

RAPORT Z PRZEBIEGU PROJEKTU Z NIEZAWODNOŚCI I DIAGNOSTYKI UKŁADÓW CYFROWYCH 2

Transmisja w systemie FEC

Autorzy

Daniel Leśniewicz - 250996

Patryk Fidrych - 248828

Grupa zajęciowa

Środa, 7:30-9:00 TN

Prowadzący zajęcia

Dr hab. inż. Henryk Maciejewski

Spis treści

Cel i założenia projektu	2
Jak modelujemy kanał transmisyjny?	2
Jakie kody korekcyjne chcemy badać?	2
Z jakiego środowiska programistycznego skorzystamy, jakie biblioteki wykorzystamy do kodowania/dekodowania?	2
Co program powinien liczyć jako wyjście?	2
Opis symulatora	2
Organizacja eksperymentu symulacyjnego, wyniki, analiza wyników - III etap	3
Badanie skuteczności kodów korekcyjnych i ich kosztu związanego z nadmiarem kodowym	3
BER w zależności od jakości kanału	3
Nadmiarowość w zależności od liczby przesłanych bitów	5
Wnioski	8

Cel i założenia projektu

FEC (Forward Error Correction) – jest techniką dodawania nadmiarowości do transmitowanych cyfrowo informacji. Pozwala to na całkowitą lub częściową detekcję oraz korekcję błędów powstałych w wyniku zakłócenia. Główną zaletą jest to, że w takim podejściu nie ma potrzeby wykorzystywania kanału zwrotnego, przez który to miałyby przepływać informacje o błędach do nadawcy. Kodowanie korekcyjne ma zastosowanie gdy retransmisja jest kosztowna lub niemożliwa. W naszym projekcie zastosowaliśmy dwa kody korekcyjne i na ich podstawie przeprowadziliśmy analizę. Wymagało to również zamodelowania odpowiedniego kanału przez który przepływały informacje. Opisy poszczególnych elementów projektu przedstawiono poniżej.

Jak modelujemy kanał transmisyjny?

W kanale transmisyjnym pojawiać się będą losowo przekłamania. Z określonym prawdopodobieństwem dany bit będzie zmieniany na przeciwny - będzie symulowane przekłamanie. Kanał jest zamodelowany jako BSC (Binary Symetric Channel) z parametrem BER.

Jakie kody korekcyjne chcemy badać?

Każdy transmitowany bit może być potrajany. Dla takich założeń można przedstawić regułę podejmowania jak rozpoznawać dany ciąg bitów: np. większość bitów "1" w danej trójce bitów oznacza, że jest to bit "1". Dekoder według ustalonej reguły będzie podejmował decyzję jak zinterpretować przesłaną informację. Dodatkowo zbadany będzie kod BCH, którego koder oraz dekodek są zaimplementowane w środowisku Matlab.

Z jakiego środowiska programistycznego skorzystamy, jakie biblioteki wykorzystamy do kodowania/dekodowania?

Matlab, koder oraz dekodek kodu BCH zaimplementowany w środowisku Matlab.

Co program powinien liczyć jako wyjście?

- Bit Error Rate (BER)
- Jak zmienia się ilość przekłamanych bitów w zależności od prawdopodobieństwa przekłamywania bitów w kanale transmisyjnym

Opis symulatora

Głównym plikiem projektu jest plik *main.m*. W nim znajdują się wywołania poszczególnych funkcji oraz zaimplementowana jest bazowa logika symulatora. Na początku wygenerowany jest losowy wektor z odpowiednią ilością bitów. Następnie otrzymany wektor jest zakodowany przez odpowiedni koder. Na początku jest to koder potrajający każdy bit. Później następuje przesłanie tak zakodowanych danych przez kanał, który to generuje losowe zakłócenia. Po przesłaniu danych, wykonywane jest dekodowanie

otrzymanego ciągu bitów. Identyczna sytuacja występuje również dla kodowania BCH. W tym jednak przypadku w implementacji kodera wykorzystaliśmy funkcję z biblioteki – *bchenc()*. Po przesłaniu danych, następuje dekodowanie również przy użyciu funkcji z biblioteki – *bchdec()*. W każdym z przypadków po zdekodowaniu, następuje wyliczenie BER przy pomocy funkcji *errorCount()*. Porównuje ona dwie wektory – jeden przed wysłaniem przez kanał, a drugi po zdekodowaniu. Na ich podstawie zlicza ilość błędów.

Na wyjściu program tworzy plik o nazwie „WYNIKI.txt”, w którym znajdują się zbiorcze wyniki podsumowujące ilość przesłanych bitów oraz wyliczony BER.

Organizacja eksperymentu symulacyjnego, wyniki, analiza wyników - III etap

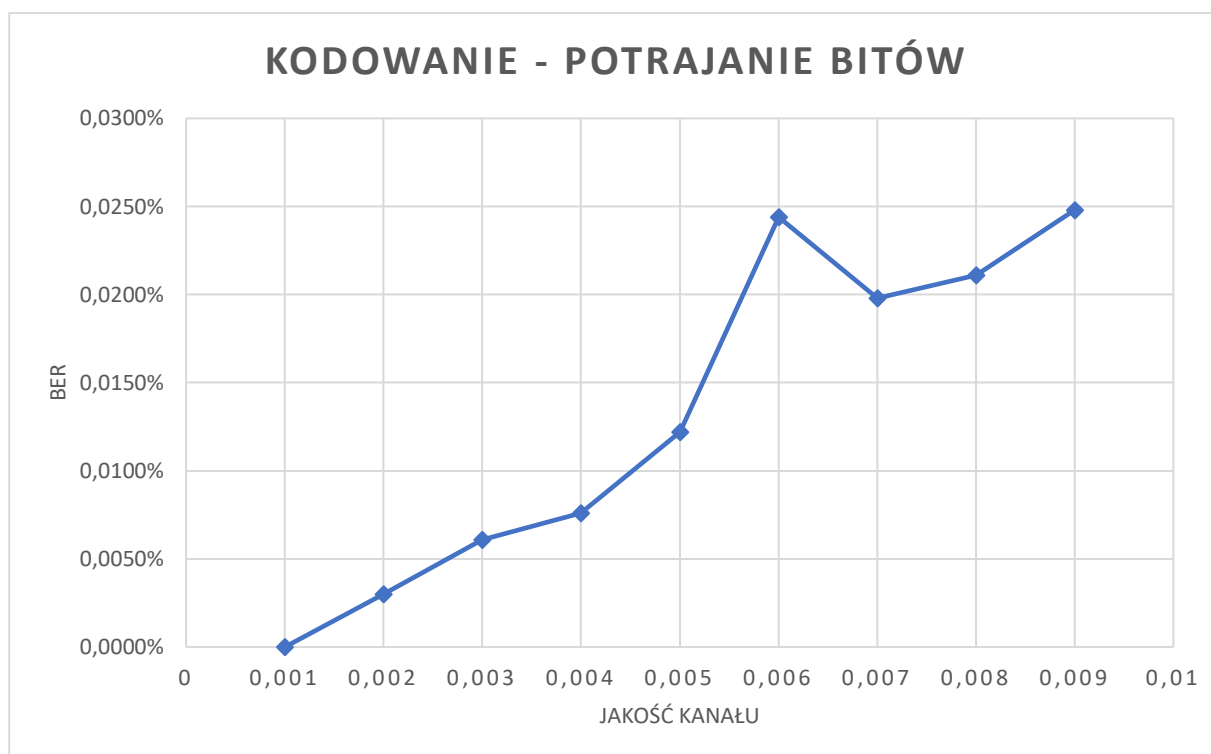
Badanie skuteczności kodów korekcyjnych i ich kosztu związanego z nadmiarem kodowym

BER w zależności od jakości kanału

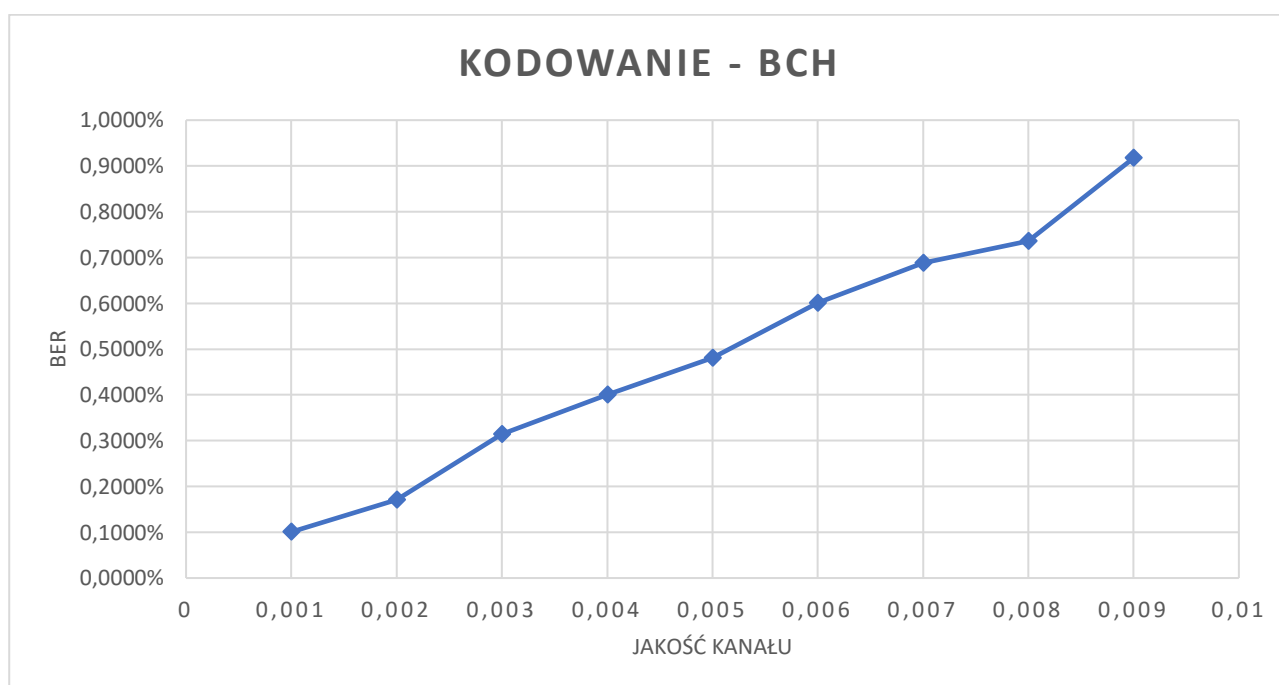
W symulatorze, który zaimplementowaliśmy w poprzednim etapie, przeprowadziliśmy badanie dotyczące BER w zależności od jakości kanału. Wyniki dla obu metod kodowania zostały przedstawione poniżej.

Kodowanie - potrajanie bitów		
Ilość bitów	Jakość kanału	BER
65519	0,001	0,0000%
	0,002	0,0030%
	0,003	0,0061%
	0,004	0,0076%
	0,005	0,0122%
	0,006	0,0244%
	0,007	0,0198%
	0,008	0,0211%
	0,009	0,0248%

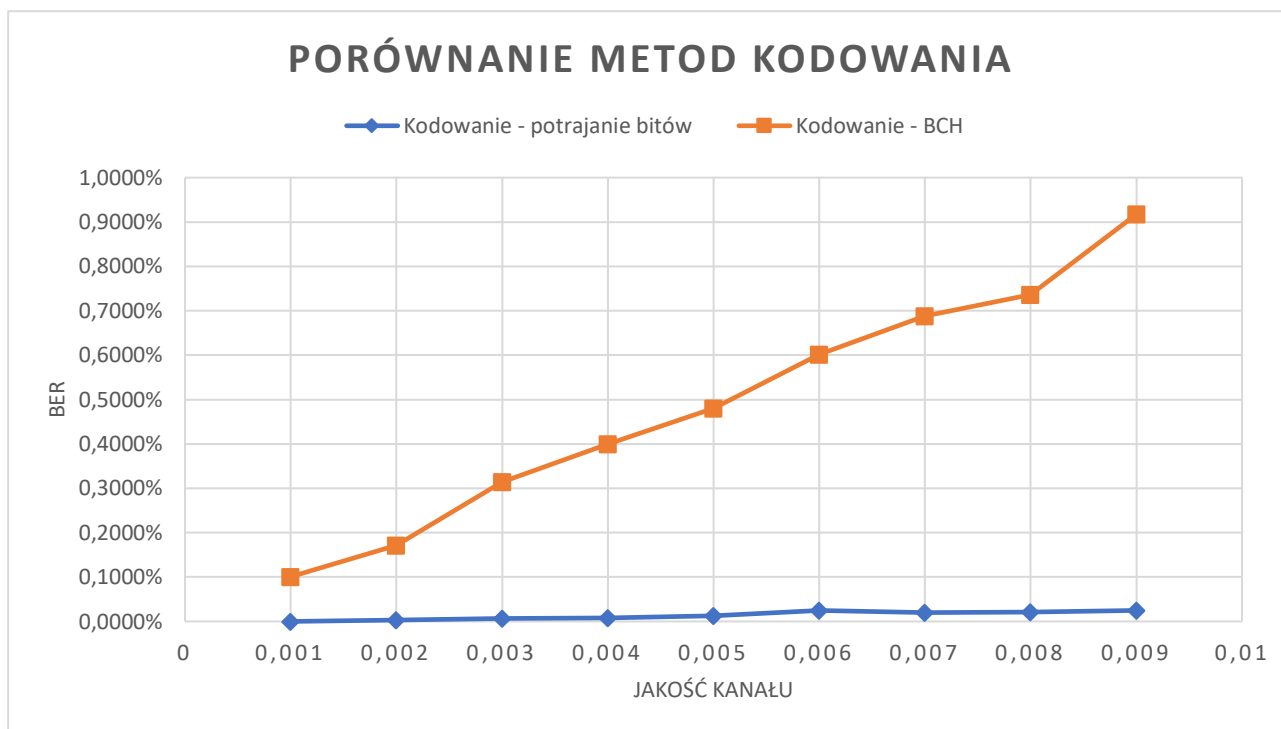
Kodowanie - BCH		
Ilość bitów	Jakość kanału	BER
65519	0,001	0,1007%
	0,002	0,1709%
	0,003	0,3144%
	0,004	0,3999%
	0,005	0,4808%
	0,006	0,6014%
	0,007	0,6883%
	0,008	0,7357%
	0,009	0,9173%



Wykres 1: BER w zależności od jakości kanału dla kodowania - potrajanie bitów



Wykres 2: BER w zależności od jakości kanału dla kodowania - BCH



Wykres 3: BER w zależności od jakości kanału – porównanie metod kodowania

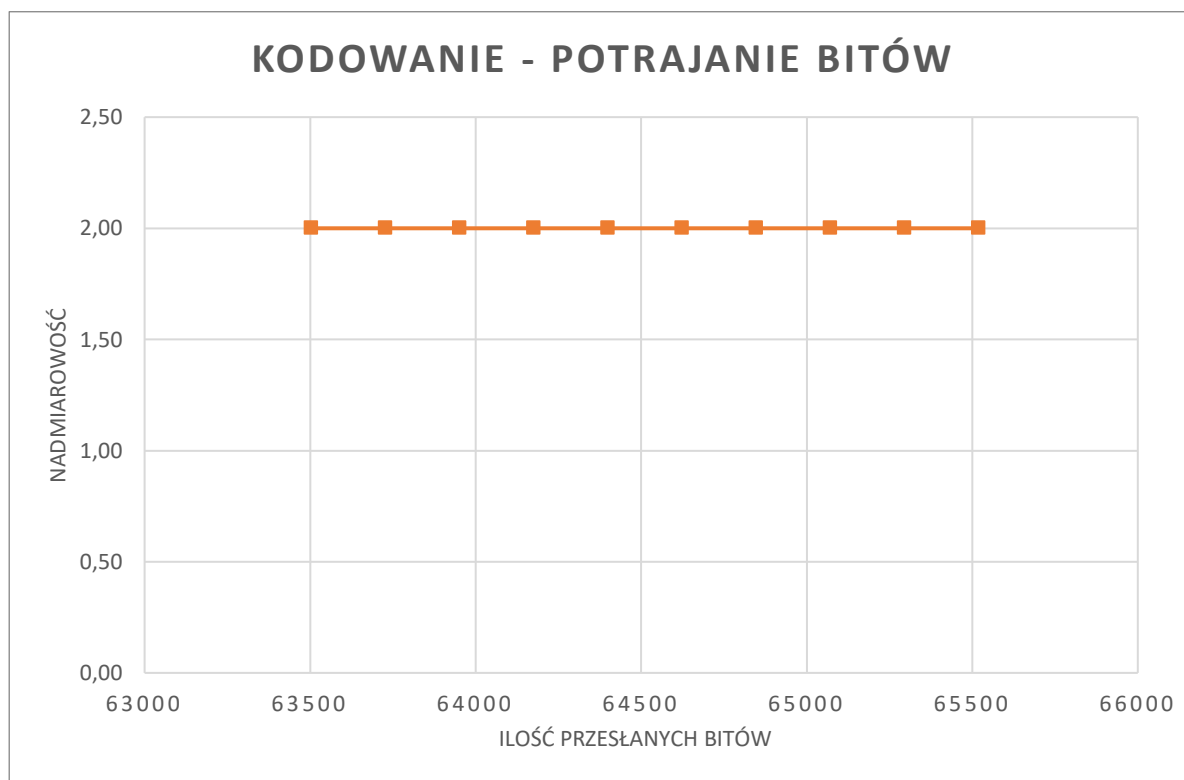
Jak widać z powyższych wyników przeprowadzonych badań, w przypadku metody kodowania BCH, wskaźnik BER jest o wiele wyższy. Wynika to głównie z powodu, że BCH ma o wiele mniejszą nadmiarowość przy transmisji niż potrajanie każdego bitu. Mając do wyboru tylko jedną z powyższych metod kodowania, należy zastanowić się co jest ważniejsze. Czy niezawodność transmisji kosztem ilości przesyłanych danych, czy też o wiele mniejszy narzut informacyjny kosztem trochę mniejszej niezawodności.

Nadmiarowość w zależności od liczby przesłanych bitów

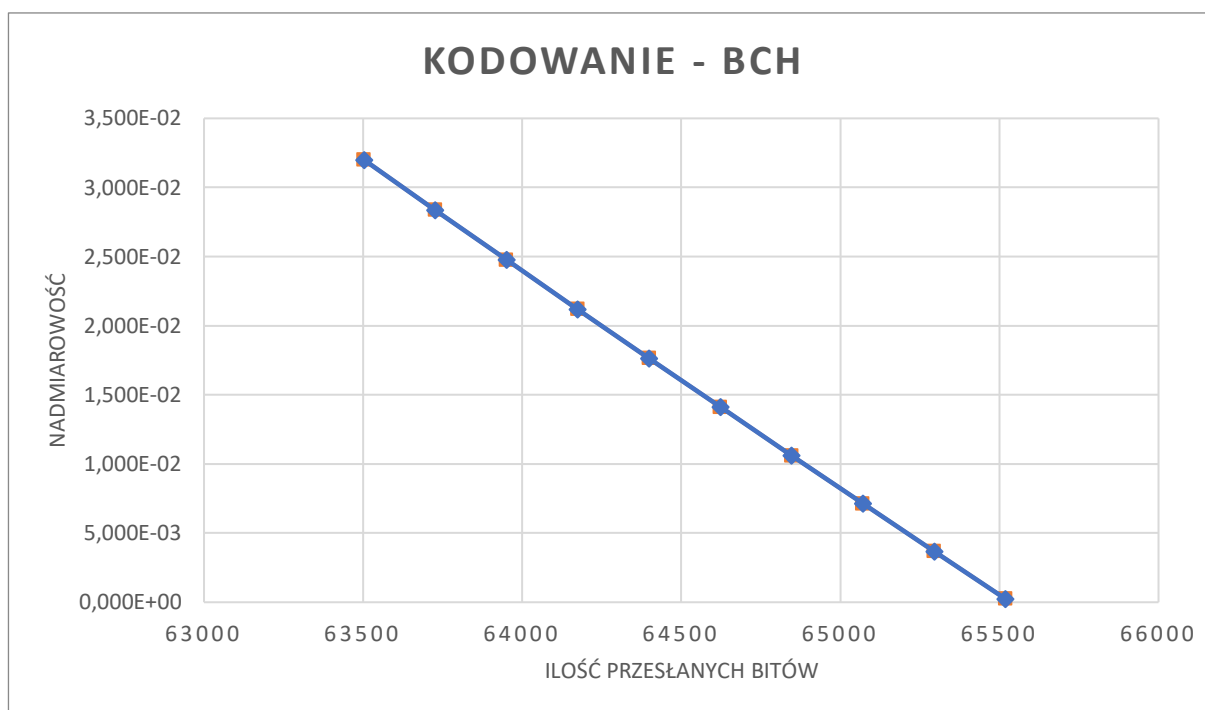
W symulatorze, który zaimplementowaliśmy w poprzednim etapie, przeprowadziliśmy badanie dotyczące nadmiarowości - nadmiaru kodowego w zależności od jakości kanału. Wyniki dla obu metod kodowania zostały przedstawione poniżej.

Kodowanie - potrajanie bitów	
Ilość przesłanych bitów	Nadmiar kodowy
65519	2,00
65295	2,00
65071	2,00
64847	2,00
64623	2,00
64399	2,00
64175	2,00
63951	2,00
63727	2,00

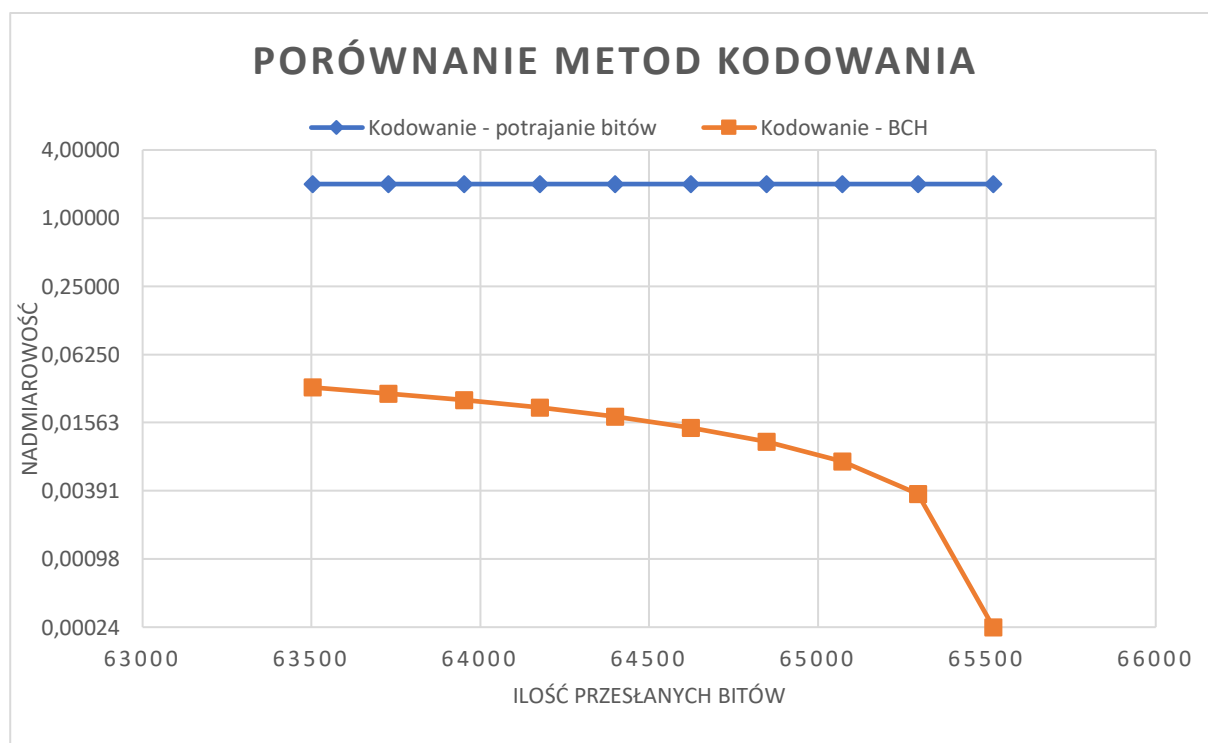
Kodowanie - BCH	
Ilość przesłanych bitów	Nadmiar kodowy
65519	0,000244204
65295	0,003675626
65071	0,007130673
64847	0,010609589
64623	0,014112622
64399	0,017640026
64175	0,021192053
63951	0,024768964
63727	0,028371020



Wykres 4: Nadmiarowość w zależności od ilości przesłanych bitów dla kodowania - potrajanie bitów



Wykres 5: Nadmiarowość w zależności od ilości przesłanych bitów dla kodowania - BCH



Wykres 6: Nadmiarowość w zależności od ilości przesłanych bitów dla kodowania - porównanie metod kodowania
Dla dobrego porównania wartości zastosowano skalę logarymiczną.

Jak widać z powyższych wyników przeprowadzonych badań, w przypadku metody kodowania BCH, nadmiarowość jest o wiele niższa i maleje ze wzrostem liczby przesłanych bitów. Dla potrajania bitów stosunek dodanych bitów do ich pierwotnej ilości jest zawsze stały i wynosi dwa. Dla kodu BCH zastosowano podane parametry: $m=16$, $n=2^{16} - 1$, k = ilość przesłanych bitów.

Wnioski

Projekt pozwolił nam zapoznać się z działaniem kodowania korekcyjnego. Przeprowadzenie pomiarów oraz ich analiza umożliwiła wyciągnięcie wniosków o dwóch sposobach kodowania, które zostały zaimplementowane w projekcie. W przypadku metody kodowania BCH, wskaźnik BER jest o wiele wyższy. Z drugiej strony w kodowaniu BCH nadmiarowość jest o wiele niższa i maleje ze wzrostem liczby przesłanych bitów. Oczywiście dla potrajania bitów ten stosunek jest stały i wynosi zawsze dwa. Sprawia to, że przy wyborze odpowiedniej metody kodowania należy zastanowić się co jest ważniejsze i bardziej opłacalne. Są sytuacje, w których o wiele bardziej istotniejsza jest niezawodność transmisji, nawet kosztem ilości przesyłanych danych. Jednak gdy występują ograniczenia związane z narzutem informacyjnym warto rozważyć kodowanie BCH, mimo trochę mniejszej niezawodności.

Projekt również przybliżył nam budowę oraz sposób implementacji symulatorów w środowisku Matlab. Zbudowanie środowiska symulacyjnego wymagało zapoznania się z biblioteką Matlaba oraz odpowiednią składnią stosowaną w tym języku.