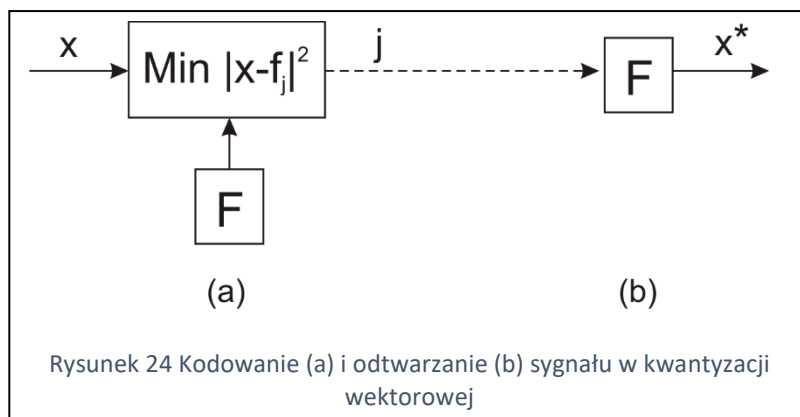
Kwantyzacja wektorowa polega na zaokrągleniu wektora z mgławicy do najbliższego wektora ze słownika $F = \{f_1, f_2, \dots, f_L\}$, pełniącego rolę zbioru poziomów kwantyzacji. Na rys.23 wektory słownika są czarnymi punktami. Zbiór wszystkich wektorów z mgławicy, dla których wektor f_k jest najbliższym wektorem słownika, tworzy k-tą komórkę Voronoy'a $P(f_k)$ (od nazwiska ukraińskiego matematyka, profesora Uniwersytetu Warszawskiego). Komórki Voronoy'a są wielowymiarowymi przedziałami kwantyzacji.

Jeśli przeznaczamy b bitów na zakodowanie jednej próbki sygnału, wówczas mamy bN bitów na wektor, co oznacza, że liczba wektorów w słowniku (zwanym też książką kodową) wynosi

$$L = 2^{bN} \quad (33)$$

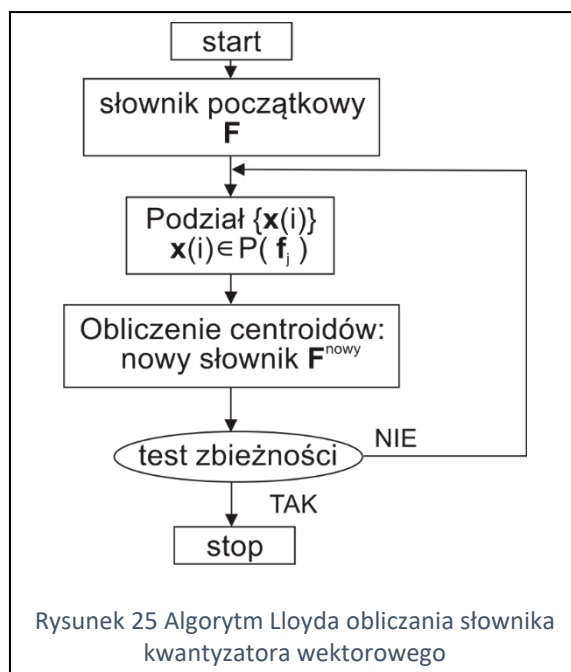
Na rys.24 pokazano koder i dekoder kwantyzacji wektorowej. Dla N-wymiarowego wektora próbek x poszukujemy najbliższego wektora w słowniku F , po czym, wykorzystując bN bitów, przesyłamy jego numer (j)

do dekodera. Tym numerem adresujemy wektor f_j w identycznym słowniku w dekodерze i wyprowadzamy N próbek skwantowanego sygnału.



Jeżeli podstawimy wymiar wektora $N=1$, wówczas wrócimy do kwantyzatora skalarnego. Słownik stanie się zbiorem poziomów kwantyzacji, a komórki Voronoy'a – przedziałami kwantyzacji.

Jak zaprojektować słownik kwantyzatora wektorowego? Rys.23 sugeruje, że granice komórek powinny przebiegać w równej odległości od najbliższych wektorów słownika, a wektor słownika powinien dobrze reprezentować wektory należące do jego komórki. Oznacza to, że wektor słownika winien być równy wartości średniej (centroidowi) wektorów z komórki. Te dwa warunki trudno jest spełnić od razu, ale można skonstruować algorytm iteracyjny, na przemian obliczając komórki i wektory słownika. Mając zbiór wektorów słownika, przeprowadzamy kwantyzację przydzielając każdy wektor z mgławicy do jednej z komórek Voronoy'a. Następnie obliczamy wartość średnią wektorów w każdej komórce otrzymując wektor nowego słownika. Taki algorytm nazywa się algorytmem Lloyd'a lub algorytmem K średnich. W istocie mamy tu problem klasteryzacji zbioru wektorów tworzących mgławicę. Sieć działań pokazano na rys.25.

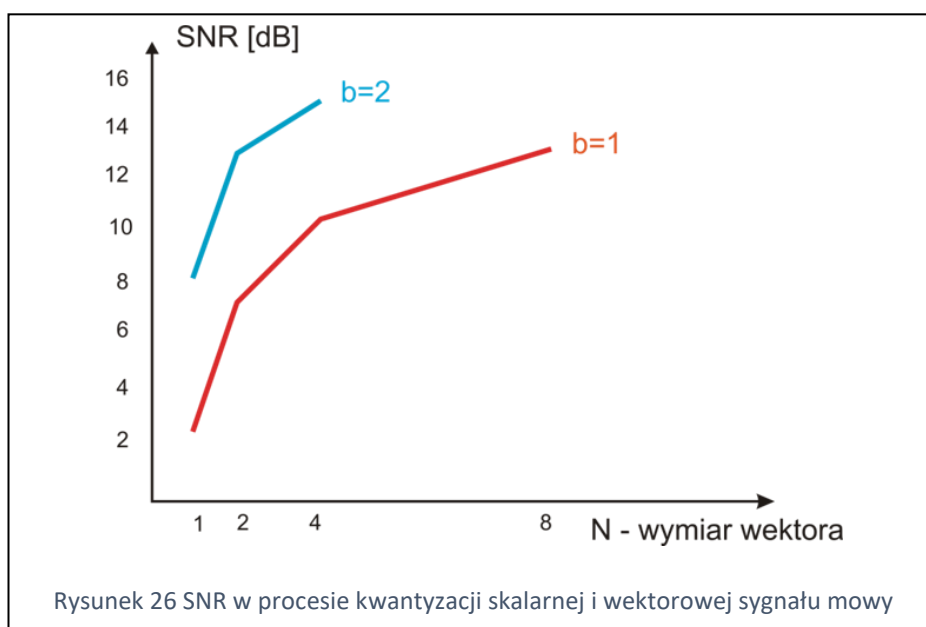


Działania algorytmu kończą się z chwilą osiągnięcia punktu stałego: w dwóch kolejnych iteracjach pojawia się ten sam słownik.

Słownik początkowy może być otrzymany metodą losowania L wektorów z mgławicy. Algorytm Lloyd'a nie gwarantuje otrzymania najlepszego z możliwych słowników, ale w każdej iteracji następuje zmniejszenie energii błędu kwantyzacji wektorów z mgławicy.

Wariantem algorytmu Lloyd'a jest algorytm LBG (od nazwisk jego twórców: Linde, Buzo, Gray). Polega on na uruchamianiu algorytmu Lloyd'a kolejno dla 1, 2, 4, 8, ... wektorów. W pierwszym uruchomieniu otrzymujemy wartość średnią wszystkich wektorów z mgławicy (bazy danych uczących). Otrzymany wektor rozszczepiamy na dwa blisko siebie leżące i uruchamiamy algorytm Lloyd'a dla dwóch wektorów. Po osiągnięciu zbieżności, rozszczepiamy każdy wektor i uruchamiamy algorytm dla 4 wektorów, itd. aż do otrzymania słownika liczącego L wektorów.

W Module 4 opisany jest symulator kwantyzacji wektorowej. Zainteresowanych zachęcam uruchomienia procedury Lloyd'a i LBG – można obserwować ewolucję wektorów słownika i komórki Voronoy'a. Interesujące jest porównanie kwantyzatora wektorowego i skalarne (ten ostatni otrzymujemy podstawiając wymiar wektora $N=1$). Porównanie należy przeprowadzić przy tej samej przepływności, tzn. tej samej liczbie b bitów na próbkę. Przy stałej wartości b bit/próbkę możemy zmieniać wymiar wektorów (N) jednocześnie zmieniając ich liczbę L wg wzoru (33). Przykładowe wyniki pokazano na rys.26. Jakość skwantowanego sygnału rośnie skutkiem zwiększania wymiaru wektora. Ten wzrost jest jednak ograniczony, gdyż nie można zmniejszyć szumu kwantyzacji poniżej wartości opisanej funkcją RDF (rys.1). Przy zwiększaniu wymiaru wektora N szum kwantyzacji kwantyzatora wektorowego dąży do wartości RDF (pokazano to na rys.1). Oznacza to, że kwantyzator wektorowy jest asymptotycznie optymalnym koderem sygnału.

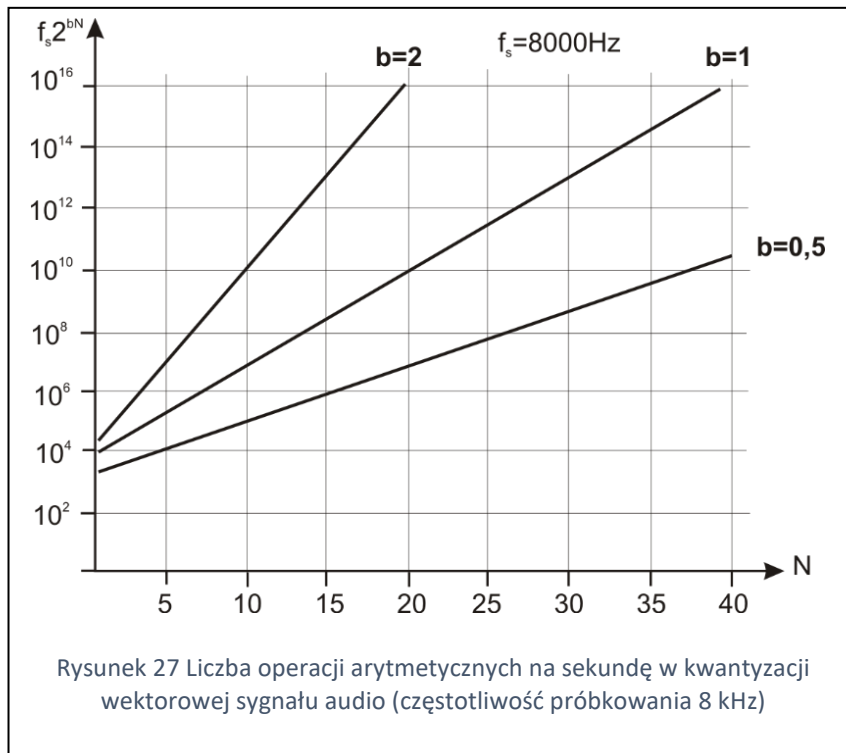


Przewaga kwantyzatora wektorowego nad kwantyzatorem skalarnym wynika z:

- - wykorzystania korelacji między kolejnymi N próbkami
- - wykorzystania właściwości wielowymiarowej gęstości prawdopodobieństwa próbek

Przy braku korelacji mgławica przybiera kształt okrągły, jednak i w tym przypadku można uzyskać niewielką poprawę SNR w porównaniu z kwantyzacją skalarną.

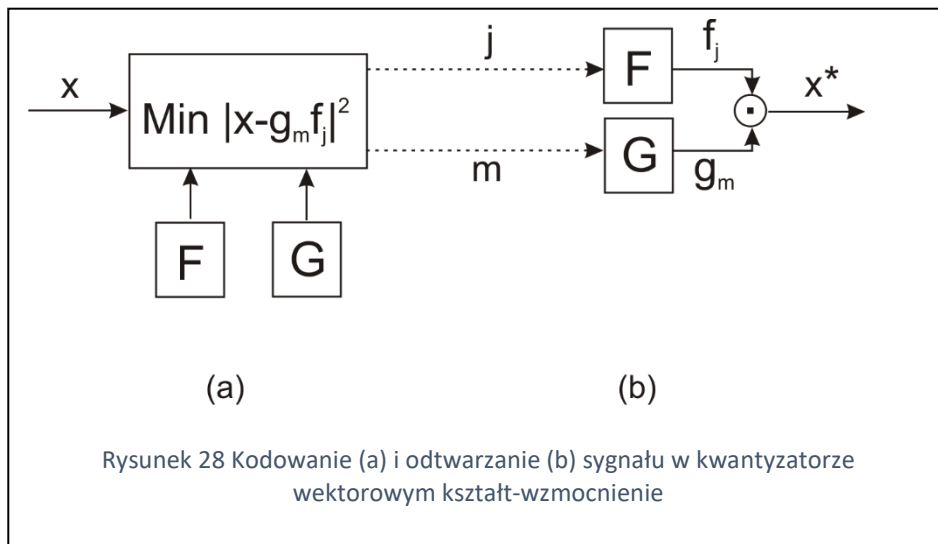
Poważnym problemem w implementacji kwantyzacji wektorowej jest złożoność obliczeniowa. Wynika to z eksponencjalnego wzrostu liczby wektorów słownika przy wzroście wymiaru wektorów (wzór 33). Każdy kwantowany wektor należy porównać ze wszystkimi wektorami słownika, co pochłania LN operacji arytmetycznych. W procesie kwantowania sygnału audio mamy $\frac{f_s}{N}$ wektorów na sekundę (f_s – częstotliwość próbkowania). W ciągu sekundy należy zatem wykonać $\frac{f_s}{N}LN = f_sL = f_s2^{bN}$ operacji. Już dla wektorów o wymiarze rzędu kilkunastu jest to liczba wykluczająca implementację w czasie rzeczywistym – rys.27.



Aby obniżyć złożoność obliczeniową, stosuje się m.in. **kwantyzator wektorowy typu kształt – wzmacnienie (SGVQ - shape - gain vector quantizer)**. Tu wektor próbek x jest reprezentowany jako iloczyn wektora kształtu i skalarnego wzmacnienia:

$$x^* = g_m f_j \quad (34)$$

W ten sposób ten sam słownik kształtów $F = \{f_j\}$, zawierający L_s wektorów, może być wykorzystany w procesie kwantyzacji sygnałów o różnej mocy. Schemat kodera i dekodera SGVQ pokazano na rys.28. Oprócz słownika kształtów wykorzystuje się słownik wzmacnień $G = \{g_m\}$, czyli zbiór L_g poziomów kwantyzacji dla wzmacnienia.



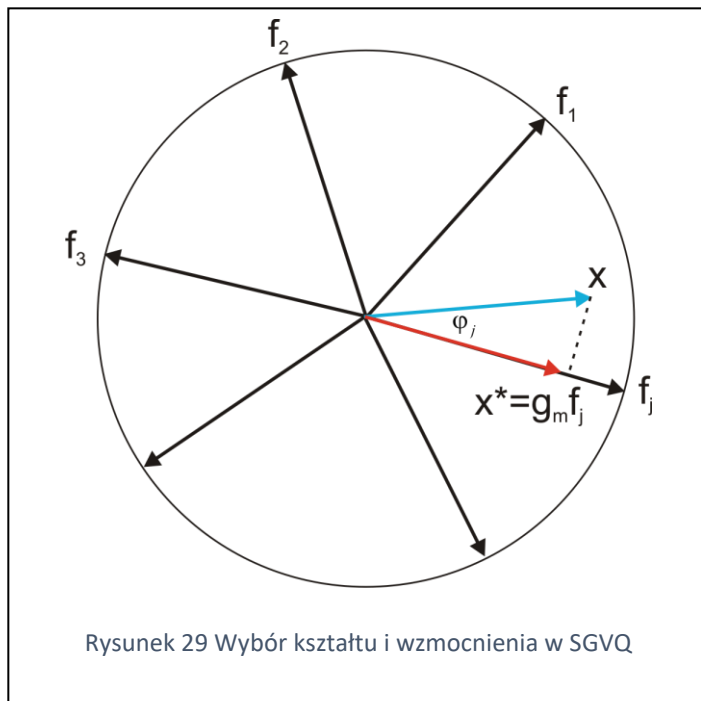
Wektor x^* można utworzyć na $L_s L_g$ sposobów, ale nie testujemy wszystkich kombinacji $g_m f_j$. Najczęściej w pierwszej kolejności wybiera się kształt (L_s porównań wektora x z wektorami kształtu ze słownika F), a potem, dla wybranego kształtu, szuka się wzmacnienia (L_g prób). Pokazano to na rys.29. Kształt wybieramy na minimum kąta, jaki tworzy on z kwantowanym wektorem x . Równoważnym podejściem jest szukanie maksimum kosinusa kąta:

$$\max_j \cos(\phi_j) = \max_j \frac{x^T f_j}{\|x\| \|f_j\|} \quad (35)$$

Następnie dla wybranego kształtu f_j szukamy optymalnego wzmocnienia g_{opt} tak, aby wektor $g_{opt} f_j$ był rzutem ortogonalnym x na f_j : $x - g_{opt} f_j \perp f_j$. Ortogonalność oznacza zerowanie się iloczynu skalarnego wektorów: $x^T f_j - g_{opt} f_j^T f_j = 0$. Ostatecznie otrzymujemy

$$g_{opt} = \frac{x^T f_j}{f_j^T f_j} \quad (36)$$

W końcu kwantujemy g_{opt} , wybierając ze słownika wzmocnień \mathbf{G} wartość g_m , najbliższą g_{opt}



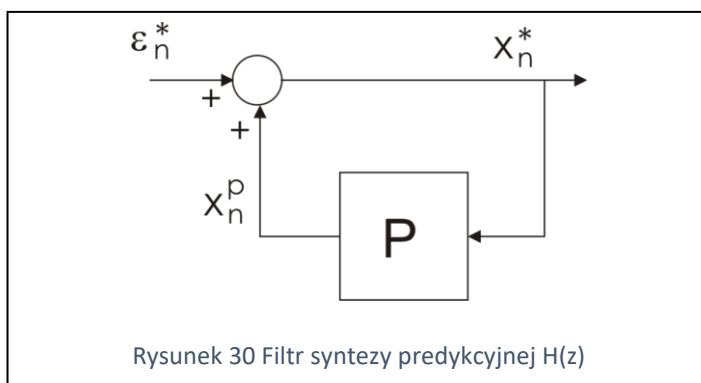
W Module 4 opisano symulator kwantyzatora wektorowego, w tym kwantyzatora kształt- wzmocnienie. Zainteresowany czytelnik może przekonać się, że przy tej samej przepływności SGVQ nie zapewnia tak dobrej jakości sygnału jak klasyczny kwantyzator wektorowy. Zaletą SGVQ jest znacznie mniejsza liczba operacji arytmetycznych, umożliwiającą implementację kodera w czasie rzeczywistym.

5 Kodery predykcyjno – wektorowe CELP i wokodery predykcyjne

Kodery predykcyjne wykorzystują filtr syntezy predykcyjnej (rys.30), opisany w p.3.5. Filtr ten ma transmitancję (wzory 31,32)

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 - \sum_{i=1}^p a_i z^{-i}} \quad (37)$$

gdzie a_1, a_2, \dots, a_p – współczynniki predykcji.



Wymienione poniżej kodery predykcyjne różnią się sposobem generowania sygnału pobudzającego ϵ_n^* :

- ADPCM - sygnał pobudzający kwantowany skalarnie
- CELP - sygnał pobudzający kwantowany wektorowo
- Wokodery - sygnał pobudzający generowany sztucznie.

Poza tym stosuje się różne algorytmy obliczania współczynników predykcji, opisujące filtr $H(z)$:

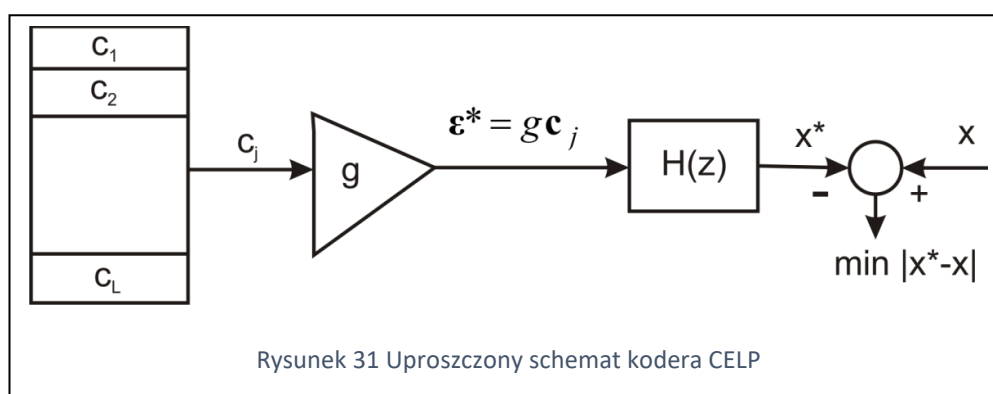
- ADPCM - współczynniki odtwarzane metodami sekwencyjnymi (nie transmituje się)
- CELP - transmisja współczynników predykcji co 10-20 ms (za wyjątkiem kodera LDCELP)
- Wokodery - zawsze transmituje się współczynniki filtru.

Można wymienić następujące zastosowania koderów predykcyjnych:

- ADPCM: - transmisja sygnału telefonicznego z szybkością 40, 32, 24, 16 kbit/s (standard telekom. ITU-T G.726 [5])
 - transmisja mowy o poszerzonym paśmie do 7 kHz (standard ITU-T G.722 [9])
- CELP: - transmisja mowy w sieciach telekom. z przepływnościami 4-16 kbit/s:
 - LDCELP: 16 kbit/s, standard G.728 [6]
 - VoIP, audiokonferencje: G.729 (8 kbit/s) [7],
 - G.723.1 (5.3/6.3 kbit/s) [8]
 - telefonia komórkowa (GSM-FR, GSM-HR, GSM-EFR [12], GSM-AMR [13]), również w USA (standardy IS54, IS96) i Japonii
- Wokodery: - łączność specjalna, np. cyfrowe radiostacje krótkofalowe (przepływność 800-2400 bit/s)

Kodery ADPCM opisano w p.3, obecnie zajmiemy się koderami predykcyjno- wektorowymi **CELP (Code Excited Linear Prediction)**, które mają największe zastosowanie.

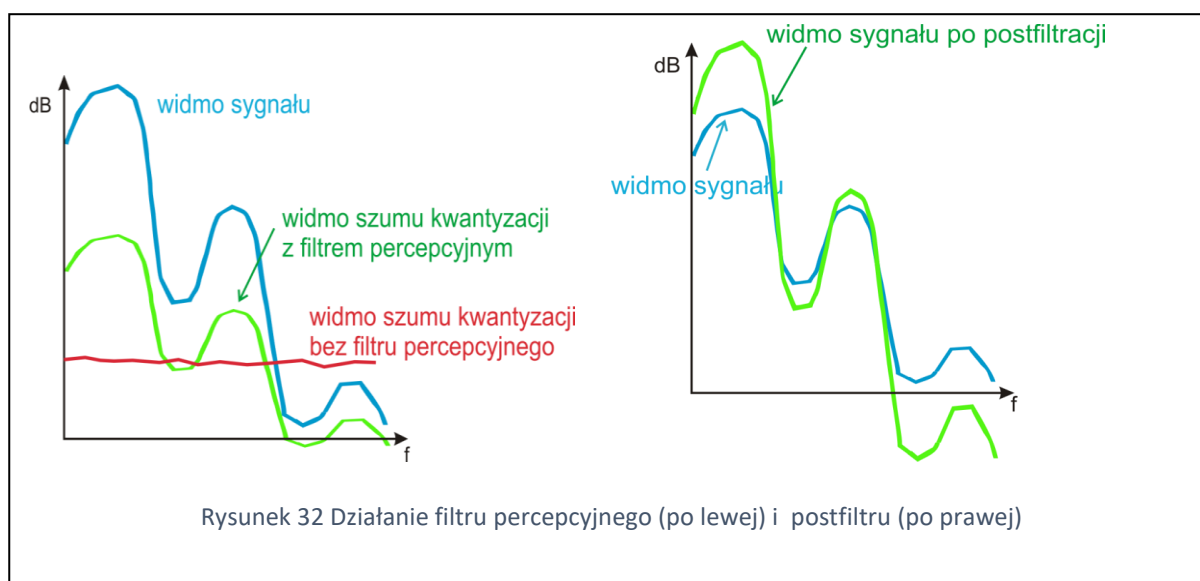
Uproszczony schemat kodera CELP pokazano na rys.31. Sygnał przetwarzany jest wektorowo, wektory składają się z N (od 5 do 60) sąsiednich próbek mowy. Koder jest właściwie kwantizatorem wektorowym kształt-wzmocnienie z dołączonym filtrem predykcyjnym $H(z)$. W słowniku kształtów jest L wektorów N -wymiarowych. W większości implementacji słownik nie jest przechowywany w pamięci, ale kolejne wektory są na bieżąco generowane wg ustalonego algorytmu (mówimy wtedy o słowniku algebraicznym). Po wzmacnieniu i odfiltrowaniu w filtrze predykcyjnym $H(z)$, otrzymany wektor x^* jest porównywany z wektorem sygnału mowy x . Wybiera się ten wektor ze słownika i to wzmacnienie, które zapewniają najmniejszą energię błędu $\|x^* - x\|^2$. Takie podejście nazywamy analizą przez syntezę, gdyż nadajnik jest w zasadzie wielokrotnie uruchamianym odbiornikiem (syntezatorem sygnału mowy).



Według schematu z rys.31 działa standardowy koder CELP o małym opóźnieniu (Low Delay CELP - LD-CELP G.728 [6]). Ma on następujące cechy:

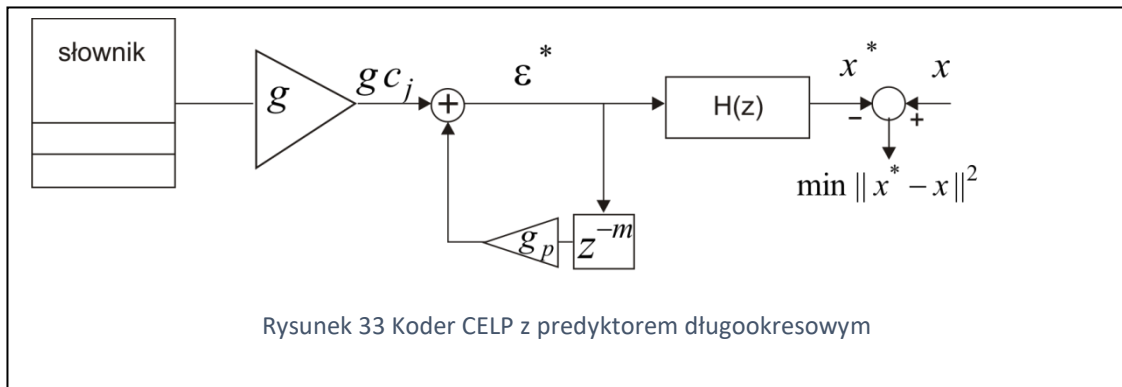
- Częstotliwość próbkowania: 8 kHz
- Operacje na wektorach $N=5$ -wymiarowych (stąd opóźnienie 0.625ms)
- Liczba wektorów w słowniku: $L=128$ (indeks wybranego wektora (j) przesyła się w 7 bitach)
- Liczba bitów na zakodowanie współczynnika wzmocnienia g : 3 (wzmocnienie kwantuje się z wykorzystaniem osobnego predyktora wzmocnienia, co zapewnia mały błąd kwantyzacji mimo tylko $2^3=8$ poziomów kwantyzacji)
- Przepływność binarna: 16 kbit/s: (7+3) bitów na wektor razy 8000/5 wektorów na sekundę)
- Obliczanie współczynników predykcji z przeszłego sygnału x^* , dostępnego po stronie nadawczej i odbiorczej. Współczynniki odtwarza się w odbiorniku bez ich transmisji.

Warto wspomnieć o tzw. filtrze percepcyjnym i postfiltrze, nie pokazanych na rys.31. Dążenie do zmniejszenia energii błędu kwantyzacji prowadzi do „wypłaszczenia” widma błędu (rys.32). Nie jest to korzystne, gdyż szum kwantyzacji o szerokim paśmie częstotliwości jest dość dobrze słyszalny. Zadaniem filtra percepcyjnego jest takie ukształtowanie widma szumu kwantyzacji, żeby go zamaskować widmem sygnału. Taki zamaskowany szum może mieć większą moc, ale będzie mniej dokuczliwy dla słuchacza niż szum niezamaskowany. Filtr percepcyjny może działać jak filtr preemfazy, tłumiący w pewnym stopniu rezonanse formantowe w sygnale mowy przed podaniem tego sygnału na wejście kodera, oraz filtr deemfazy, przywracający rezonanse formantowe na wyjściu dekodera i jednocześnie kształtujący widmo szumu kwantyzacji.

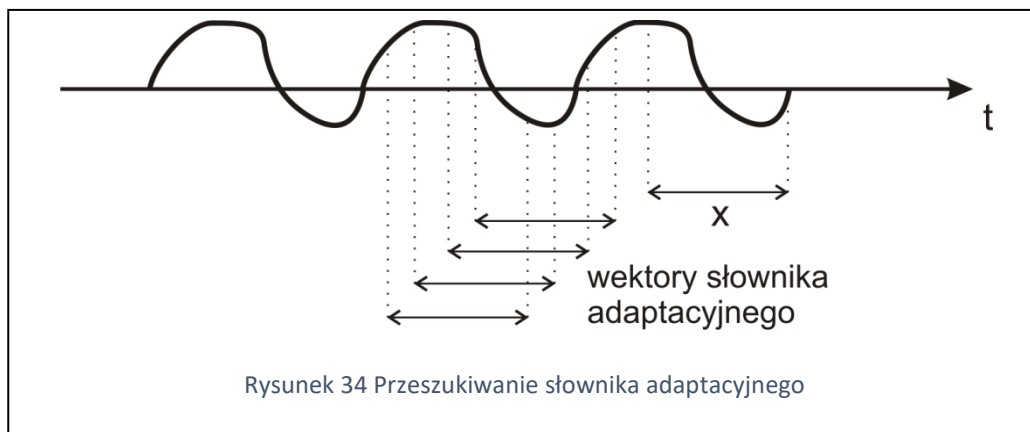


Filtr percepcyjny kształtuje widmo szumu kwantyzacji, nie zmienia widma sygnału. Z kolei postfiltr zmienia widmo sygnału na wyjściu odbiornika, prowadząc do „wyostrzenia” rezonansów formantowych i pogłębienia efektu maskowania szumu kwantyzacji.

Koder LD-CELP nie jest typowym koderem CELP. Typowy koder CELP przetwarza wektory o większym wymiarze (40-60) i uzyskuje współczynniki predykcji z oryginalnego sygnału mowy, co wymaga ich transmisji do odbiornika. Ponadto zawiera bardzo istotny element, tzw. predyktor tonu krtaniowego, zwany też predyktorem długookresowym (LTP- long term predictor). Pozwala on na lepsze odtworzenie quasiperiodycznej mowy dźwięcznej (rys.2, Moduł1). Schemat takiego kodera pokazano na rys.33. Predyktor długookresowy kopiuje sygnał pobudzający filtr $H(z)$ w poprzednim okresie tonu krtaniowego. Opóźnienie w pętli sprzężenia zwrotnego (m próbek) powinno być równe okresowi drgań strun głosowych mówcy. Wzmocnienie (g_p) pozwala na odtworzenie fazy narastania i wybrzmiewania dźwięku.

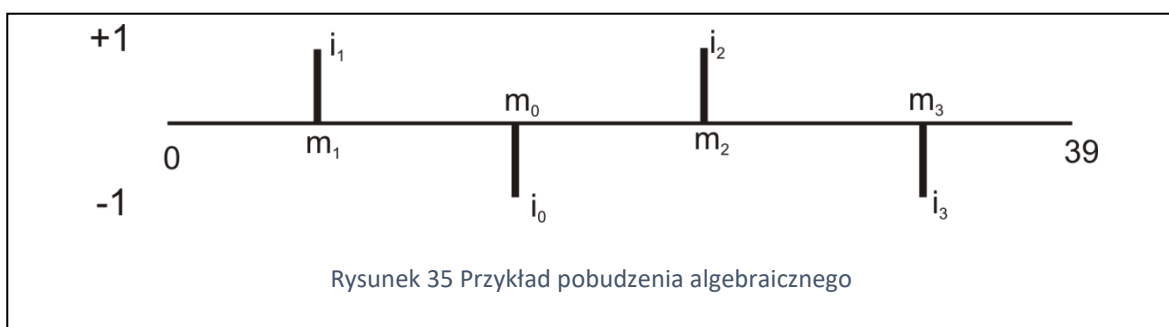


Pomiaru opóźnienia dokonuje się przez wycinanie z przeszłego sygnału pobudzającego ε^* wektorów o wymiarze N , aż do znalezienia wektora, który po odfiltrowaniu filtrem $H(z)$ zapewni najmniejszą odległość sygnału x^* od wektora próbek mowy x . Jest to równoważne przeszukiwaniu słownika zawierającego wektory wycięte z sygnału ε^* - zwanego słownikiem adaptacyjnym (rys.34).



Po znalezieniu opóźnienia m i wzmocnienia g_p , przechodzimy do szukania najlepszego wektora w słowniku stałym c_j i odpowiedniego wzmocnienia g . Jak już wspomniano, wektorów $\{c_j\}$ najczęściej nie przechowuje się w pamięci, tylko konstruuje na bieżąco, a traktuje ich przeszukiwanie. Taki wirtualny słownik nazywa się słownikiem algebraicznym. Jako przykład kodera z takim słownikiem weźmy koder ACELP (algebraic code excited linear prediction) G.729 [7]. Podstawowe parametry tego kodera są następujące:

- -filtr syntezy $H(z)$ oparty na 10 współczynnikach predykcji
- -obliczanie (i transmisja) parametrów filtru co 10 ms (80 próbek)
- - wymiar wektora sygnału pobudzającego: $N=40$ próbek
- - opóźnienie algorytmiczne 15 ms (120 próbek, potrzebnych do obliczenia współczynników predykcji)
- Pobudzenie ternarne (ze słownika stałego) zawierające 4 impulsy o amplitudach +1 lub -1 i położeniach m_0, m_1, m_2, m_3 – rys.35



40-wymiarowy wektor sygnału pobudzającego ε^* jest wektorem „rzadkim” (sparse) gdyż zawiera jedynie 4 składowe niezerowe. Jego konstruowanie polega na kolejnym rozmieszczaniu 4 impulsów. Każdy impuls ma 8 możliwych położeń (jedynie ostatni z nich ma 16 dopuszczalnych położeń). Dopuszczalne położenia są następujące:

$m_0=0, 5, 10, 15, 20, 25, 30, 35$

$m_1=1, 6, 11, 16, 21, 26, 31, 36$

$m_2=2, 7, 12, 17, 22, 27, 32, 37$

$m_3=3, 8, 13, 18, 23, 28, 33, 38,$

$4, 9, 14, 19, 24, 29, 34, 39$

Wektor pobudzenia algebraicznego jest kodowany w 17 bitach (położenia w 3+3+3+4 bitach amplitudy w 4 bitach). Cały wektor jest mnożony przez skwantowany współczynnik wzmocnienia g . W sumie transmisja parametrów niezbędnych do działania dekodera (pobudzenie algebraiczne, opóźnienie predyktora długookresowego, współczynniki wzmocnienia g i g_p , współczynniki predykcji) wymaga 8000 bit/s,

Podobny algorytm zastosowano w koderach G.723.1 [8], GSM -EFR (enhanced full rate [12]), GSM-AMR (adaptive multi-rate [13]) i wielu innych. Parametry kodera GSM-EFR są następujące:

- przepływność 12.2 kbit/s
- ramka podstawowa 20 ms (160 próbek)
- obliczanie predyktora co 10 ms
- wektory pobudzenia 40-wymiarowe (5 ms)
- pobudzenie ternarne:
- 10 impulsów, kod 35 -bitowy
- dopuszczalne położenia impulsów:

$m_0, m_5 = 0, 5, 10, 15, 20, 25, 30, 35$

$m_1, m_6 = 1, 6, 11, 16, 21, 26, 31, 36$

$m_2, m_7 = 2, 7, 12, 17, 22, 27, 32, 37$

$m_3, m_8 = 3, 8, 13, 18, 23, 28, 33, 38,$

$m_4, m_9 = 4, 9, 14, 19, 24, 29, 34, 39$

W Module 4 opisano ćwiczenie laboratoryjne „Kompresja sygnału mowy”. Zawiera ono symulator kodera GSM-AMR (Adaptive Multi-Rate) – nie w pełni kompatybilny ze standardem [13]. Koder ten ma kilka przepływności binarnych, podstawowym wariantem jest GSM-EFR (12.2 kbit/s). Zmniejszając liczbę impulsów w 40-wymiarowym wektorze pobudzenia, uzyskuje się mniejsze przepływności (Tabela2). Można obserwować działanie kodera z różnymi przepływnościami binarnymi, przebiegi sygnału pobudzającego, wejściowego, wyjściowego i ich widma, wpływ predykcji długookresowej na jakość sygnału (SNR).

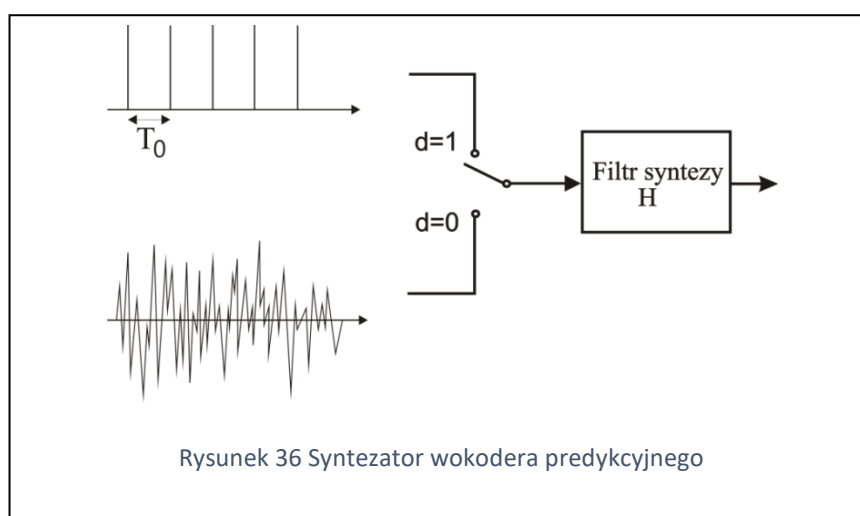
Tabela 2 Parametry kodera GSM-AMR

Przepływność [kbit/s]	liczba impulsów
12.2	10 *
10.2	8
7.4, 7.95	4 **
6.7	3
5.9	2

* Równoważny GSM EFR

** Położenia impulsów jak w G.729

Kodery CELP dominują w kompresji mowy w zakresie przepływności 4-16 kbit/s. Przy mniejszych przepływnościach są stosowane kodery parametryczne, zwane wokoderami. W koderach parametrycznych nie dąży się do uzyskania podobnego przebiegu czasowego mowy na wyjściu odbiornika i wejściu nadajnika. Chodzi o odtworzenie tych cech, które są niezbędne do prawidłowego zrozumienia mowy przez odbiorcę. Są to rezonanse formantowe (koncentracje energii na pewnych częstotliwościach, charakterystycznych dla poszczególnych głosek), dźwięczny lub bezdźwięczny charakter, częstotliwość drgania strun głosowych. Uproszczony schemat syntezy wokodera pokazano na rys. 36.

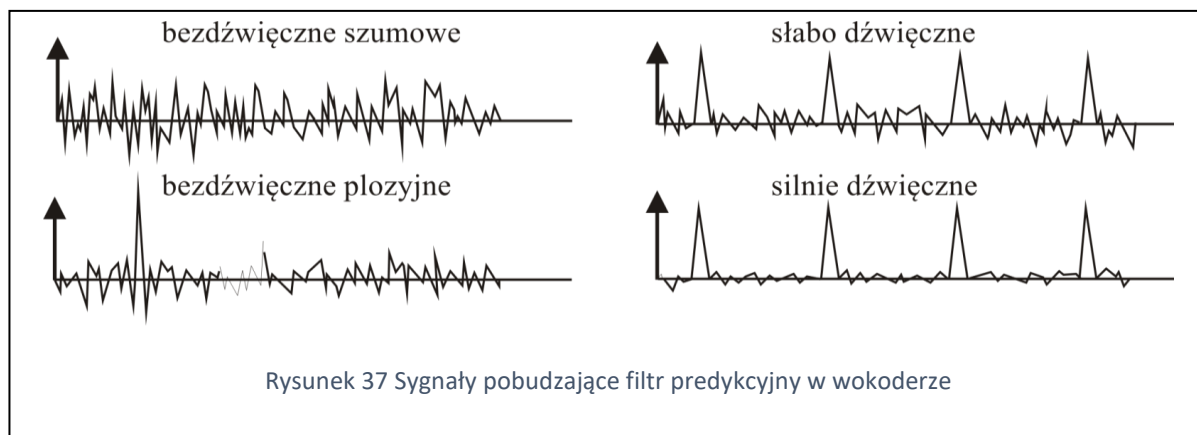


Filtr syntezy predykcyjnej $H(z)$ odtwarza częstotliwości formantowe, charakteryzujące poszczególne głoski. Sygnał pobudzający dla mowy bezdźwięcznej ($d=0$) pochodzi z generatora szumu, a dla mowy dźwięcznej ($d=1$) – z generatora okresowych impulsów. Okres repetycji impulsów (T_0) musi być równy okresowi tonu krtaniowego.

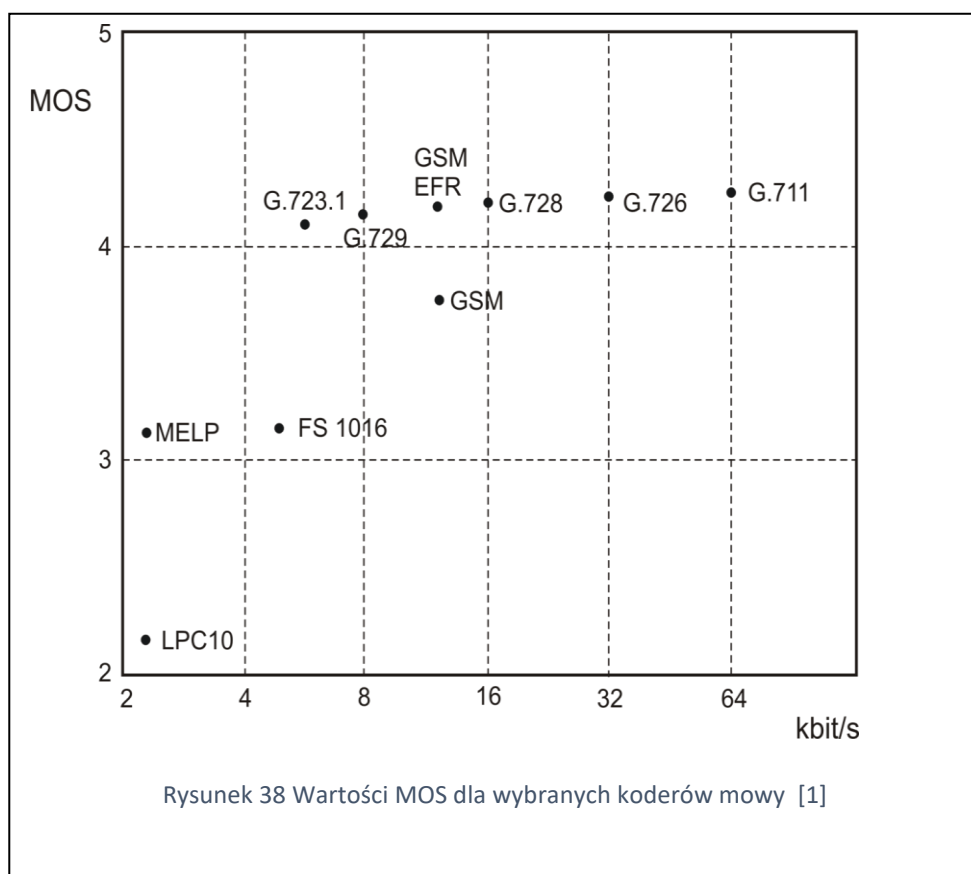
Przy przepływności 2400 bit/s mowa była zrozumiała, choć brakowało jej naturalności. Jakość można poprawić, wprowadzając więcej klas pobudzeń – rys. 37. Np. głoski plosyjne („p”, „b”) nie powinny być odtwarzane drogą przepuszczania szumu przez filtr $H(z)$, raczej potrzebne są tu aperiodyczne impulsy. Podobnie słabo dźwięczne głoski jak („ż”, „ź”) wymagają też szumu dla ich odpowiedniego brzmienia.

Pod względem jakości mowy kodery parametryczne nie dorównują jednak koderom predykcyjno-wektorowym CELP, z tego względu nie przyjęły się w telefonii powszechnego użytku. Dzięki niskiej przepływności znajdują

zastosowanie w warunkach dużych zakłóceń, gdy tylko powolna transmisja zapewni akceptowalną stopę błędów.



Porównując kodery mowy, należy zwrócić uwagę na jakość mowy, przepływność binarną, opóźnienie. Jakość można mierzyć wartością SNR, lepiej jest jednak przeprowadzić testy odsłuchowe. Najczęściej wynik testowania podaje się jako MOS – Mean Opinion Score. Jest to średnia ocena grupy słuchaczy, podawana w skali 1-5. Istnieją też standardy ITU, umożliwiające obliczenie przypuszczalnego wyniku testu odsłuchowego w oparciu o porównanie widm sygnału oryginalnego i testowanego [16]. Na rys.38 podano wyniki dla kilku koderów mowy o paśmie 4 kHz. Kodery zaaprobowane przez ITU (oznaczenie standardu G.XXX) z reguły osiągają MOS > 4. Standardy LPC10 i MELP to wokodery predykcyjne. Koderami mowy i muzyki o szerszym paśmie zajmiemy się w następnym punkcie.



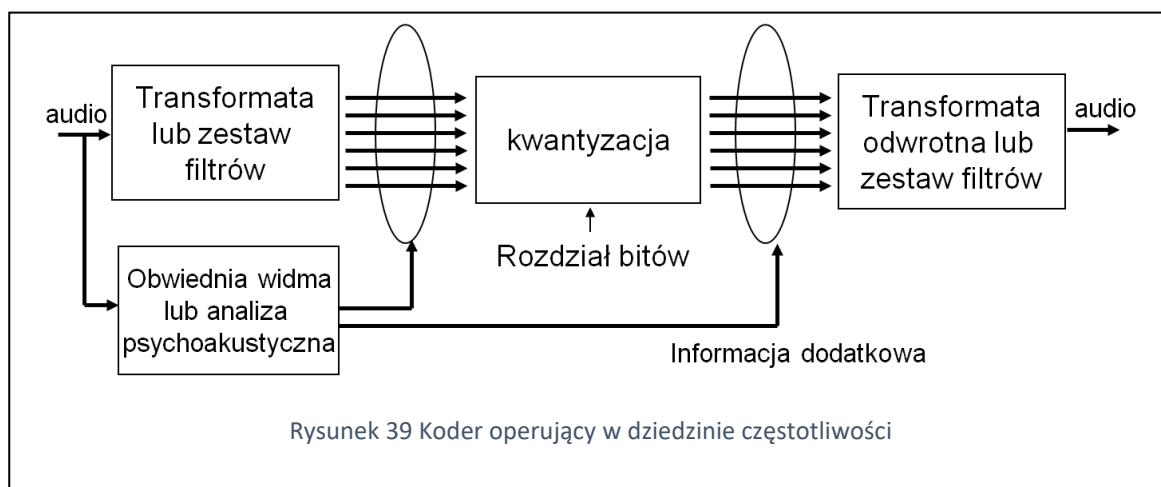
6 Kodowanie w dziedzinie częstotliwości

Kodowanie sygnałów audio w dziedzinie częstotliwości obejmuje dwa typy koderów: kodery subpasmowe i kodery transformaty. W obu przypadkach otrzymuje się reprezentację sygnału audio w podpasmach częstotliwości. W koderach subpasmowych są to próbki wąskopasmowych sygnałów, a w koderach transformaty – współczynniki transformaty (najczęściej DCT), odnoszące się do szeregu częstotliwości. Operacja przejścia na skalę częstotliwości jest odwracalna – sumując sygnały wąskopasmowe lub obliczając transformatę odwrotną uzyskujemy niezmienioną kopię sygnału oryginalnego. Kodowanie w dziedzinie częstotliwości jest jednak operacją stratną, gdyż sygnały pasmowe lub współczynniki transformaty podlegają kwantyzacji.

Celem kodowania w dziedzinie częstotliwości jest:

- zmniejszenie mocy szumu kwantyzacji
- odpowiednie ukształtowanie widma szumu kwantyzacji

Powyższe cele osiąga się, kwantując w różny sposób sygnały o różnych częstotliwościach, jak to pokazano na rys.39. Wymaga to wykorzystania wielu kwantyzatorów o różnych parametrach: zakresie pracy i liczbie poziomów kwantyzacji. Parametry kwantyzatorów wyznacza się w oparciu o obwiednię widma (sygnał pasmowy o większej mocy wymaga większego zakresu pracy kwantyzatora i większej liczby poziomów kwantowania). Niekiedy uwzględnia się zjawisko maskowania szumu kwantyzacji sygnałem audio (analiza psychoakustyczna). Po stronie odbiorczej sumuje się sygnały subpasmowe otrzymane z ich skwantowanych próbek (wymaga to zastosowania zestawu filtrów pasmowych i sumatora) albo oblicza się transformatę odwrotną ze skwantowanych współczynników

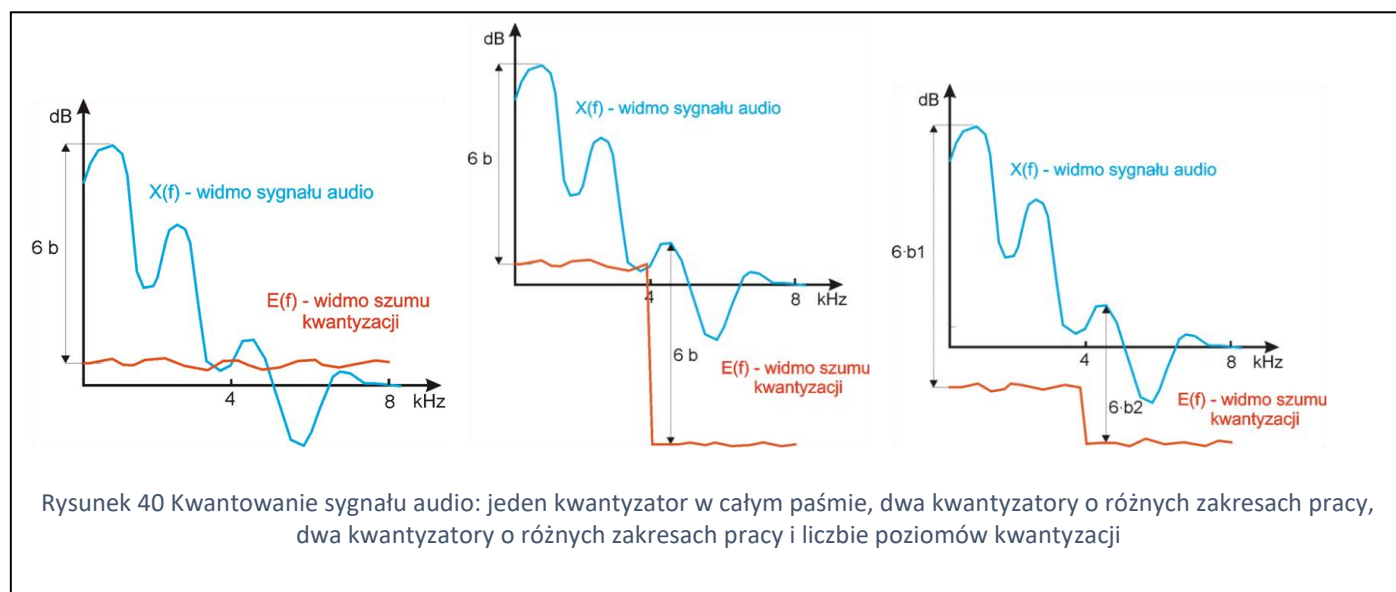


Dlaczego kwantyzacja w dziedzinie częstotliwości jest bardziej opłacalna niż kwantyzacja w dziedzinie czasu, jak w koderze PCM? Wyjaśniono to na rys.40. Załóżmy że mamy sygnał o paśmie 8 kHz, próbkowanie przebiega więc z szybkością 16000 próbek na sekundę. Kwantujemy próbki w dziedzinie czasu przeznaczając b bitów na próbkę (kwantyzator ma $L=2^b$ poziomów kwantyzacji). Zgodnie z zasadą „6 dB na bit” moc szumu kwantyzacji powinna się znaleźć $6b$ decybeli pod mocą sygnału (przyjmujemy, że główna część mocy znajduje się w okolicach maksimum widma). Szum kwantyzacji ma raczej płaskie widmo, więc w zakresie wyższych częstotliwości może dominować nad sygnałem audio i stać się dokuczliwym.

Można temu zaradzić, dzieląc sygnał na 2 sygnały pasmowe i kwantując każdy z nich osobnym kwantyzatorem. Sygnał w górnym paśmie ma o wiele mniejszą amplitudę, a więc można „zagęścić” siatkę poziomów kwantyzacji i zmniejszyć moc błędu kwantowania, do wartości o 6b decybeli niższej niż moc sygnału w górnym paśmie. Zauważmy, że mimo stosowania dwóch kwantyzatorów liczba próbek na sekundę nie zwiększy się, gdyż częstotliwość próbkowania sygnałów pasmowych wynosi 8 kHz (2 razy szerokość pasma).

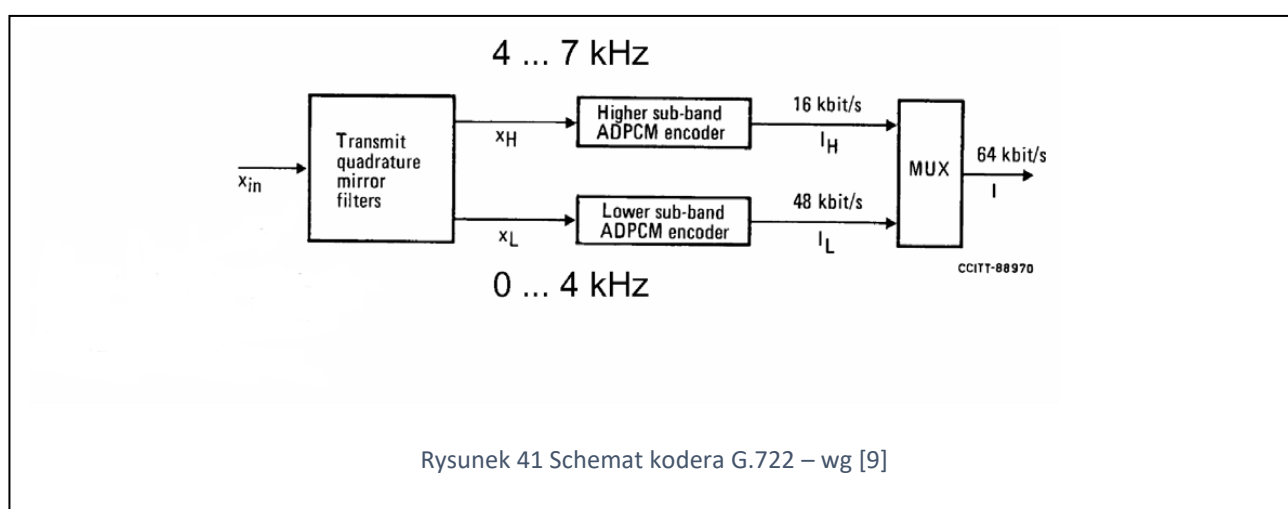
Jednocześnie można zmniejszyć liczbę bitów na próbkę w górnym paśmie i zwiększyć w dolnym paśmie, zachowując tę samą szybkość transmisji $8000 b_1 + 8000 b_2 = 16000 b$ bit/s. Wówczas moc szumu

kwantyzacji w obu podpasmach może się wyrównać, ale na niższym poziomie, niż to zapewniał pojedynczy kwantyzator działający w całym paśmie częstotliwości.



Przykładem koderu 2-pasmowego jest standard G.722 [9]. Kodery z serii G.722 [9], [10], [11] służą do kompresji sygnału mowy i innych sygnałów akustycznych w paśmie 0-7 kHz. Obecnie w telefonii odchodzi się od sygnału mowy o paśmie zawężonym do niespełna 4 kHz, ze względu na niezadowalającą jakość takiego sygnału. W tych koderach stosuje się częstotliwość próbkowania 16 kHz, co umożliwia przetwarzanie sygnałów w paśmie 8 kHz, jednak pasmo 7-8 kHz jest wytłumione.

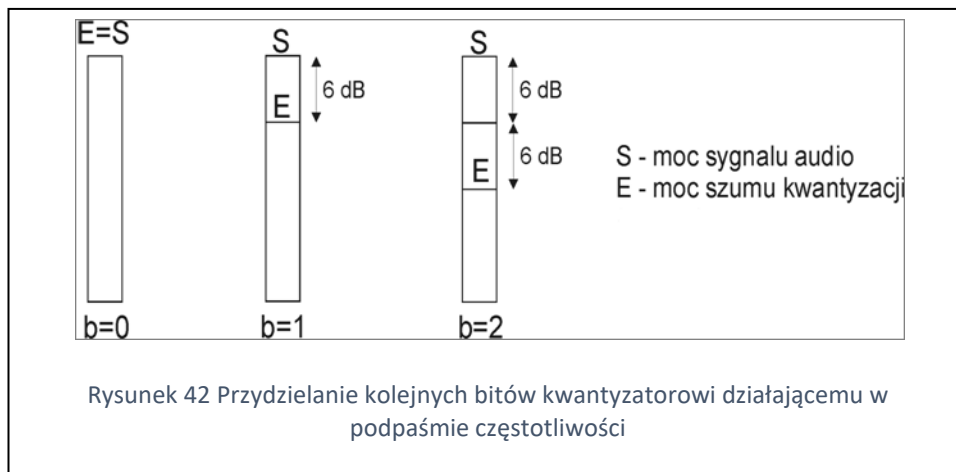
W koderze G.722 stosuje się, w każdym paśmie, algorytm kompresji ADPCM (rys.41). W kanale dolnym jest to ADPCM z kwantyzatorem adaptacyjnym 6-bitowym, a w kanale górnym ADPCM z kwantyzatorem adaptacyjnym 2-bitowym. Uwzględniając 8000 próbek na sekundę w paśmie dolnym i górnym, mamy przepływność binarną $6 \times 8000 + 2 \times 8000 = 64000$ bit/s. Jest ona równa przepływności koderu PCM – G.711 [4], jednak standard G.722 oferuje dwa razy szersze pasmo mowy.



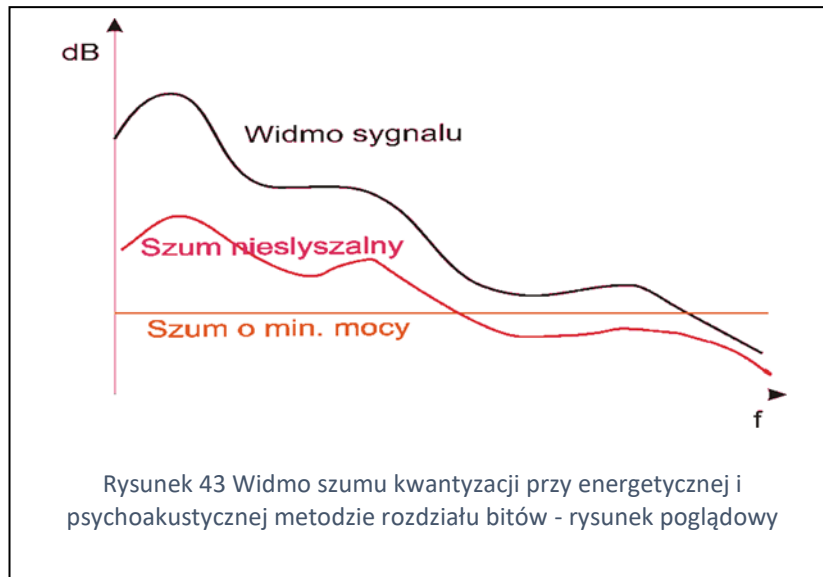
W koderze G.722 rozdział bitów pomiędzy kwantyzatory jest stały, w większości koderów jest on jednak zmienny, zależny od widma przetwarzanego fragmentu sygnału. Stosuje się dwie grupy algorytmów rozdziału (allokacji) bitów:

- Zasada „energetyczna”: minimum mocy szumu kwantyzacji
- Zasada „psychoakustyczna”: największa jakość sygnału akustycznego

Bity przydzielane są kolejno – przydzielenie 1 bitu obniża szum kwantyzacji o 6 dB. Kolejny bit jest przydzielany temu podpasmu, gdzie błąd kwantyzacji jest największy. Przed przydzieleniem bitów jako błąd traktujemy moc sygnału w danym paśmie, gdyż nie kodując sygnału (zastępując go ciszą) tracimy całą jego moc – rys.42..



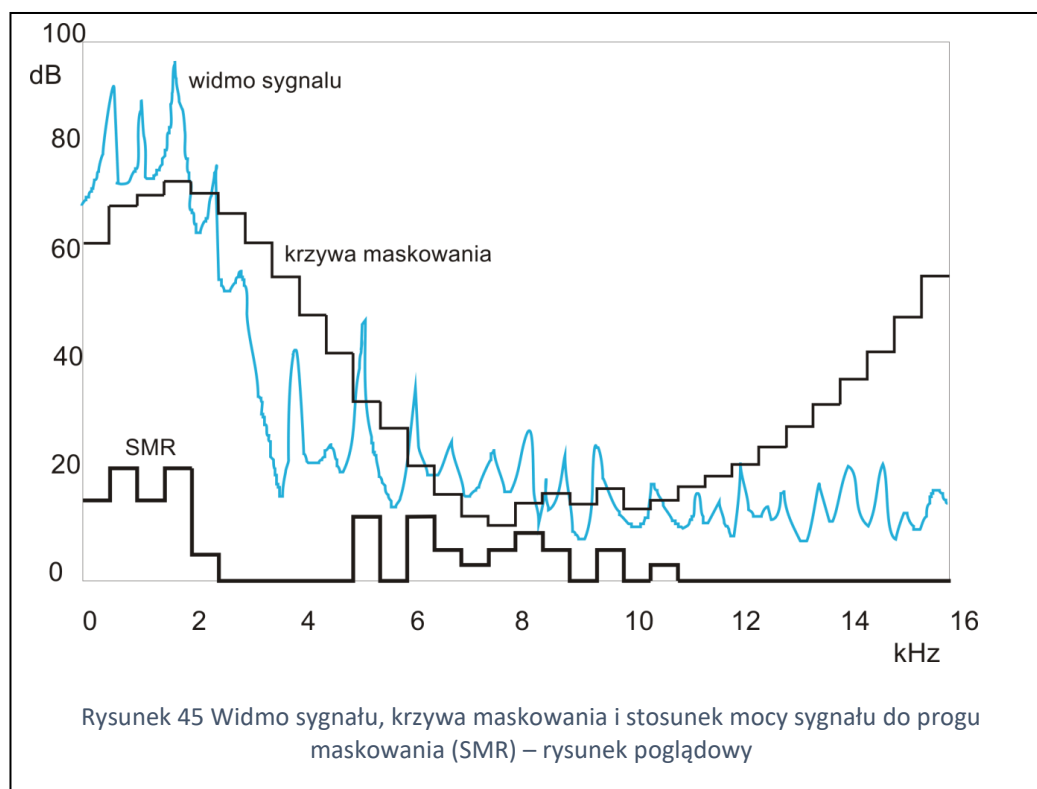
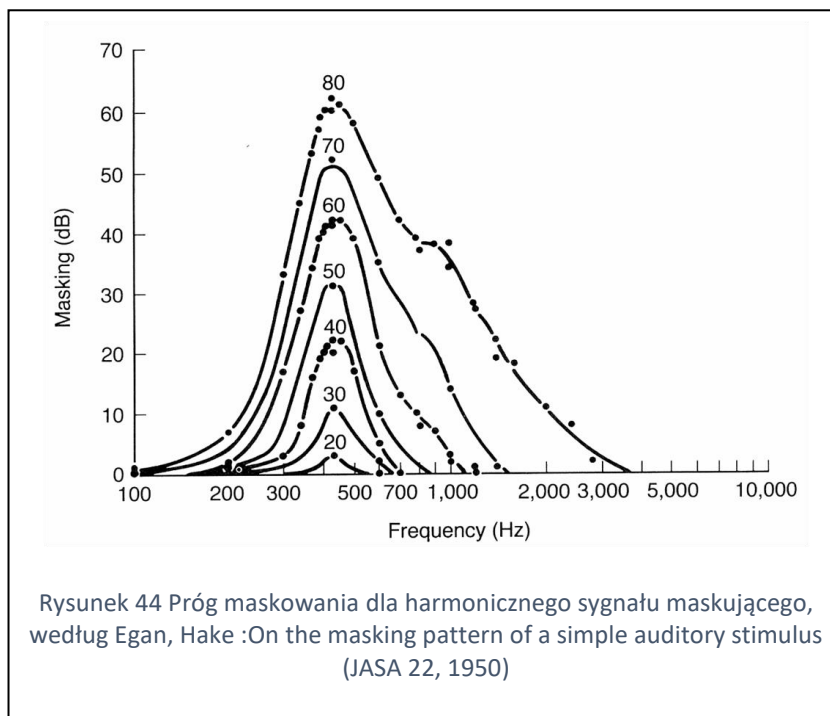
Przydzielając bity tam, gdzie szum kwantyzacji ma największą moc, doprowadzamy do spłaszczenia widma szumu kwantyzacji. Nie jest to korzystne dla brzmienia sygnału, gdyż szum kwantyzacji nie jest maskowany sygnałem audio – rys.43.



Aby zastosować metodę psychoakustyczną alokacji bitów, należy obliczyć **próg maskowania** (krzywą maskowania). Jest to próg słyszalności szumu kwantyzacji (i innych sygnałów) w obecności sygnału audio. Krzywa maskowania jest funkcją częstotliwości i zmienia się wraz ze zmianami widma sygnału audio. Gdy sygnał maskujący jest pojedynczym tonem, krzywa maskowania przebiega tak, jak to pokazano na rys. 44. Moc sygnału maskującego (od 20 dB do 80 dB) podano powyżej krzywej maskowania. Kwantowany sygnał audio składa się z wielu prążków widmowych, więc na krzywą maskowania składa się wiele funkcji pokazanych na rys.44.

Mając próg maskowania w każdym podpasmie, można sprawdzić, czy leży on powyżej czy poniżej widma sygnału w tym podpasmie. Jeśli leży powyżej, znaczy to, że sygnał w tym podpasmie nie jest słyszalny i nie trzeba go kwantować. Jeśli leży poniżej, wówczas możemy obliczyć stosunek mocy sygnału do progu maskowania (SMR – Signal to Mask Ratio). Obliczenia przeprowadzamy w decybelach, odejmując próg maskowania od mocy sygnału w danym podpasmie. Gdy np. $SMR=12$ dB, wówczas przydzielenie 2 bitów powinno uczynić szum kwantyzacji niesłyszalnym (zgodnie z zasadą 6 dB na bit). Najczęściej rozdzielamy bity pojedynczo, kanałowi o największej wartości SMR, po przydzieleniu bitu zmniejszając wartość SMR o 6 dB.

Jeśli mamy wystarczającą liczbę bitów do rozdzielania, to szum kwantyzacji jesteśmy w stanie całkowicie zamaskować.

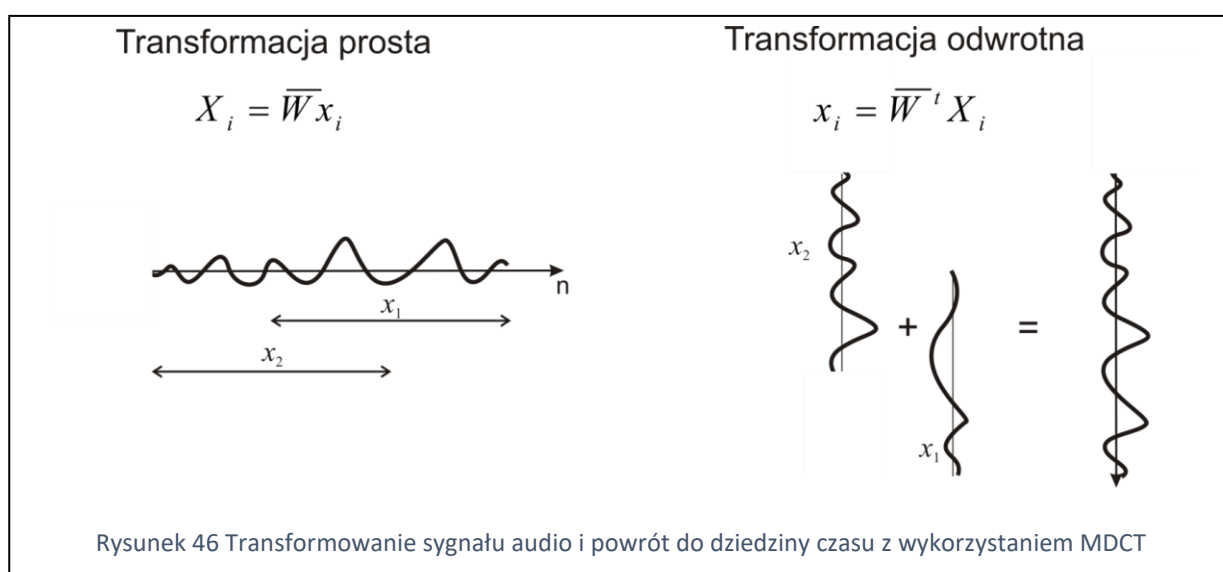


Przykładem kodera wykorzystującego „energetyczną” metodę alokacji bitów jest koder transformaty G.722.1 [10]. Podstawowe parametry tego kodera są następujące:

- Częstotliwość próbkowania 16 kHz
- Pasmo sygnału akustycznego 7 kHz
- Transformata MLT (MDCT), $N=320$ współczynników transformaty
- 14 podpasem po 20 próbek transformaty (szerokość podpasma = 0.5 kHz) - podpasem 7-7.5 kHz i 7.5-8 kHz nie koduje się
- Przepływności 24 i 32 kbit/s

- „energetyczny” algorytm rozdziału bitów
 - Wykorzystanie kwantyzacji wektorowej do kwantowania współczynników transformaty w podpasmach
- Przy przepływności 32 kbit/s jakość sygnału jest porównywalna z koderem G.722 o przepływności 64 kbit/s.

Wyjaśnienia wymaga zastosowana transformata. Należy ona do tzw. MLT (modulated lapped transforms), zwana jest też MDCT (zmodyfikowana dyskretna transformata kosinusoidalna). Podobnie jak DCT, jest ona opisana macierzą, ale nie kwadratową tylko prostokątną: $\bar{W}_{N \times 2N}$. Oznacza to, że N współczynników transformaty powstaje z $2N$ próbek sygnału audio. Wydawałoby się, że taka operacja nie może być odwracalna, gdyż znika połowa danych. Tak jednak nie jest, gdyż okno czasowe przesuwają się o pół swojej długości – kolejny wektor zawiera 50% próbek należących do poprzedniego wektora. Na każde N nowych próbek sygnału wejściowego mamy więc N próbek transformaty, gdyż okna nakładają się. Z kolei po stronie odbiorczej z N skwantowanych współczynników transformaty otrzymujemy $2N$ próbek sygnału. Otrzymane wektory dodajemy z 50% pokryciem, co zapewnia powrót do sygnału wejściowego z dokładnością do błędu kwantyzacji. Operacje transformacji prostej i odwrotnej pokazano na rys.46.



Przykładem koderów sygnału audio wykorzystujących „psychoakustyczną” metodę alokacji bitów są kodery MPEG-Audio [14], [15]. W najprostszej wersji jest to koder subpasmowy (rys.39) o następujących parametrach:

- częstotliwość próbkowania $f_s = 44100$ Hz
- 32 kanały od $f=0$ do $f=f_s/2$
- 32 kwantyzatory z adaptacją „w przód”
- dynamiczny rozdział bitów pomiędzy kwantyzatory
- dobra jakość stereofonicznego sygnału audio przy przepływności binarnej 160 kbit/s

Bardziej złożony wariant MPEG-Audio, powszechnie stosowany i znany jako MP3 [15], zawiera zarówno filtry pasmowe jak i transformaty MDCT. Każdy z 32 sygnałów pasmowych jest rozdzielany na 18 podpasm z wykorzystaniem transformaty. Otrzymujemy w ten sposób $32 \times 18 = 576$ podpasm, co daje większe możliwości alokacji bitów. Dobrą jakość sygnału stereofonicznego otrzymujemy przy przepływności binarnej 128 kbit/s. Zainteresowanym polecam symulator koder transformaty opisany w Module 4 (Kodowanie transformaty sygnału audio). Wykorzystywana jest transformata DCT. Otrzymujemy 512 próbek transformaty, które można podzielić na podpasma. W każdym podpaśmie działa inny kwantyzator skalarny. Możemy narzucić mu określoną liczbę bitów na próbkę lub uruchomić algorytm automatycznego rozdziału bitów pomiędzy kwantyzatory. Możliwe jest również zasymulowanie „psychoakustycznej” alokacji bitów. Można wówczas obserwować krzywe maskowania dla kolejnych fragmentów sygnału audio.

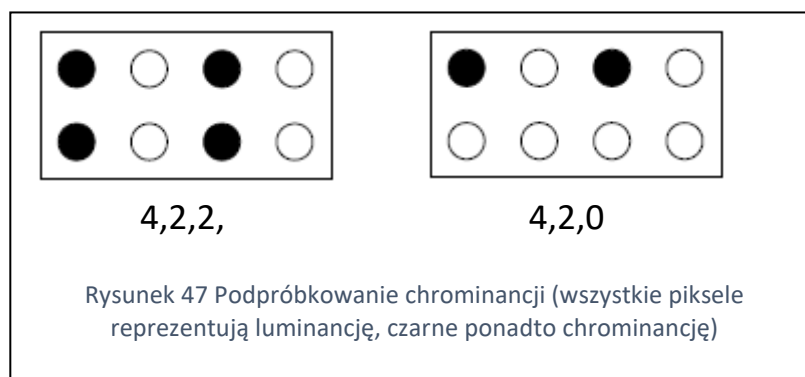
7 Kodowanie obrazu nieruchomego i sekwencji wideo

7.1 Reprezentacje obrazu nieruchomego i ruchomego

Obraz jest przetwarzany na szereg pikseli, podobnie jak dźwięk na szereg próbek. To próbkowanie obrazu jest możliwe dzięki ograniczonej rozdzielczości ludzkiego oka. Jesteśmy w stanie rozróżnić kilkadziesiąt punktów w jednym stopniu kątowym naszego pola widzenia, jeśli liczba szczegółów jest większa, postrzegamy je jako jednolitą plamę. Nasza wrażliwość na zmianę jasności jest większa niż wrażliwość na zmianę koloru. Piksele reprezentujące kolor (chrominancję) mogą zatem występować w większej odległości niż piksele reprezentujące jasność (luminancję). Aby z tego skorzystać, trzeba wpiery oddzielić luminancję od chrominancji. Załóżmy, że obraz został zapisany w postaci trzech zbiorów pikseli dla 3 podstawowych kolorów: R,G,B. Jasność poszczególnych pikseli niech będzie zapisana w postaci liczb całkowitych od 0 do 255. Taka 8-bitowa reprezentacja jest wystarczająca dla przeważającej części obrazów. Wówczas możemy zastosować odwracalne przekształcenie liniowe 3 kolorów na wartość luminancji (Y') i dwóch składowych chrominancji (Cb, Cr) – wzór (38). Jest wiele podobnych przekształceń, zainteresowanym polecam książki [17], [18].

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \quad (38)$$

Piksele chrominancji występują w zapisanym cyfrowo obrazie rzadziej niż piksele luminancji. Na rys. 47 pokazano dwie metody podpróbkowania chrominancji: schemat 4,2,2 (na 4 piksele luminancji przypadają 2 pary pikseli chrominancji) i schemat 4,2,0 (na 4 piksele luminancji mamy 2 pary pikseli chrominancji lub brak pikseli chrominancji).



Obraz ruchomy może zostać zapisany jako ciąg obrazów nieruchomych (tzw. ramek). Jest to możliwe dzięki ograniczonej rozdzielczości czasowej naszego oka: Widząc kilkadziesiąt ramek na sekundę wydaje się nam, że oglądamy płynny, ciągły ruch

7.2 Kwantyzacja skalarna i wektorowa obrazu

Kwantowanie pikseli obrazu rządzi się podobnymi prawami, co kwantowanie próbek sygnału audio. Można stosować kwantyzatory równomierne o różnej liczbie poziomów kwantyzacji, można optymalizować poziomy w celu zmniejszenia energii błędu kwantyzacji. Obowiązuje zasada „6 decybeli na bit”: dodanie jednego bitu kwantyzatorowi powoduje czterokrotne zmniejszenie energii błędu kwantyzacji (czyli zmniejszenie o 6 dB).

Do oceny jakości skwantowanego obrazu nie stosuje się jednak globalnej ani segmentowej wartości SNR, jak w sygnałach akustycznych. Wartości pikseli bliskie zera opisują ciemne fragmenty obrazu, a bliskie wartości maksymalnej – jasne. Błąd kwantyzacji pikseli jest tak samo dokuczliwy w ciemnych i jasnych elementach obrazu, więc wartości pikseli obrazu nie powinny wpływać na ocenę jego jakości. Dobrym wskaźnikiem jakości

byłaby energia błędu, zależy ona jednak od skali wartości pikseli (najczęściej są one wyskalowane od 0 do 1, albo od 0 do 255). Aby uniezależnić się od skali, wprowadzono normalizację mocy błędu kwantyzacji z wykorzystaniem mocy piksela o największej możliwej jasności (np. 1 lub 255²). Najczęściej podaje się Peak Signal to Noise Ratio (PSNR):

$$PSNR = \frac{x_{max}^2}{E(x_i - x_i^*)^2} \quad (39)$$

gdzie $E(x_i - x_i^*)^2 \cong \frac{1}{M} \sum_{i=1}^M (x_i - x_i^*)^2$ – średnia moc błędu kwantyzacji. PSNR najczęściej podaje się w decybelach.

W kwantowaniu obrazu nie znajduje zastosowania kwantyzator pseudologarytmiczny ani adaptacyjny. Ich celem było wyrównanie SNR dla sygnałów cichych i głośniejszych. W kwantyzacji obrazu odpowiednikiem cichego sygnału audio byłby ciemny obraz o znikomym kontraście – a więc bezużyteczny. Sensowne jest za to optymalizowanie poziomów kwantyzacji w celu zmniejszenia błędu kwantyzacji. Do obliczania poziomów kwantyzacji stosuje się algorytm Lloyd'a, o którym była mowa w p.4. Wyniki kwantowania skalarne obrazu w odcieniach szarości pokazano w tabeli 3. Liczba poziomów kwantyzacji wynosi $L = 2^b$, gdzie b – liczba bitów na piksel. Zauważmy, że zasada „6 dB na bit” jest spełniona jedynie w pewnym zakresie zmian b . Wyprowadziliśmy tę zasadę zakładając równomierny rozkład prawdopodobieństwa kwantowanych wartości, co nie jest spełnione dla pikseli obrazu. Ponadto metoda Lloyd'a nie gwarantuje otrzymania optymalnych poziomów kwantyzacji.

Tabela 3 Kwantowanie skalarne pikseli obrazu		
Liczba poziomów kwantyz.	b bitów na piksel	PSNR [dB]
oryginał * (L=256)	8	-
L=2	1	17.3
L=4	2	18.3
L=8	3	24.7
L=16	4	30.5
L=32	5	32.5

*Otwórz hyperlink aby zobaczyć obraz

Sąsiadujące ze sobą piksele są silnie skorelowane, często należą do rozległego obiektu o tym samym kolorze. Jeżeli blisko siebie leżące piksele połączymy w wektory, to można się spodziewać poprawy jakości przy stosowaniu kwantyzacji wektorowej. W tabeli 4 porównano wyniki kwantyzacji obrazu z rozdzielczością $b=1$ bit/piksel przy stosowaniu kwantyzacji skalarnej (L=2 odcienie szarości) i wektorowej. Utworzono 4-wymiarowe wektory z pikseli leżących w kwadracie 2x2 piksele, a następnie 9-wymiarowe wektory leżące w kwadracie 3x3 piksele. Liczba wektorów w słowniku wynosi $L=2^{bN}$ (wzór 33). Kwantyzacja wektorowa dała tu wyjątkowo dobre wyniki, zwłaszcza dla 9-wymiarowych wektorów. Porównując obrazy, pamiętajmy, że w każdym wypadku przeznaczamy 1 bit na zakodowanie 1 piksela.

Tabela 4 Kwantowanie wektorowe obrazu, $b=1$ bit/piksel		
Wymiar wektora	Liczba wektorów w słowniku	PSNR [dB]
N=1 (skalarne) *	L=2 poziomy kwantyzacji	17.3
N=4 (kwadraty 2 x 2)	L=16	26
N=9 (kwadraty 3 x 3)	L=512	31

*Otwórz hyperlink aby zobaczyć obraz

7.3 Transformaty dwuwymiarowe

W kodowaniu sygnałów audio najczęściej stosowana jest Dyskretna Transformata Kosinusoidalna DCT (lub jej zmodyfikowana wersja MDCT). Podobnie w kodowaniu obrazów dominuje dwuwymiarowa transformata DCT. Opisana jest ona tą samą macierzą \mathbf{W} , co transformata jednowymiarowa (p.5.2, Moduł1), jednak stosowana jest ona dwukrotnie: do kolumn i wierszy pikseli obrazu \mathbf{X} .

$$\mathbf{Y} = \mathbf{W}\mathbf{X}\mathbf{W}^t \quad (40)$$

gdzie \mathbf{X} – obraz zawierający $N \times N$ pikseli, \mathbf{Y} – transformata DCT tego obrazu. Widmo \mathbf{Y} też jest obrazem zawierającym $N \times N$ pikseli, które obliczamy wg wzoru (40). Po rozpisaniu na elementy macierzy otrzymujemy następujący wzór na współczynnik widma DCT, leżący w wierszu u i kolumnie v :

$$Y(u, v) = \alpha(u)\alpha(v) \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} X(i, j) \cdot \cos \left[\frac{(2i+1)u\pi}{2N} \right] \cdot \cos \left[\frac{(2j+1)v\pi}{2N} \right] \quad (41)$$

gdzie indeksy u, v to liczby całkowite o wartościach od 0 do $N-1$, natomiast współczynniki α są równe

$$\alpha(u) = \sqrt{1/N} \quad \text{dla } u = 0$$

$$\alpha(u) = \sqrt{2/N} \quad \text{dla } u = 1, 2, \dots, N-1$$

Dwuwymiarowa transformacja DCT jest odwracalna, mając widmo \mathbf{Y} możemy wrócić do obrazu oryginalnego stosując wzór

$$\mathbf{X} = \mathbf{W}^t \mathbf{Y} \mathbf{W} \quad (42)$$

Powyższe równanie opisuje odwrotną transformatę kosinusoidalną (IDCT). Po rozpisaniu na elementy macierzy otrzymujemy następujący wzór na piksel obrazu, leżący w wierszu i i kolumnie j :

$$X(i, j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) Y(u, v) \cdot \cos \left[\frac{(2i+1)u\pi}{2N} \right] \cdot \cos \left[\frac{(2j+1)v\pi}{2N} \right] \quad (43)$$

Ze wzoru (43) wynika, że obraz \mathbf{X} jest sumą $N \times N$ obrazów elementarnych. Każdy obraz elementarny jest zbiorem pikseli o współrzędnych (i, j) , ma ten sam wymiar co obraz \mathbf{X} i jest określony parą liczb u, v :

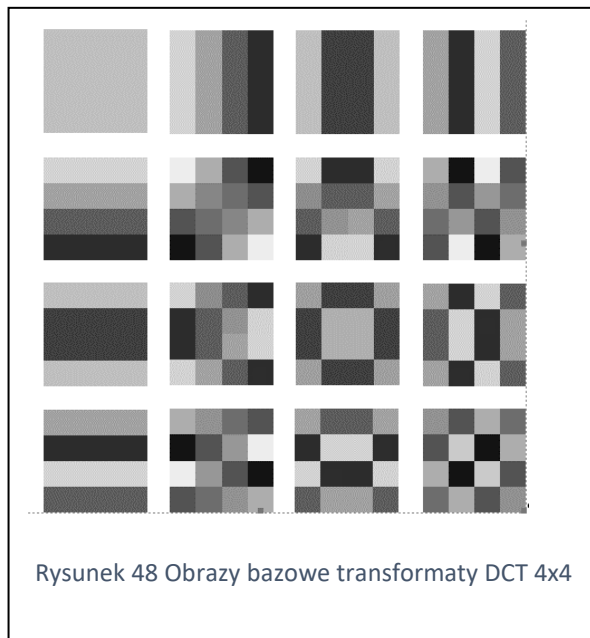
$$\tilde{Y}(u, v) = \alpha(u)\alpha(v) \cos \left[\frac{(2i+1)u\pi}{2N} \right] \cdot \cos \left[\frac{(2j+1)v\pi}{2N} \right] \quad (44)$$

Np. wszystkie piksele obrazu $\tilde{Y}(0,0)$ mają wartość $\alpha(0)^2 = \frac{1}{N}$. Obraz ten jest powierzchnią o jednolitym kolorze. Na rys.48 pokazano obrazy elementarne (zwane obrazami bazowymi) dla transformaty DCT o wymiarach 4×4 .

Wzór (43) można przepisać jako sumę ważoną obrazów bazowych:

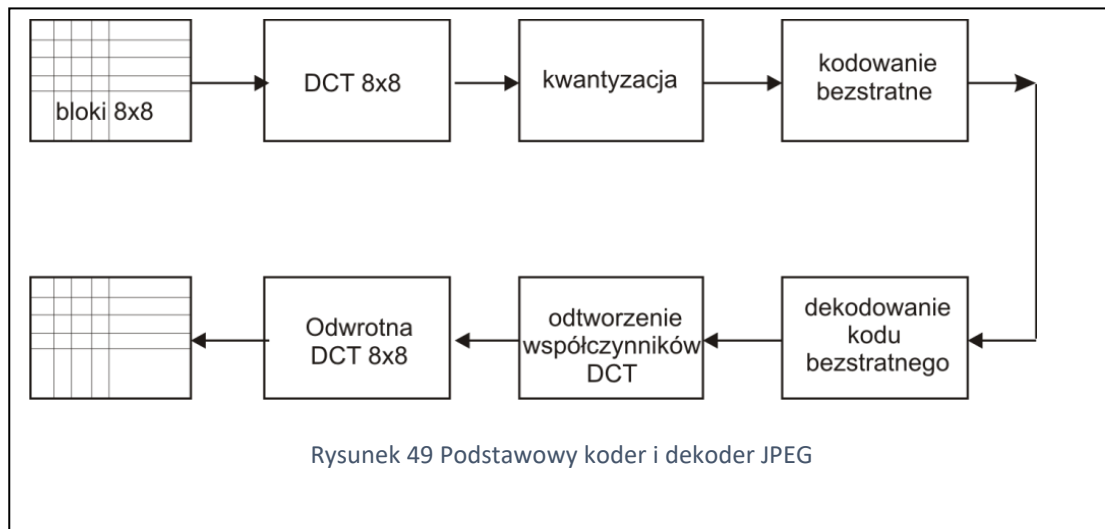
$$\mathbf{X} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} Y(u, v) \cdot \tilde{Y}(u, v) \quad (45)$$

Taj więc współczynnik DCT $Y(u, v)$ określa zawartość obrazu bazowego $\tilde{Y}(u, v)$ w obrazie \mathbf{X} . W szczególności współczynnik $Y(0,0)$ określa jasność całego obrazu. Rys.48 zawiera obrazy bazowe o niskich częstotliwościach przestrzennych (duże obiekty) i wysokich częstotliwościach przestrzennych (drobne szczegóły). Te pierwsze gromadzą się w lewym górnym rogu a te drugie – w prawym dolnym rogu.



7.4 Kompresja obrazu w dziedzinie transformaty

Dwustronna transformata DCT reprezentuje obraz w dziedzinie częstotliwości przestrzennych. Jeśli obraz nie zawiera drobnych szczegółów, to wystarczy kilka obrazów bazowych do jego zapisania lub przesłania. Oznacza to, że wystarczy wykorzystać tylko kilka współczynników DCT do jego reprezentacji. Mamy więc metodę kompresji: obliczenie DCT i transmisja tylko kilku najistotniejszych współczynników. Na tej zasadzie działa algorytm kompresji JPEG (Joint Photographic Expert Group) [14], [17], [18].



Na rys.49 pokazano podstawowy wariant kodera JPEG. DCT oblicza się dla podobrazów 8x8 pikseli: $Y = WXW^t$ (wzór 40). Dla każdego podobrazu mamy 64 współczynniki DCT. Kwantyzacja tych współczynników jest procesem dwuetapowym. Na początku dzieli się każdy współczynnik DCT przez odpowiedni element tablicy kwantyzacji $Q(i, j)$.

$$\hat{Y}(i, j) = Y(i, j)/Q(i, j) \quad (46)$$

Przykład tablicy kwantyzacji podano na rys.50. Znormalizowane współczynniki DCT zaokrągla się do najbliższej liczby całkowitej:

$$\hat{Y}^*(i, j) = \text{round}(\hat{Y}(i, j)) \quad (47)$$

Im większe liczby umieścimy w tablicy kwantyzacji, tym więcej elementów zerowych otrzymamy w wyniku zaokrąglenia i tym głębsza będzie kompresja obrazu. Stosując tablicę kwantowania z rys.50, otrzymamy mniej elementów zerowych w lewym górnym rogu (reprezentacja dużych obiektów) a mniej w prawym dolnym (reprezentacja drobnych szczegółów).

16	11	10	16	24	15	1	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	81	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Rysunek 50 Przykład tablicy kwantyzacji [18]

Zaokrąglenie do wartości całkowitych jest procesem stratnym, tu powstaje błąd kwantyzacji. Pozostałe kroki algorytmu JPEG są bezstratne. Współczynnik $\hat{Y}^*(0,0)$, zwany współczynnikiem DC (określa on ogólną jasność podobrazu) jest kodowany różnicowo (oblicza się różnicę współczynników DC dwóch sąsiednich bloków 8x8 pikseli). Pozostałe współczynniki (AC) szereguje się od najniższych do najwyższych częstotliwości przestrzennych, poruszając się „zygzakiem” od lewego górnego rogu do prawego dolnego. Kodowaniu podlegają tylko niezerowe wartości współczynników $\hat{Y}^*(i,j)$. Dla każdego niezerowego współczynnika kodowana jest:

- Liczba poprzedzających zer,
- Liczba bitów dla kodowania wartości współczynnika,
- Wartość współczynnika.

W końcu otrzymane 3 parametry na niezerowy współczynnik DCT koduje się kodem Huffmana o zmiennej długości słowa.

W odbiorniku (rys.49) dekoduje się kod Huffmana, otrzymując współczynniki $\hat{Y}^*(i,j)$. Następnie odwraca się operację normalizacji (46):

$$Y^*(i,j) = \hat{Y}^*(i,j) Q(i,j) \quad (48)$$

W końcu otrzymuje się skwantowany podobraz, stosując transformatę odwrotną IDCT: $X^* = W^t Y^* W$ (wzór 42).

Przykładowe wyniki otrzymano z wykorzystaniem softweru opublikowanego w [18] i zamieszczono w tablicy 5. Wskaźnik jakości Q wpływa na zawartość tablicy kwantyzacji kwantyzacji $Q(i,j)$. Im większa wartość Q, tym mniejsze są elementy tablicy kwantyzacji i tym dokładniejsze kwantowanie (wyższe wartości PSNR dla 3 kolorów). Oczywiście płacimy za to większym rozmiarem pliku po kompresji.

Tabela 5 Kompresja obrazu metodą JPEG

Wskaźnik jakości	Rozmiar pliku (kB)	PSNR [dB] R G B
<u>Oryginał-kolor</u> *	-	- - -
<u>Q=70</u>	46.6	37.8 , 29.3 , 36.2
<u>Q=50</u>	36.0	35.1 , 36.3 , 34.1
<u>Q=30</u>	25.1	32.6 , 33.6 , 31.9
<u>Q=10</u>	14.3	28.0 , 29.0 , 27.5
<u>Q=5</u>	10.9	25.0 , 26.0 , 24.3

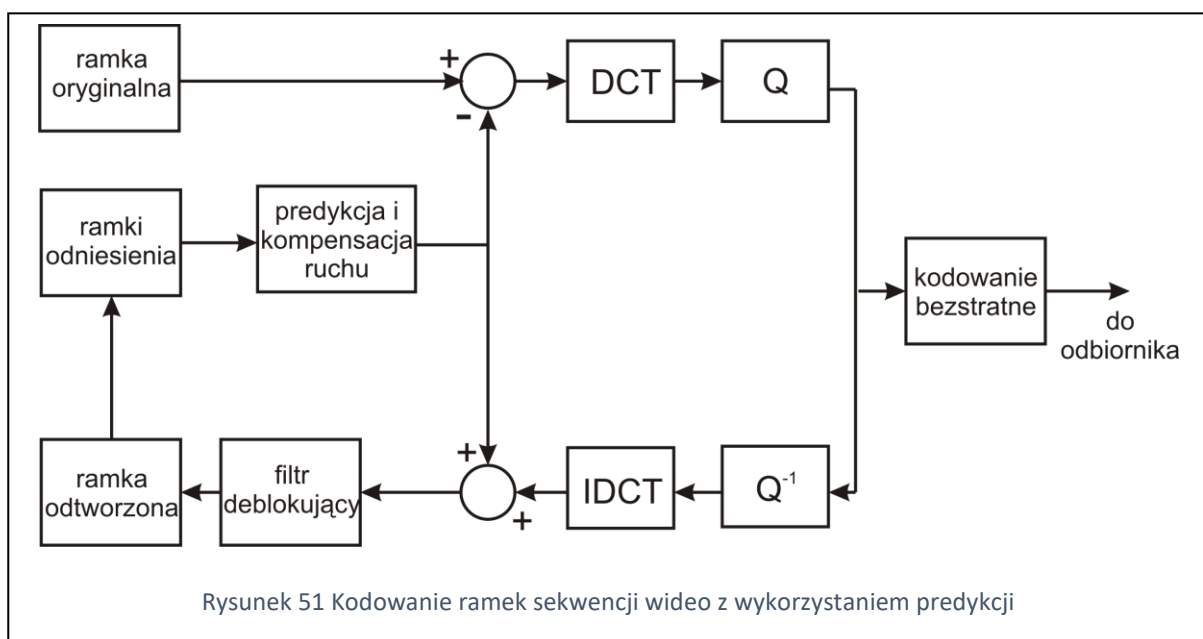
*Otwórz hiperlink aby zobaczyć obraz

7.5 Kompresja sekwencji wideo

Obraz ruchomy (sekwencja wideo) składa się z szeregu ramek (obrazów nieruchomych), które można byłoby kodować np. z wykorzystaniem algorytmu JPEG. Taka metoda nosi nazwę MJPEG (Motion JPEG) i wykorzystywana przy montażu sekwencji wideo, gdy rozdziela się i łączy poszczególne ujęcia. Dużo głębszą kompresję uzyskuje się dzięki wykorzystaniu podobieństwa kolejnych ramek. Można wówczas zastosować predykcję, która w kodowaniu obrazu pozwala na znaczną redukcję przepływności binarnej z zachowaniem wymaganej jakości. Na tej zasadzie działają algorytmy MPEG (Moving Pictures Expert Group).

Jako przykład weźmy popularny koder sekwencji wideo H.264 [17]. W koderze tym, podobnie jak w wielu innych, wyróżnia się ramki typu I (intra), P (predictive) i B (bi-predictive). Ramki typu I są kodowane z pominięciem informacji pochodzącej z poprzednich ramek. Ma to na celu zatrzymanie propagacji błędu, pochodzącego z przekłamań transmitowanych bitów. Pewien rodzaj predykcji jest tu jednak stosowany - wykorzystuje się podobieństwo sąsiednich bloków o rozmiarach 16x16 lub 4x4 piksele. Do zakodowania ramki I potrzeba znacznie więcej bitów niż dla ramek wykorzystujących predykcję.

Ramka typu P jest kodowana z wykorzystaniem systemu odwołań do ramek poprzednich (ramek referencyjnych lub ramek odniesienia). Schemat kodowania ramek typu P jest pokazany na rys. 51.



Predykcji podlegają bloki pikseli o rozmiarach od 4x4 do 16x16. W ramach odniesienia poszukuje się bloków, które uległy przemieszczeniu skutkiem ruchu obiektów lub ruchu kamery. Koduje się wektory przesunięć

kolejnych bloków. Odejmując wynik predykcji od oryginalnej ramki, otrzymujemy bloki ramki rezydualnej, zawierającej błąd predykcji. Bloki pikseli błędu predykcji są kodowane w dziedzinie transformaty DCT, podobnie jak to miało miejsce w algorytmie JPEG. Sam proces kwantyzacji polega na dzieleniu współczynników DCT przez pewną liczbę, zwaną krokiem kwantyzacji (Qstep) i zaokrągleniu wyniku do liczby całkowitej.

Skwantowane współczynniki DCT są poddawane bezstratnej kompresji (cały proces kodowania jest jednak stratny, ze względu na błąd kwantyzacji). Koduje się niezerowe wartości współczynników, stosując algorytmy CAVLC i CABAC. Pierwszy jest rodzajem adaptacyjnego kodu Huffmana, a drugi koderem arytmetycznym [17]. Oprócz ramek I oraz P można (opcjonalnie) wprowadzić ramki typu B, które są obliczane jako wynik predykcji z wykorzystaniem poprzednich i następnych ramek. Wymagają one niewielkiej liczby bitów do ich zakodowania.

W koderze H.264 można zadeklarować strukturę danych w postaci sekwencji ramek I,P i B, tzw. GOP (Group Of Pictures). Jest dostępnych 8 GOP (w nawiasie podano liczbę ramek w grupie): IPPPP(5), IPPPPPP(7), IPPPPPPPP(10), IPPPPPPPPPP(12), IBPBP(5), IBBPBBP(7), IBBPBBPBBP(10), IBBPBBPBBPBB(12).

Przeptywność binarna zależy od rozmiarów obrazu, liczby ramek na sekundę, wymaganej jakości, oraz samej sekwencji wideo. W systemie VBR (variable bit rate) szybki ruch powoduje zwiększenie przepływności a sekwencje statyczne (typu „mówiąca głowa”) dają się zakodować bardzo oszczędnie.

8 Zadania i testy

8.1 Przykładowe zadania z rozwiązaniami

1.

Kwantując sygnał mowy w pełni wysterowanym kwantyzatorem równomiernym o 16 poziomach kwantowania otrzymano $SNR = 22$ dB. Jakiej wartości SNR należy się spodziewać przy użyciu w pełni wysterowanego kwantyzatora równomiernego o 64 poziomach kwantyzacji? Odpowiedz uzasadnij

Rozwiązanie: $L=16$ poziomów to $b=4$ bity na próbkę. Dodanie 1 bitu oznacza wzrost SNR o 6 dB. 64 poziomy to 6 bitów na próbkę. Musimy dodać 2 bity do posiadanych 4. SNR zwiększy się o 12 dB i osiągnie wartość $22+12=34$ dB.

2.

Kwantując sygnał analogowy w pełni wysterowanym kwantyzatorem równomiernym o 128 poziomach kwantowania otrzymano $SNR = 10000$ (chodzi o stosunek mocy). Jakiej wartości SNR należy się spodziewać dla kwantyzatora o 256 poziomach kwantowania?

Rozwiązanie: W poprzednim zadaniu podane były wartości SNR w dB, teraz jako stosunek mocy. Można przeliczyć stosunek mocy na decybele (10000 to 40 dB). Można też uwzględnić, że wzrost mocy o 6 dB to wzrost 4-krotny. 128 poziomów to 7 bitów na próbkę, 256 poziomów to 8 bitów na próbkę. Dodano 1 bit, więc SNR wzrośnie 4 razy i osiągnie wartość 40000. W decybelach to będzie $40+6=46$ dB.

3.

Naszkcuj SNR_0 (na wyjściu akustycznym) w funkcji mocy sygnału podlegającego kwantyzacji (σ_x^2) dla kwantyzatora równomiernego, logarytmicznego i adaptacyjnego. Osie mają być wyskalowane w decybelach.

Wskazówka: Rys.8 dla kwantyzatora równomiernego i logarytmicznego. Teoretycznie kwantyzator adaptacyjny powinien mieć stałą wartość SNR_0 . W praktyce zawsze będzie przesterowanie. Dla bardzo cichych sygnałów pojawi się wpływ szumów układów elektronicznych. W każdym razie płaska część charakterystyki

kwantyzatora adaptacyjnego powinna rozciągać się w szerszym zakresie zmian mocy sygnału niż miało to miejsce dla kwantyzatora logarytmicznego.

4.

Sygnał telefoniczny poddano próbkowaniu (8000 próbek/s), kwantowaniu i transmisji z przepływnością 32 kbit/s, otrzymując $SNR = 22$ dB. Następnie zwiększono przepływność do 56 kbit/s. Jakiej wartości SNR należy się teraz spodziewać? Odpowiedź uzasadnij.

Rozwiązanie: $32000/8000=4$ bity na próbkę. $56000/8000=7$ bitów na próbkę. Przybyły 3 bity, a więc SNR wzrośnie o 3 razy $6 = 18$ dB. Osiągnie wartość $22+18=40$ dB

5.

Współczynniki autokorelacji sygnału wynoszą $R_0 = 1$, $R_1 = 0.8$ (pozostałych współczynników nie znamy). Zastosowano najprostszy (nieoptymalny) predyktor, przewidujący wartość bieżącej próbki jako wartość próbki poprzedniej: $x_n^p = x_{n-1}$. Jaką moc błędu predykcji udało się osiągnąć? Jaki jest zysk predykcji?

Rozwiązanie: Dla tego predyktora $\sigma_\varepsilon^2 = 2R_0 - 2R_1$ (wzór 23). Wynik: $\sigma_\varepsilon^2 = 0.4$. Zysk predykcji to stosunek mocy sygnału $\sigma_x^2 = R_0$ do mocy błędu predykcji σ_ε^2 . Wynik $G_p = 2.5$.

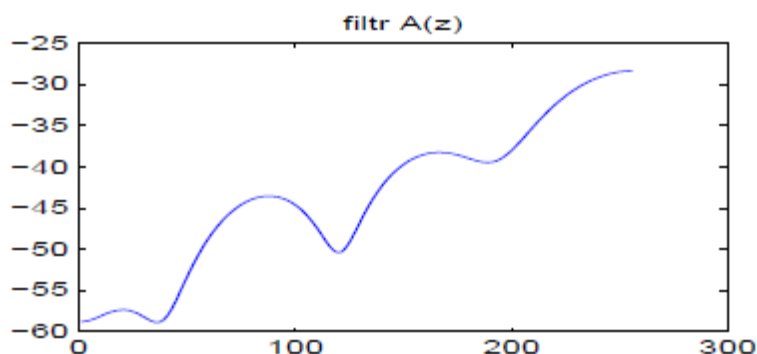
6.

Obliczono współczynniki autokorelacji sygnału akustycznego: $R_0 = 1$, $R_1 = 0.8$. Predyktor przewiduje próbkę x_n w następujący sposób: $x_n^p = a_1 x_{n-1}$. Oblicz optymalny współczynnik predykcji a_1 , który zapewnia najmniejszą moc błędu predykcji. Jaki jest zysk predykcji?

Rozwiązanie: Wg wzoru (24) $a_1 = \frac{R_1}{R_0} = 0.8$. Moc błędu predykcji $\sigma_\varepsilon^2 = R_0 - 2a_1R_1 + (a_1)^2R_0 = 0.36$. Zysk $G_p = 2.77$. Zauważmy że jest to predyktor nieco lepszy od predyktora z poprzedniego zadania.

7.

Na rysunku pokazano charakterystykę częstotliwościową (w zakresie od częstotliwości 0 do połowy częstotliwości próbkowania) filtra $A(z) = 1 - P(z) = 1 - \sum_{i=1}^N a_i z^{-i}$ przetwarzającego sygnał mowy na błąd predykcji. Ile (co najmniej) wynosi liczba współczynników predykcji N ? Odpowiedź uzasadnij.

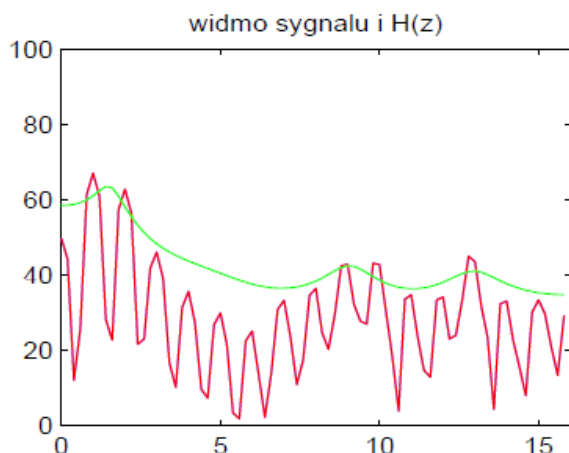


Rozwiązanie: Filtr $A(z)$ ma N zer, w związku z tym jego charakterystyka częstotliwościowa może mieć najwyżej N minimów w zakresie częstotliwości od 0 do częstotliwości próbkowania. Widzimy 4 minima do połowy częstotliwości próbkowania. Po dorysowaniu lustrzanego odbicia będzie ich 8. Trzeba jednak zauważyć, że za minimum na częstotliwości zerowej i minimum na częstotliwości próbkowania odpowiada to samo zero leżące na osi rzeczywistej. Tak więc zer będzie 7. Może ich być więcej (jeśli leżą w pobliżu początku układu współrzędnych, to nie wywołują widocznych zmian charakterystyki częstotliwościowej filtra). Nie może jednak być ich mniej niż 7. Patrz p.7 Moduł 1.

8.

Na rysunku pokazano widmo fragmentu mowy i charakterystykę częstotliwościową filtra predykcyjnego $H(z)$ (przetwarzającego sygnał błędu predykcji na sygnał mowy) w zakresie częstotliwości od zera do połowy częstotliwości próbkowania. Transmitancja filtra wynosi $H(z) = \frac{1}{1-P(z)} = \frac{1}{1-\sum_{i=1}^M a_i z^{-i}}$.

Ile (co najmniej) wynosi liczba współczynników predykcji M ? Odpowiedź uzasadnij.



Rozwiązanie: Filtr $H(z)$ ma M biegunów, nie ma zer wpływających na charakterystykę częstotliwościową (zera leżące w początku układu współrzędnych nie mają takiego wpływu). Każdy biegun może wytworzyć maksimum charakterystyki. Jest 6 maksimumów w zakresie częstotliwości od zera do częstotliwości próbkowania (na rysunku widzimy 3, bo nie obserwujemy lustrzanego odbicia). 6 maksimumów to co najmniej 6 biegunów, a więc co najmniej 6 współczynników predykcji.

9.

Kwantyzator wektorowy posiada słownik liczący $L = 256$ wektorów, kwantuje on wektory złożone z $N = 4$ kolejnych próbek sygnału mowy. Jaka jest rozdzielczość tego kwantyzatora wektorowego? (b , w bitach na próbkę). Ile poziomów kwantyzacji miałby kwantyzator skalarny pracujący z tą samą rozdzielczością?

Rozwiązanie: Ze wzoru $L = 2^{bN}$ od razu wynika $b = 2$. Kwantyzator skalarny o $b=2$ bitach na próbkę ma $L = 2^b = 4$ poziomy kwantyzacji.

10.

Częstotliwość próbkowania wynosi $f_s = 32\text{kHz}$, rozdzielczość kwantyzatora wektorowego $b = 2$ bit/próbkę, wymiar kwantowanych wektorów $N = 5$. Ile jest wektorów w słowniku? Ile operacji arytmetycznych (mnożeń) na sekundę wykonuje nadajnik kwantyzatora wektorowego przy przeszukiwaniu słownika?

Rozwiązanie: Liczba wektorów $L = 2^{bN} = 1024$. Przeszukując słownik wykonuje się $LN = 5120$ operacji. Słownik przeszukiwany jest $32000/5 = 6400$ razy na sekundę. Daje to $6400 \times 5120 = 32768000$ operacji na sekundę.

11.

Częstotliwość próbkowania wynosi 8kHz , słownik kształtów (kwantyzatora wektorowego typu kształt-wzmocnienie) liczy 512 wektorów 4-wymiarowych, a wzmacnienie jest kodowane w 8 poziomach. Jaka jest przepływność binarna?

Rozwiązanie: Kształt jest kodowany w 9 bitach ($2^9 = 512$), a wzmacnienie w 3 bitach ($2^3 = 8$), razem na wektor przypada $9 + 3 = 12$ bitów. Transmituje się $8000/4 = 2000$ wektorów na sekundę. Przepływność wynosi zatem $2000 \times 12 = 24000$ bitów na sekundę.

12.

W koderze 2-kanalowym moce sygnału fonicznego wynoszą w obu podpasmach częstotliwości: 30 dB i 18 dB. Rozdzielić 4 bity pomiędzy kwantyzatory adaptacyjne operujące w tych podpasmach w taki sposób, aby ogólna moc szumu kwantowania była jak najmniejsza. Jaka będzie moc szumu w każdym kanale?

Rozwiązanie:

Przed przydzieleniem bitów kwantyzatory nie działają, w odbiorniku można jedynie odtworzyć ciszę. Strata (błąd, wynikający z nieodtworzenia sygnału) jest większa w pierwszym podpaśmie (30 dB > 18 dB), więc tam przydzielamy pierwszy bit. Kwantyzator 1-bitowy (2-poziomowy) ma szum kwantowania około 6 dB poniżej mocy kwantowanego sygnału, a więc moc szumu kwantowania wyniesie 30dB - 6 dB = 24 dB.

Po przydzieleniu pierwszego bitu, szum w pierwszym kanale jest nadal większy, niż strata w drugim kanale (24 dB > 18 dB). Drugi bit przydzielamy więc pierwszemu kanałowi, skutkiem czego moc szumu kwantowania obniży się o dalsze 6 dB (zasada 6dB na bit), osiągając wartość 24-6=18 dB. Teraz błąd w obu kanałach jest równy, wynosi po 18 dB. Dwa pozostałe bity rozdzielimy więc po jednym na każdy kanał. Ostatecznie szum kwantowania w obu kanałach będzie wynosił 18-6=12 dB. Pierwszy kanał otrzymał 3 bity, drugi 1 bit.

8.2 Zadania do rozwiązania

1.

Kwantując sygnał mowy w pełni wysterowanym kwantyzatorem równomiernym o 16 poziomach kwantowania otrzymano $SNR = 22$ dB. Ile poziomów kwantyzacji trzeba użyć, aby osiągnąć $SNR=40$ dB? Odpowiedz uzasadnij

2.

Sygnał telefoniczny poddano próbkowaniu (16000 próbek/s), kwantowaniu i transmisji z przepływnością 32 kbit/s, otrzymując $SNR = 10$ dB. Następnie zwiększono przepływność do 64 kbit/s. Jakiej wartości SNR należy się teraz spodziewać? Odpowiedz uzasadnij.

3.

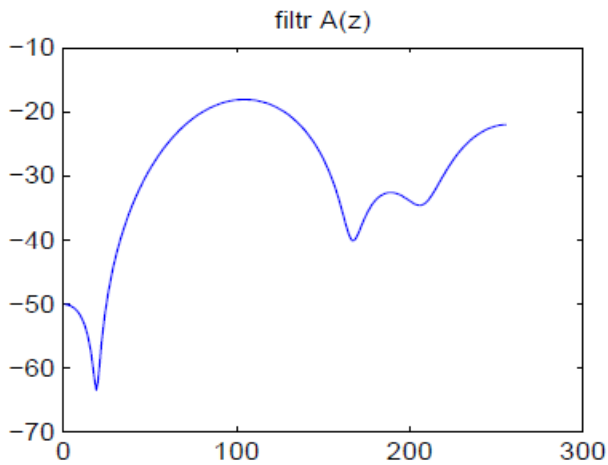
Obliczono współczynniki autokorelacji sygnału akustycznego: $R_0 = 1$, $R_1 = 0.9$. Predyktor przewiduje próbkę x_n w następujący sposób: $x_n^p = x_{n-1}$. Czy jest to predyktor optymalny? Jeśli nie, to zaproponuj lepszy predyktor.

4.

Współczynniki autokorelacji sygnału wynoszą $R_0 = 1$, $R_1 = 0.5$ (pozostałych współczynników nie znamy). Zastosowano predyktor, przewidujący wartość bieżącej próbki jako iloczyn wartości poprzedniej próbki i liczby a_1 . Dla jakich wartości a_1 moc błędu predykcji będzie mniejsza od mocy sygnału? Jaka wartość a_1 gwarantuje uzyskanie najmniejszej mocy błędu predykcji?

5.

Na rysunku pokazano charakterystykę częstotliwościową (w zakresie od częstotliwości 0 do połowy częstotliwości próbkowania) filtra $A(z) = 1 - P(z) = 1 - \sum_{i=1}^N a_i z^{-i}$ przetwarzającego sygnał mowy na błąd predykcji. Ile (co najmniej) wynosi liczba współczynników predykcji N ? Odpowiedz uzasadnij.



6.

Kwantyzator wektorowy posiada słownik liczący $L = 256$ wektorów, kwantuje on wektory złożone z $N = 16$ pikseli obrazu (kwadraty 4×4 piksele). Jaka jest rozdzielczość tego kwantyzatora wektorowego (b , w bitach na piksel). Czy można skonstruować kwantyzator skalarny pracujący z tą samą rozdzielczością?

7.

Kwantyzator wektorowy kwantuje obrazy niekolorowe (odcienie szarości). Posiada słownik liczący $L=16$ wektorów, kwantuje on wektory złożone z $N = 4$ pikseli obrazu (kwadraty 2×2 piksele). Jaka jest rozdzielczość tego kwantyzatora wektorowego (b , w bitach na piksel). Ile poziomów kwantyzacji posiada kwantyzator skalarny pracujący z tą samą rozdzielczością?

8.

Częstotliwość próbkowania wynosi 16kHz, słownik kształtów (kwantyzatora wektorowego typu kształt-wzmocnienie) liczy 16 wektorów 4-wymiarowych, a wzmacnienie jest kodowane w 16 poziomach. Jaka jest przepływność binarna?

9.

W koderze subpasowym 2-kanalowym moce sygnału użytecznego wynoszą 24 dB w pierwszym kanale i 12 dB w drugim. Jak rozdzielić 4 bity, aby błąd kwantyzacji był jak najmniejszy? Jaki będzie błąd kwantyzacji w każdym kanale?

10.

W koderze subpasowym 3-kanalowym moce sygnału użytecznego wynoszą 48 dB w pierwszym kanale, 30 dB w drugim i 36 dB w trzecim. Jak rozdzielić 4 bity, aby błąd kwantyzacji był jak najmniejszy? Jaki będzie błąd odtworzenia sygnału w każdym kanale?

9 Biblioteka

Z Modułem 2 są związane dwa ćwiczenia symulacyjne: Kwantyzacja skalarna i Kompresja sygnału mowy. Pierwsze z nich zapoznaje studenta z kwantyzatorami: równomiernym, pseudologarytmicznym i adaptacyjnym. Do kwantyzatora adaptacyjnego można dołączyć predyktor, otrzymując układ ADPCM. Drugie ćwiczenie zawiera symulator koder GSM-AMR. Można obserwować sygnały występujące w koderze CELP, w

tym sygnał pobudzenia algebraicznego, predykcji długookresowej. Jest możliwość odsłuchu fraz mowy poddanych stratnej kompresji i porównanie z wartością SNR.

Ponadto opracowano opcjonalne symulacje ilustrujące wykład: Kwantyzacja wektorowa i Kodowanie transformaty sygnału audio.

Laboratorium i materiały pomocnicze opisano w Module 4.

10 Bibliografia

1. T.P. Zieliński "Cyfrowe przetwarzanie sygnałów w telekomunikacji", PWN, 2014
2. A.Dąbrowski, P.Dymarski (ed.) „Podstawy Transmisji Cyfrowej” Oficyna Wyd. Politechniki Warszawskiej, 2013
3. A.Drozdek “Wprowadzenie do kompresji danych”, WNT 2007
4. ITU-T Recommendation G.711: Pulse Code Modulation (PCM) of voice frequencies,1984.
5. ITU-T Recommendation G.726: 40-, 32-, 24-, and 16 kbit/s adaptive differential pulse code modulation.
6. ITU-T Recommendation G.728: Coding of speech at 16 kbit/s using low delay code excited linear prediction, Sep. 1992.
7. ITU-T Recommendation G.729: Coding of speech at 8kbit/s using conjugate structure algebraic code excited linear prediction CS-ACELP, March 1996.
8. ITU-T Recommendation G.723.1: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s, March 1996.
9. ITU-T Recommendation G.722: 7 kHz audio-coding within 64 kbit/s, Nov. 1988
10. ITU-T Recommendation G.722.1 : Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss, 2005
11. ITU-T Recommendation G.722.2 : Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB), 2003
12. ETSI Standard GSM 06.60: Digital cellular telecommunications system; enhanced full rate EFR speech transcoding, 1996.
13. ETSI EN 301 703: Digital cellular telecommunications system (Phase 2+); Adaptive Multi-Rate (AMR); Speech processing functions; General description. European Telecommunications Standards Institute. 1999.
14. W.Skarbek "MULTIMEDIA – algorytmy i standardy kompresji" PLJ 1998
15. ISO/IEC Standard IS 11172-3 Information technology -- Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s -- Part 3: Audio, 1993
16. ITU-T Recommendation P.862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs, 02/2001
17. D.Karwowski: Zrozumieć kompresję obrazu, ISBN 978-83-953420-0-4, Poznań 2019, www.zrozumiieckompresje.pl
18. S.E. Umbaugh, Digital image processing and analysis, CRC Press, 2011