**DOKUMENTACJA TECHNICZNA   
PROJEKTOWA**Projekt zespołowy  
Aplikacja kompresji bezstratnej plików wav   
C++

*GRUPA NR 1*

Spis treści

[1.Zakres projektu 2](#_Toc504035576)

[2. Budowa pliku .wav 2](#_Toc504035577)

[3.Działania wstępne 3](#_Toc504035578)

[4. Dostępne pliki dźwiękowe 4](#_Toc504035579)

[5.Kodowanie predykcyjne – działania wstępne 6](#_Toc504035580)

[6. Kodowanie predykcyjne - Metoda LU rozwiązywania układów równań 7](#_Toc504035581)

[7. Kodowanie bitowe 9](#_Toc504035582)

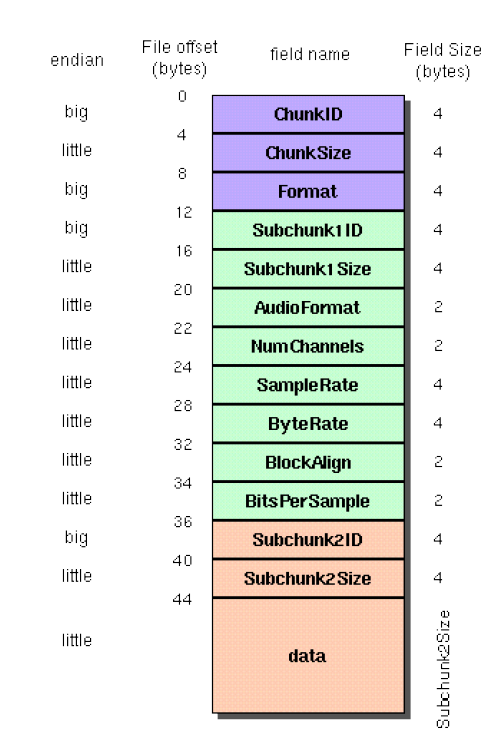
[8. Kodowanie bitowe – dzielenie pliku 13](#_Toc504035583)

[9. Dekodowanie 14](#_Toc504035584)

[10. Badania stereo 15](#_Toc504035585)

1.Zakres projektu

Przedmiotem niniejszego opracowania jest aplikacja służąca do bezstratnej kompresji plików dźwiękowych w formacie .wav.

2. Budowa pliku .wav  


# 3.Działania wstępne

Na początku pracy wszystkie dane zostały sczytane z pliku .wav na podstawie budowy pliku zawartego w punkcie 2.

Na podstawie schematu budowy pliku .wav zawartego w punkcie 2 sczytane zostały dane z plików .wav. W ten sposób uzyskaliśmy wszystkie niezbędne informacje na ich temat by móc przystąpić do wykonywania obliczeń. Po odczytaniu próbek z pliku, obliczone zostało prawdopodobieństwo wystąpienia każdej z próbek w pliku według wzoru 1. Działanie to było konieczne do wykonania kolejnych kroków, obliczenia entropii dla próbek z obu kanałów pliku oraz entropii dla obu kanałów po skanowaniu różnicowym według ogólnego wzoru 2.

*Pi – prawdopodobieństwo wystąpienia próbki*

*Xi-wartość próbki*

*N-ilość próbek*

*H(s) – entropia*

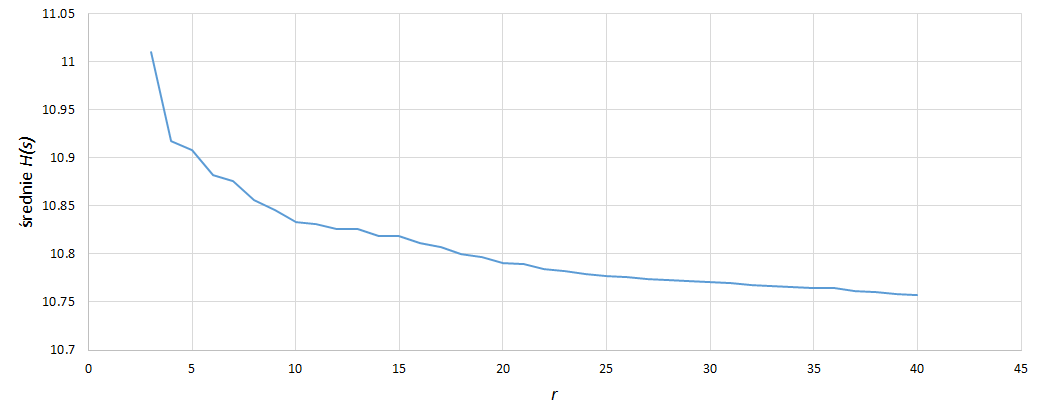
Dla danych wejściowych podczas liczenia entropii używamy przedziału od -215 do 215-1, natomiast dla danych po skanowaniu różnicowym od -216 do 216-1.

4. Dostępne pliki dźwiękowe

Tabela 1 – wartości energii oraz entropii po kodowaniu różnicowym

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Lp | Nazwa | Liczba próbek | Energia | Energia po kodowaniu różnicowym | Entropia | Entropia po kodowaniu |
| 1 | ATrain.wav | 1688532 | 984594,6573 | 45939,878405 | 11,882513 | 9,422548 |
| 2 | BeautySlept.wav | 1378524 | 2179309,207 | 297319,01 | 11,889538 | 10,719252 |
| 3 | Chanchan.wav | 1063086 | 25028113,11 | 1148861,33 | 14,182758 | 11,693296 |
| 4 | death2.wav | 1652752 | 61346235,94 | 1622998,78 | 14,54073 | 10,056352 |
| 5 | Experiencia.wav | 1764594 | 54135050,72 | 4055978,39 | 14,831842 | 12,680249 |
| 6 | female\_speech.wav | 1692324 | 5706822,885 | 517869,67 | 12,45567 | 9,494601 |
| 7 | FloorEssence.wav | 1764390 | 43845078,29 | 2696927,52 | 14,685036 | 12,352193 |
| 8 | ItCouldBeSweet.wav | 1765502 | 78248399,62 | 1841792,57 | 15,101419 | 11,863736 |
| 9 | Layla.wav | 1749212 | 12223517,69 | 1612342,22 | 13,737941 | 11,506189 |
| 10 | LifeShatters.wav | 1756162 | 51664219,33 | 3390928,80 | 14,806516 | 12,862775 |
| 11 | Macabre.wav | 1540680 | 15838507,54 | 361481,97 | 13,831368 | 11,164276 |
| 12 | male\_speech.wav | 1583474 | 7765487,974 | 89120,08 | 12,372756 | 9,361494 |
| 13 | SinceAlways.wav | 1604478 | 45135680,34 | 6092121,145 | 14,576205 | 12,941404 |
| 14 | thear1.wav | 1703936 | 79263527,73 | 4073791,35 | 15,100597 | 12,970789 |
| 15 | TomsDiner.wav | 1751504 | 5268717,822 | 299478,93 | 12,642939 | 10,395441 |
| 16 | Velvet.wav | 1047690 | 53783610,41 | 7136233,34 | 13,828787 | 11,84226 |
|  | średnia | 1594177,5 | 33901054,58 | 2205199,06 | 13,779163 | 11,33292848 |

Na podstawie zawartości plików została obliczona energia oraz entropia a następnie wyniki zostały zoptymalizowane poprzez skanowanie różnicowe.



Rysunek 1 – stosunek średniej entropii dla wszystkich plików po kodowaniu różnicowym

Przy równomiernym zwiększaniu rozmiaru macierzy o 1 uśredniona entropia dla wszystkich 16 plików się zmniejsza (rysunek 1).

# 5.Kodowanie predykcyjne – działania wstępne

Do zrealizowania kodowania predykcyjnego potrzebny jest wektor *A* zawierający współczynniki predykcyjne. W celu jego ustalenia potrzebny jest rząd predykcji *r* oraz przewidywane wartości próbek. Naszym pierwszym krokiem było zbudowanie macierzy *X* (wzór 3) oraz *P* (wzór 4). Z wykorzystaniem wyliczonych macierzy możliwe było wyznaczenie wektora *A*.

Wzór (8) przedstawia założenie dotyczące współrzędnych predykcji mówiące, że suma współczynników predykcji powinna być wartością bliską 1. Na podstawie przeprowadzonych badań doszliśmy jednak do wniosku, że suma współczynników predykcji jest wartością poniżej 1.

6. Kodowanie predykcyjne - Metoda LU rozwiązywania układów równań

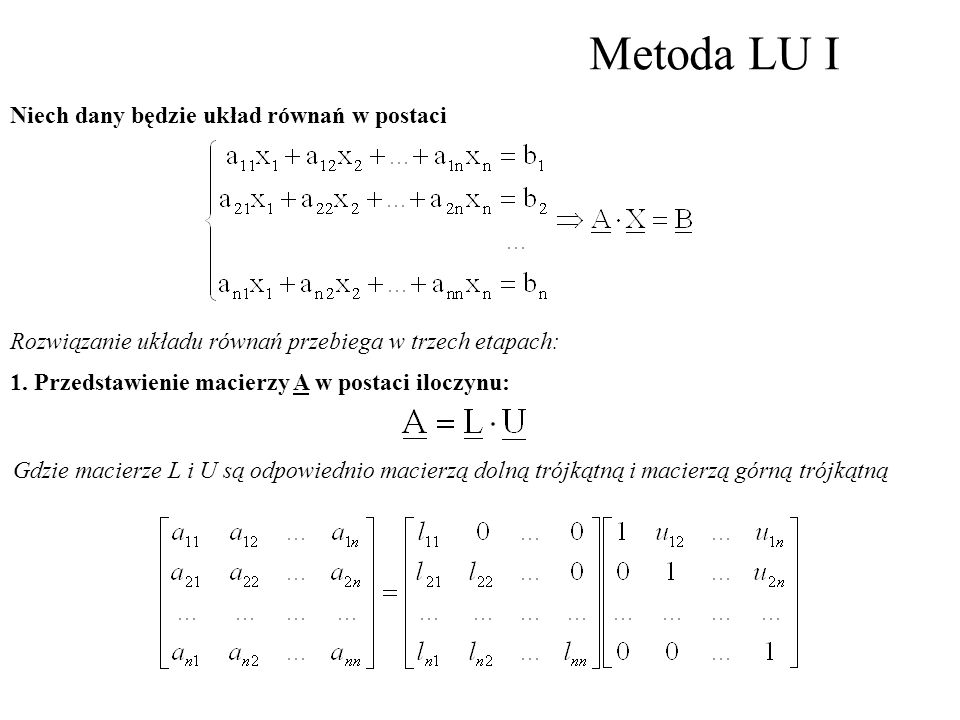
Aby jeszcze bardziej zmniejszyć wartość entropii, wykorzystaliśmy metodę rozwiązywania układu równań LU. Macierz *X* jest macierzą kwadratową *r* x *r*, przy czym *r* jest naszym współczynnikiem ustalonym z góry. Wektor *P* jest sumą kolejnych próbek.  
  
  
  
  
Obliczony wektor *A*, który przechowuje wyliczone współczynniki ze wzoru (X) wykorzystaliśmy do obliczenia błędu średnio-kwadratowego dzięki czemu byliśmy w stanie jeszcze bardziej zoptymalizować wartości entropii względem kodowania różnicowego. Tabela 2 przedstawia wartości entropii dla wszystkich plików po kodowaniu predykcyjnym dla współczynnika predykcji *r* = 3 oraz *r* = 40:

Tabela 2 – wartości entropii po kodowaniu predykcyjnym rzędu 3 i 40

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LP | Nazwa | Liczba próbek | Entropia po kodowaniu predykcyjnym MMSE r = 3 | Entropia po kodowaniu predykcyjnym MMSE r = 40 |
| 1 | ATrain.wav | 1688532 | 8,906222 | 8,472375 |
| 2 | BeautySlept.wav | 1378524 | 10,750141 | 10,547946 |
| 3 | Chanchan.wav | 1063086 | 10,930071 | 10,761007 |
| 4 | death2.wav | 1652752 | 10,532157 | 10,500842 |
| 5 | Experiencia.wav | 1764594 | 12,506854 | 12,269933 |
| 6 | female\_speech.wav | 1692324 | 9,428587 | 9,08874 |
| 7 | FloorEssence.wav | 1764390 | 11,939889 | 11,911469 |
| 8 | ItCouldBeSweet.wav | 1765502 | 11,672295 | 11,416037 |
| 9 | Layla.wav | 1749212 | 11,568866 | 11,2868 |
| 10 | LifeShatters.wav | 1756162 | 11,821592 | 11,376994 |
| 11 | Macabre.wav | 1540680 | 10,677783 | 10,308141 |
| 12 | male\_speech.wav | 1583474 | 8,532004 | 8,169717 |
| 13 | SinceAlways.wav | 1604478 | 12,744912 | 12,482144 |
| 14 | thear1.wav | 1703936 | 12,374375 | 12,155747 |
| 15 | TomsDiner.wav | 1751504 | 10,157716 | 9,835262 |
| 16 | Velvet.wav | 1047690 | 11,622075 | 11,527256 |
|  | Średnia | 1594178 | 11,010346 | 10,756900 |

# 7. Uwzględnienie wielkości nagłówka i jego optymalizacja

W celu zmniejszenia kosztu pamięci należy ustalić optymalną ilość bitów do kompresji. Aby znaleźć optymalną ilość bitów wyznaczyliśmy najmniejszą wartość średniej bitowej według wzoru 8 dla rzędu predykcji z przedziału od 10 do 600. Uzyskane wyniki prezentuje tabela 3.

*L*śrliczymy ze wzoru:

*L*śr – średnia bitowa  
32 – bo 4 bajty float  
b+1 liczba znaku bit  
10 nie wiem   
r – 1 zmniejszamy rząd predykcji

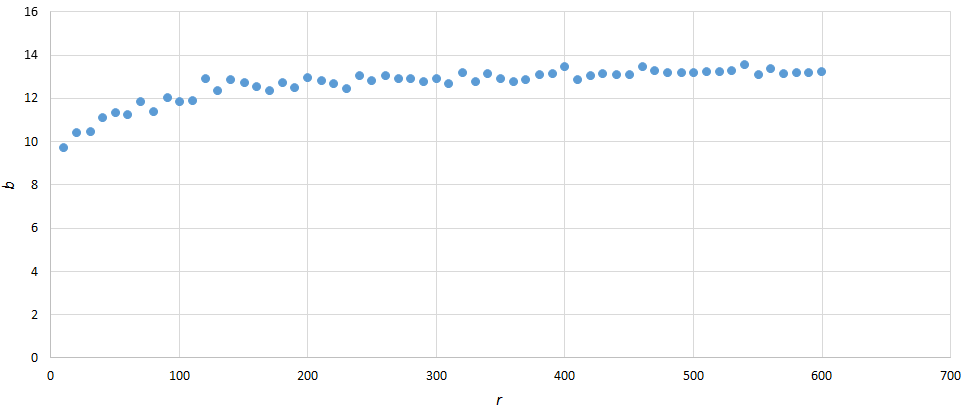
*H*(s) – entropia

*b* – ilość bitów

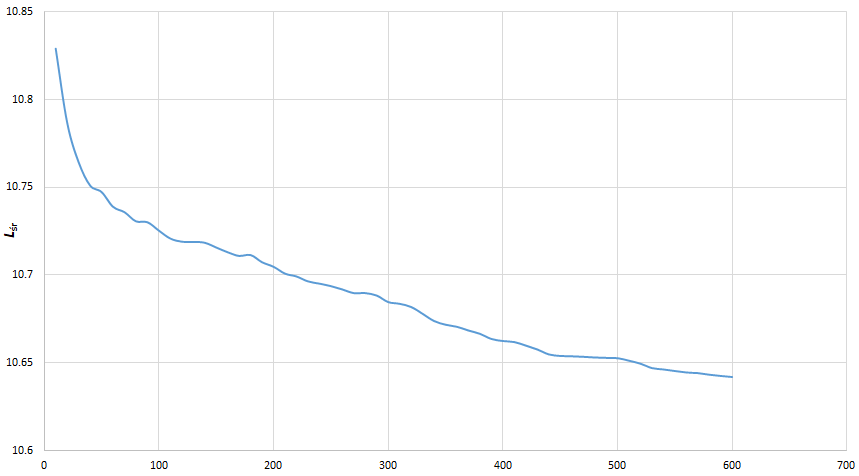
*r ­*– rząd predykcji

Tabela 3 – minimalne wartości średniej bitowej

|  |  |  |
| --- | --- | --- |
| *r* | *b* | *min Lśr* |
| 10 | 9,78125 | 10,829 |
| 20 | 10,5 | 10,7867 |
| 30 | 10,5313 | 10,7643 |
| 40 | 11,1875 | 10,751 |
| 50 | 11,4063 | 10,7473 |
| 60 | 11,2813 | 10,7389 |
| 70 | 11,9063 | 10,7358 |
| 80 | 11,4375 | 10,7306 |
| 90 | 12,0938 | 10,7301 |
| 100 | 11,9063 | 10,7253 |
| 110 | 11,9688 | 10,7208 |
| 120 | 13 | 10,719 |
| 130 | 12,4063 | 10,7189 |
| 140 | 12,9375 | 10,7184 |
| 150 | 12,7813 | 10,7157 |
| 160 | 12,5938 | 10,713 |
| 170 | 12,375 | 10,7109 |
| 180 | 12,7813 | 10,7114 |
| 190 | 12,5313 | 10,7073 |
| 200 | 13,0313 | 10,7047 |
| 210 | 12,875 | 10,7008 |
| 220 | 12,7188 | 10,6992 |
| 230 | 12,5 | 10,6964 |
| 240 | 13,0938 | 10,6951 |
| 250 | 12,875 | 10,6937 |
| 260 | 13,0625 | 10,6918 |
| 270 | 13 | 10,6897 |
| 280 | 13 | 10,6897 |
| 290 | 12,8438 | 10,6883 |
| 300 | 12,9688 | 10,6846 |
| 310 | 12,6875 | 10,6836 |
| 320 | 13,25 | 10,6818 |
| 330 | 12,8125 | 10,6779 |
| 340 | 13,1875 | 10,6738 |
| 350 | 13 | 10,6717 |
| 360 | 12,8438 | 10,6705 |
| 370 | 12,9063 | 10,6684 |
| 380 | 13,125 | 10,6665 |
| 390 | 13,1875 | 10,6636 |
| 400 | 13,5313 | 10,6624 |
| 410 | 12,9234 | 10,6618 |
| 420 | 13,1023 | 10,6598 |
| 430 | 13,2032 | 10,6576 |
| 440 | 13,1543 | 10,6548 |
| 450 | 13,13 | 10,6539 |
| 460 | 13,5264 | 10,6537 |
| 470 | 13,3253 | 10,6534 |
| 480 | 13,2543 | 10,653 |
| 490 | 13,239 | 10,6528 |
| 500 | 13,2188 | 10,6526 |
| 510 | 13,2805 | 10,6512 |
| 520 | 13,2843 | 10,6495 |
| 530 | 13,3554 | 10,647 |
| 540 | 13,6232 | 10,6462 |
| 550 | 13,125 | 10,6453 |
| 560 | 13,4282 | 10,6445 |
| 570 | 13,2123 | 10,6441 |
| 580 | 13,2453 | 10,6432 |
| 590 | 13,2683 | 10,6425 |
| 600 | 13,2813 | 10,6419 |



Rysunek 2 - Stosunek optymalnej wartości bitów do rzędu predykcji



Rysunek 3 - Stosunek minimalnego *L*śr do rzędu predykcji

W dalszych krokach w celu skompresowania pliku należy wyznaczyć maksymalną wartość z wektora *A* (amax). Będzie ona potrzebna przy kolejnych obliczeniach. Aby zmniejszyć wymaganą pamięć należy zapisać współczynniki predykcji na typach zajmujących mniejszą pamięć. W tym celu tworzymy dodatkową zmienną *s*i przechowującą informację o znaku. Jeśli współczynnik jest większy od 0 *s*iprzyjmuje wartość 1, w przeciwnym wypadku 0. Przekształcamy wartości z wektora *A* według wzoru 9.

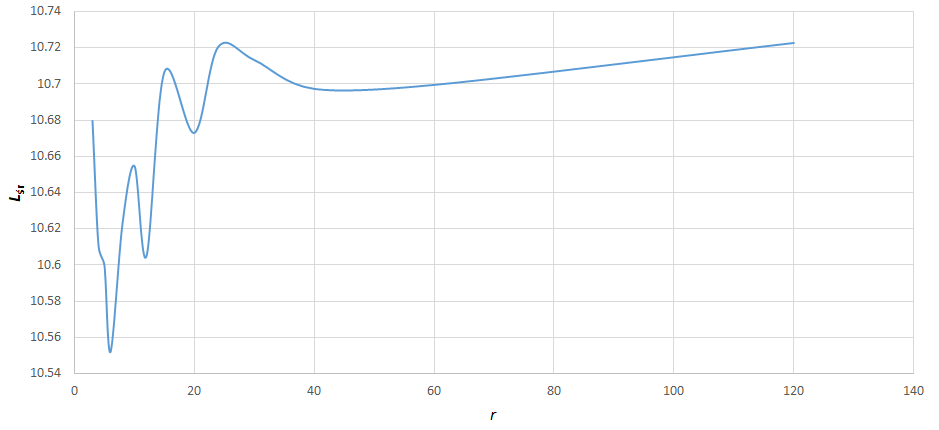
Po zastosowaniu powyższego przekształcenia otrzymujemy liczbę całkowitą dzięki której możemy zapisać *a*i na zmiennych o typach zajmujących mniej pamięci.

# **8. Kodowanie bitowe – dzielenie pliku ten punkt zly**

Kolejnym sposobem na znalezienie optymalnej liczby bitów będzie znalezienie najmniejszego *L*śr, jednak w tym przypadku plik zostaje podzielony na *k* części przy założeniu, że *k* ∙ *r* = 120. Dla wszystkich par {*k*,*r*} spełniających wspomniany warunek wyznaczone zostało minimalne *L*śr. Otrzymane w ten sposób wyniki przedstawia tabela 4 oraz rysunek 4.

Tabela 4 – wyniki badania *L*śr dla podziału pliku

|  |  |  |
| --- | --- | --- |
| *k* | *r* | *min Lśr* |
| 40 | 3 | 10,6797 |
| 30 | 4 | 10,6106 |
| 24 | 5 | 10,5995 |
| 20 | 6 | 10,552 |
| 15 | 8 | 10,6229 |
| 12 | 10 | 10,6547 |
| 10 | 12 | 10,6049 |
| 8 | 15 | 10,7068 |
| 6 | 20 | 10,6731 |
| 5 | 24 | 10,7207 |
| 4 | 30 | 10,7133 |
| 3 | 40 | 10,6974 |
| 2 | 60 | 10,6996 |
| 1 | 120 | 10,7228 |



Rysunek 4 – stosunek *L*śr do rzędu predykcji dla podziału pliku

# **9. Dekodowanie**

Zadanie dekodowania realizowane jest za pomocą wzoru 10.

do góry i si wyjasnic pzdr

# **10. Badania stereo**

Kolejnym przedmiotem badań będą sposoby wykorzystania zależności między kanałami do zwiększenia skuteczności kompresji. Podczas obliczania wartości przewidywanych próbek w badaniu stereo należy wziąć pod uwagę próbki zarówno kanału prawego jak i lewego. W pierwszej kolejności powinny zostać wyznaczone macierze **X** (wzór 13) oraz **P** (wzór 14). Wektor współczynników predykcji **A** należy obliczyć przy użyciu macierzy **X** i **P** jak przedstawiono (15). Dzięki zebranym danym możliwe będzie ustalenie wartości przewidywanych dla kanału prawego (wzór 17). Jak obrazują to wzory 18-22 postępowanie w przypadku kodowania kanału lewego jest analogiczne. PODZIAL STOSUNEK RZEDOUF

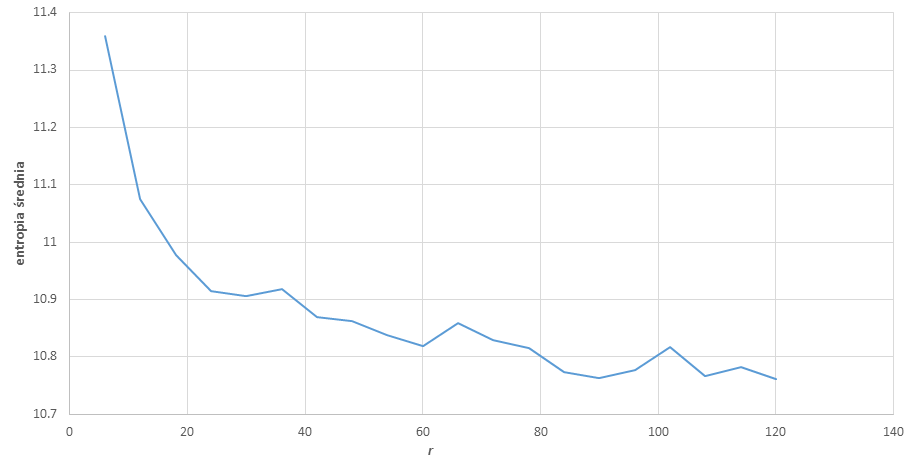
Kodowanie kanału prawego: r1 = 3 r2 = 2 r =5 Ddoppiiisz PRZYKŁAD TU I NA DOLE pochyl to

Kodowanie kanału lewego:

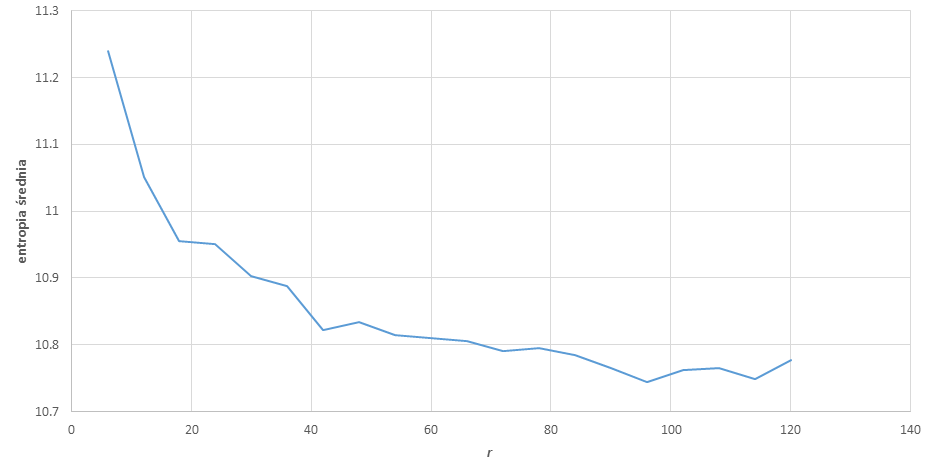
Po przeprowadzeniu kodowania obu kanałów można przejść do badania poziomu entropii w zależności od stosunku ustalonych rzędów predykcji. Wyniki przeprowadzonych testów przedstawione w tabeli 5 sugerują, że stosunek rzędów predykcji 2:1 dla kodowanych kanałów sprawdza się lepiej do pewnej wartości *r*. Dla rzędu predykcji na poziomie 114 można zauważyć na rysunku 5 (entropia dla stosunku *r* 1:1) tendencję spadkową wartości entropii. Sytuacja ma się odwrotnie dla stosunku 2:1.

Tablica 5 – entropia dla rzędów predykcji w stosunku 1:1 i 2:1

|  |  |  |
| --- | --- | --- |
| *r* | Entropia dla *r* 1:1 | Entropia dla *r* 2:1 |
| 6 | 11.3587 | 11.24 |
| 12 | 11.0744 | 11.0514 |
| 18 | 10.9778 | 10.9547 |
| 24 | 10.9153 | 10.9502 |
| 30 | 10.9056 | 10.9031 |
| 36 | 10.9185 | 10.8876 |
| 42 | 10.8698 | 10.822 |
| 48 | 10.8628 | 10.8347 |
| 54 | 10.8387 | 10.8151 |
| 60 | 10.8195 | 10.8095 |
| 66 | 10.8588 | 10.805 |
| 72 | 10.829 | 10.7912 |
| 78 | 10.8163 | 10.7958 |
| 84 | 10.7736 | 10.7844 |
| 90 | 10.7628 | 10.7659 |
| 96 | 10.778 | 10.7449 |
| 102 | 10.8173 | 10.7628 |
| 108 | 10.7668 | 10.7648 |
| 114 | 10.7831 | 10.7487 |
| 120 | 10.7616 | 10.7766 |



Rysunek 5 – entropia dla rzędów predykcji w stosunku 1:1



Rysunek 6 - entropia dla rzędów predykcji w stosunku 2:1