Transmisja w systemie FEC

Dokumentacja projektu

Rok akademicki 2019/2020

Prowadzący: *dr hab. inż. Henryk Maciejewski*

Skład grupy projektowej:

Przemysław Rychter 248820
Mirosław Kuźniar 248870
Bartosz Rudnik 248893

Spis treści

1 Wstęp

1.1 Założenia projektu

Celem niniejszego projektu jest analiza transmisji w systemie FEC w kontekście trzech kodów korekcyjnych:

- Kodu BCH
- Kodu Hamminga
- Kodu powtórzeniowego

1.2 Technika FEC

FEC (*ang. Forward Error Correction*) jest techniką dodawania nadmiarowości do transmitowanych cyfrowo informacji. Umożliwia całkowitą lub częściową detekcję i korekcję błędów powstałych w wyniku zakłóceń. Dzięki temu nie ma potrzeby wykorzystywania kanału zwrotnego, do poinformowania nadawcy o błędzie i konieczności ponownego przesłania informacji. Kodowanie korekcyjne jest więc wykorzystywane wtedy, gdy retransmisja jest kosztowna, kłopotliwa lub niemożliwa, np. ze względu na ograniczenia czasowe.

2 Kody korekcyjne

2.1 Kod BCH

Kody BCH (Bose-Chaudhuri-Hocquenghema) wynalezione zostały przez Alexisa Hocquenghema oraz Raj Bose i D.K. Ray-Chaudhuriego. Jedną z najważniejszych cech kodów BCH jest możliwość precyzyjnego sterowania ich zdolnościami korekcyjnymi. Zdolność ta umożliwia naprawę wielu źle odebranych bitów danych. Do dekodowania kodów BCH można użyć algebraicznej metody syndromów. Kody BCH są szeroko stosowane w łączności satelitarnej, sterownikach dysków, odtwarzaczach CD i DVD oraz w dwuwymiarowych kodach paskowych. Wśród kodów BCH największe znaczenie odgrywają kody binarne, dla których zachodzi własność mówiąca, że dla każdej liczby całkowitej m i t < 2m-1 będzie istniał kod BCH, którego długość będzie równa $n=2m-1$. Kod ten będzie miał zdolność do korekcji t błędnie otrzymanych bitów.

m	n	k	t	m	n	k	t	m	n	k	t	m	n	k	t
3	7	4	1	6	63	10	13	7	127	15	27	8	255	123	19
4	15	11	1			7	15			8	31			115	21
		7	2	7	127	120	1	8	255	247	1			107	22
		5	3			113	2			239	2			99	23
5	31	26	1			106	3			231	3			91	25
		21	2			99	4			223	4			87	26
		16	3			92	5			215	5			79	27
		11	5			85	6			207	6			71	29
		6	7			78	7			199	7			63	30
6	63	57	1			71	9			191	8			55	31
		51	2			64	10			187	9			47	42
		45	3			57	11			179	10			45	43
		39	4			50	13			171	11			37	45
		36	5			43	14			163	12			29	47
		30	6			36	15			155	13			21	55
		24	7			29	21			147	14			13	59
		18	10			22	23			139	15			9	63
		16	11							131	18				

2.2 Kod Hamminga

Kod Hamminga to blokowy, liniowy kod korekcyjny wynaleziony przez Richarda Hamminga. Jego zdolność korekcyjna wynosi t = 1, z tego powodu w celu uzyskania niezawodnej transmisji całą wiadomość należy podzielić na słowa (zakodowane kodem Hamminga), które po przejściu przez kanał transmisyjny uzyskują przeważnie przekłamanie tylko jednego bitu – maksymalna oczekiwana odległość Hamminga między słowem transmitowanym a odbieranym równa 1.

Dla każdej liczby całkowitej $p \geq 3$ istnieje $(n, k) = (2^m - 1, 2^m - m - 1)$ kod Hamminga

m – liczba pozycji kontrolnych
$2^m - 1$ - długość zakodowanego słowa
$2^m - m - 1$ - ilość bitów informacji w słowie

Wszystkie takie kody Hamminga mają minimalną odległość Hamminga pomiędzy wektorami kodowymi $d_{min} = 3$. Zgodnie ze wzorem na zdolnosć detekcyjną $zd = d_{min} - 1$ wszystkie wykrywają do 2 bitów przekłamań w słowie, oraz mają zdolność korekcyjną $t = zakrągli\ do\ mniejszej\ całowitej\left(\frac{d_{min}-1}{2}\right)$ czyli $t = 1$, ponieważ przy jednym przekłamanym bicie, jest możliwość jednoznacznie określenia wiadomości początkowej ale już przy dwóch błędnych bitach, najmniejsza odległość Hamminga pomiędzy otrzymaną wiadomością a danym wektorem kodowym będzie wskazywała niepoprawny wektor kodowy.

Na przykład dla m= 3 pozycji kontrolnych otrzymamy kod (7,4) w którym zakodowane słowo będzie zawierać 7 bitów, z których 4 będą wiadomością.

GENERALNY ALGORYTM KODOWANIA

Przyjmijmy, że bity parzystości znajdują się na pozycjach będących potęgami 2. Algorytm jest następujący:

1. wszystkie pozycje będące potęgami 2 (1, 2, 4, 8, 16,...) są bitami parzystości,
2. wszystkie pozycje niebędące potęgami 2 (3, 5, 6, 7, 9, 10,...) to bity informacyjne,
3. każdy bit parzystości wskazuje parzystość pewnej grupy bitów w słowie, a jego pozycja określa, które bity ma sprawdzać, a które opuszczać:

- pozycja 1: opuszcza 0 bitów, sprawdza 1 bit, opuszcza 1 bit, sprawdza 1 bit, opuszcza 1 bit itd. (1, 3, 5, 7, 9,...),
- pozycja 2: opuszcza 1 bit, sprawdza 2 bity, opuszcza 2 bity sprawdza 2 bity, opuszcza 2 bity itd. (2, 3, 6, 7, 10, 11,...),
- pozycja 4: opuszcza 3 bity, sprawdza 4 bity, opuszcza 4 bity sprawdza 4 bity, opuszcza 4 bity itd. (4, 5, 6, 7, 12, 13, 14, 15,...)
- pozycja 8: opuszcza 7 bitów, sprawdza 8 bitów, opuszcza 8 bitów sprawdza 8 bitów, opuszcza 8 bitów itd.,

  ...
- pozycja n: opuszcza n-1 bitów, sprawdza n bitów, opuszcza n bitów, sprawdza n bitów itd.

Kodowanie Hamminga to po prostu użycie dodatkowych bitów parzystości, które pozwolą skorygować błąd. W 7-bitowej wiadomości jest możliwe 7 przekłamań pojedynczego bitu, dlatego 3 bity kontrolne pozwalają skorygować $2^3 > 7$ błedów.

Zaprezentujmy przykładowe kodowanie dla kodu Hamminga(15,11)

Bity parzystości są umiejscowione na pozycjach odpowiadających kolejnym potęgą 2. Kodowanie należy dobrać tak aby bity parzystości nie zależały od siebie. Kodując według podanego algorytmu nieprawidłowe bity parzystości utworzą liczbę (syndrom – wektor) odpowiadającą pozycji przekłamanego bitu. $I = (I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9, I_{10}, I_{11})$ to wiadomość, P_1, P_2, P_3, P_4 to bity parzystości dla bitów oznaczonych gwiazdką w danym wierszu.

	P_1	P_2	I_1	P_3	I_2	I_3	I_4	P_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
P_1	*		*		*		*		*		*		*		*
P_2		*	*			*	*			*	*			*	*
P_3				*	*	*	*					*	*	*	*
P_4								*	*	*	*	*	*	*	*

P_1 *to bit parzystości* dla I_k $k = 1,3,5,7,$ P_2 dla I_k $k = 1,3,5,7,9,11,13,15$

P_3 dla I_k $k = 4,5,6,7,12,13,14,15$ P_4 dla I_k $k = 8,9,10,11,12,13,14,15$

Gwiazdki w danym wierszu oznaczają numery bitów dla których jest obliczany dany bit parzystości. Jak widać każdy bit wiadomości ma unikalną kombinacje bitów parzystości którego biorą go pod uwagę przy obliczaniu bitu parzystości oraz tych które go nie uwzględniają. Jeżeli na przykład jeden bit na danej pozycji byłby przekłamany to informacja uzyskana z obliczenia w dekoderze które, z 4 ciągów bitów są parzyste(lub nie), jednoznacznie zidentyfikuje miejsce przekłamania. Na przykład dla przekłamania I_7 pierwsze 3 ciągi bitów będą nieparzyste(lub parzysty jeżeli bit parzystości prawidłowo będzie tworzył ciągi nieparzyste), a 4 będzie parzysty(lub nieparzysty ...). Proces kodowania można zapisać jak mnożenie wektora danych I z macierzą generacji G wynikającą z powyższej tabeli. $w = I * G$. Proces detekcji błedu (sprawdzenia parzystości) jako mnożenie zakodowanego słowa z macierzą H^T, a nastepnie poddaniu jej operacji mod 2 $syndrom = mod2(w * H^T)$ $G * H^T = 0$ syndrom to wektor wskazujący czy(i gdzie) wystąpił błąd, w przypadku braku przekłamania powinien być wektorem zerowym. Proces korekcji jak korekcje bitu na który wskazuje syndrom (Jeśli kodowanie jest zaprojektowane tak aby syndrom wskazywał pozycje bitu przekłamanego). Proces Dekodowania jako pomnożenie przez macierz R której zadaniem będzie „wyłuskanie" bitów informacji ze słowa. Macierz R będzie miała kolumny równe 0 na pozycji bitów parzystości, a reszta kolumn będzie miała jedną 1 odpowiadającą danej pozycji wiadomości.

Naszym zadaniem projektowym nie jest implementacja koderów i dekoderów dlatego do kodowania oraz dekodowania użyjemy narzędzi dostępnych w Matlabie – „Communications Toolbox"

2.3 Kod powtórzeniowy

Kod powtórzeniowy (ang. Repetition Code) jest szczególnym przypadkiem kodu Hamminga. Jego dokładniejsza charakterystyka została zawarta w opisie kodu Hamminga. W tej części zostaną omówione tylko jego najważniejsze cechy.

Założeniem tego kodu jest powielanie bitu. W rozważanym przypadku jest to powielanie 3 – krotne tzn. przykładowo bit 1 zostanie przesłany jako 111. Celem jest dodanie informacji nadmiarowej w celu niwelacji efektów zakłóceń w kanale transmisyjnym.

Wiadomość	**Słowo kodowe**
0	000
1	111

Proces przywracania słowa kodowego do wysłanej wiadomości jest oparty na obliczaniu długości Hamminga i został podany poniżej.

Słowo kodowe	**Wiadomość**
000	0
001	
010	
100	
011	1
101	
110	
111	

3 Symulator

3.1 Kod BCH

Do realizacji skryptu w Matlabie przeprowadzającego symulacje kodu BCH wykorzystane zostały funkcje z biblioteki Communications. Do zakodowania wiadomości wykorzystana została funkcja bchenc(msg,n,k).

Parametry, które ta funkcja przyjmuje oznaczają:
* msg – wiadomość przeznaczona do zakodowania,
* n – długość kodowa słowa, wyznaczona na podstawie tabeli z parametrami kodów BCH,
* k – długość przekazanej wiadomości, wyznaczone na podstawie tabeli z parametrami kodów BCH.

Kolejnym przeprowadzonym etapem było umieszczenie w zakodowanej wiadomości losowych błędów. Do wprowadzenia zakłóceń wykorzystana została funkcja randi([imin, imax], n, p). Parametry tej funkcji oznaczają:
* [imin, imax] – przedział, z którego wylosowane liczby,
* n – ilość liczb do wylosowania.

Następnym przeprowadzonym etapem było zdekodowanie wiadomości. Zostało to wykonane przy
pomocy funkcji bchdec(noisycode, n, k). Parametry tej funkcji oznaczają:
* noisycode – jest to wiadomość z wprowadzonymi zakłóceniami,
* n – długość kodowa słowa,
* k – długość przekazanej wiadomości.

Ostatnim etapem wykonanego programu było obliczenie ilości parametru BER oznaczającego iloraz błędnie odebranych bitów w zdekodowanej wiadomości do bitów w wiadomości wysłanej. Symulacja została przeprowadzona dla różnych poziomów prawdopodobieństwa przekłamania oraz odchylenia standardowego. Dla każdego z tych poziomów zostało wykonanych 100 pomiarów. Wyniki pomiarów są wyświetlanie na koniec działania symulatora.

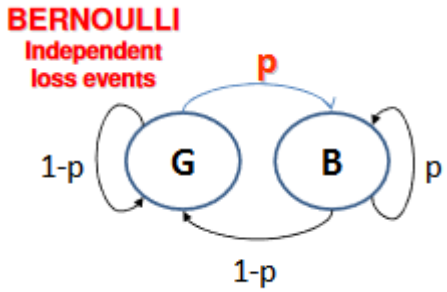
3.2 Kod Hamminga

Jest to skrypt (klasa) pozwalający na analizę statystyczną kodów korekcyjnych

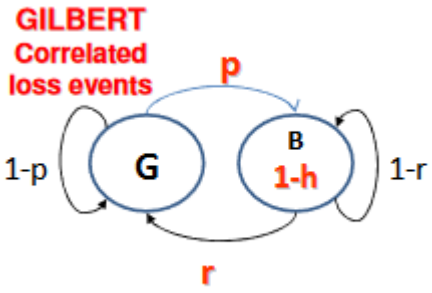
Używa, funkcji pomocniczych:

- generateData – służy do generacji wektora danych binarnych
  - Używa funkcji matlabowej randi która generuje liczby pseudolosowe z rozkładem równomiernym
- encodeHamming – służy do zakodowania danych
  - Używa matlabowej funkcji encode zauważyliśmy, że funkcja matlabowa encode używa innego kodowania niż ten podany jak główny algorytm. Bity parzystości w funkcji encode bity parzystości są zawsze na początku słowa zakodowanego
- decodeHamming – służy do odkodowania zakodowanych danych
  - Używa matlabowej funkcji decode
- bscChannel – modeluje kanał komunikacyjny - binary symmetric channel (BSC) w którym bity mogą zostać błednie odebrane w zależności od zadanego prawdopodobieństwa
  - Używa matlabowej funkcji bsc

Jest to Model Bernulliego który wprowadza pojedyncze, niezależne błędy.



- bncChannel – modeluje kanał komunikacyjny – burst-error channel którym pojawiają się błędy grupowe. Jest to dwustanowy model Gilberta parametryzowany 3 wartościami:
  - Prawdopodobieństwo przejścia ze stanu dobrego do złego - p = Good2Bad =G2B
  - Prawdopodobieństwo przejścia ze stanu złego do dobrego – r = Bad2Godd = B2G
  - Prawdopodobieństwo wystąpienie błędu w stanie złym - $1 - h$ = loss density



Założyliśmy że w stanie dobrym błędy nie występują, aby ułatwić operowanie kanałem z błędami grupowymi parametr r został wyrażony poprzez średnią długość błędu grupowego

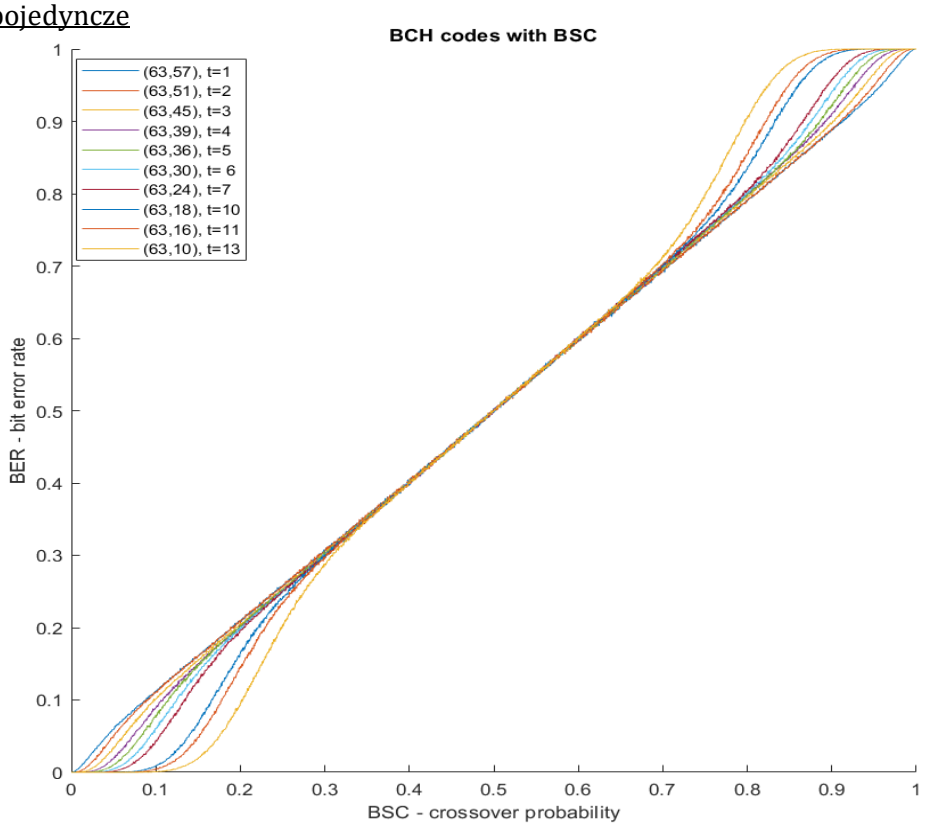
ABEL $ABEL = \frac{1}{r}$ a parameter p poprzez prawdopodobieństwo wystąpienia błędu grupowego dane wzorem $P = \frac{p}{r+p}$

4 Wyniki

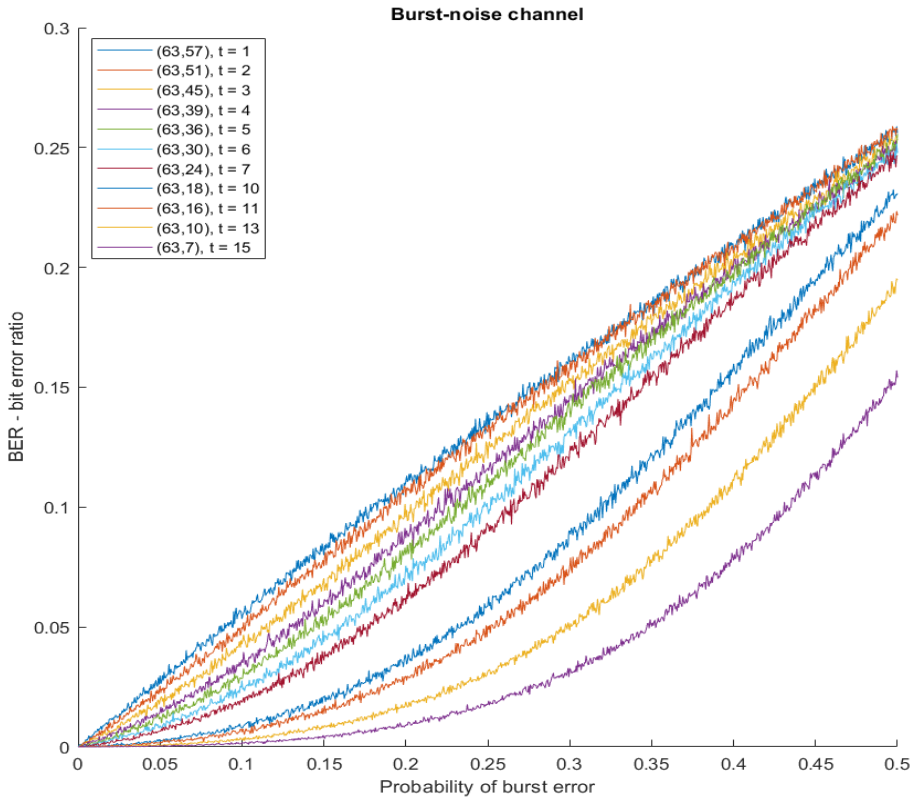
W tej sekcji zostały zgrupowane najważniejsze wykresy prezentujące uzyskane przez nas wyniki.

4.1 Zależność BER od typu błędów w kanale dla kodów BCH

4.1.1 Błędy pojedyncze

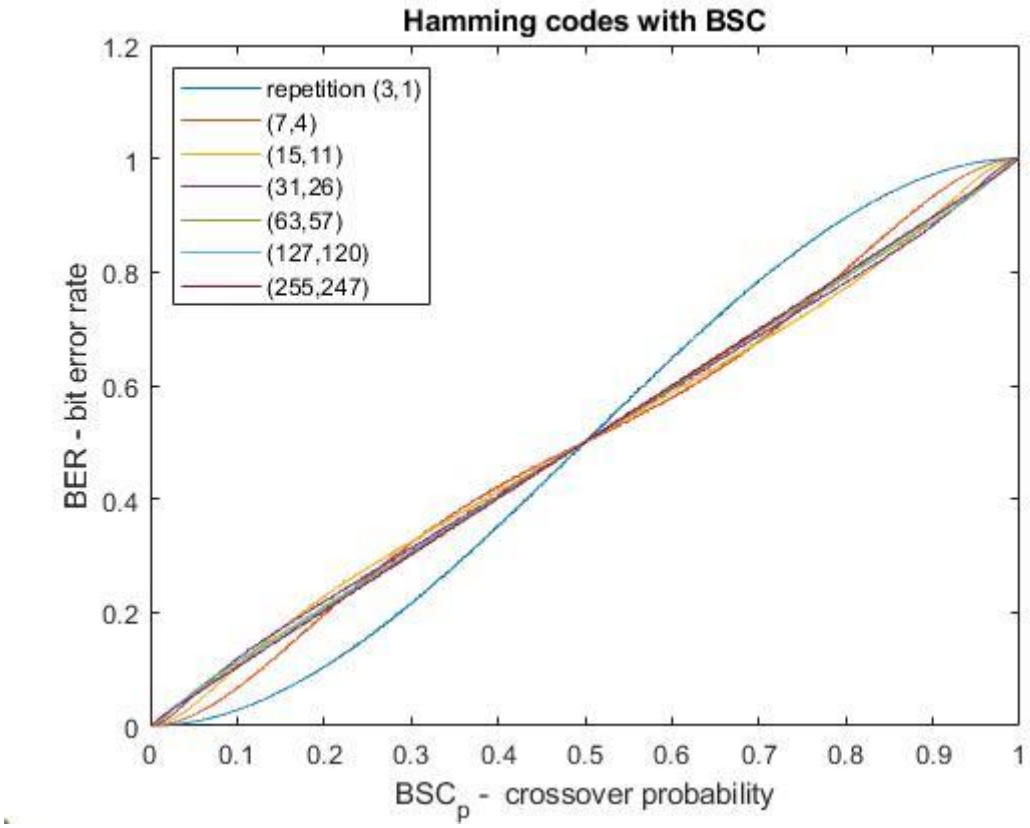


4.1.2 Błędy grupowe

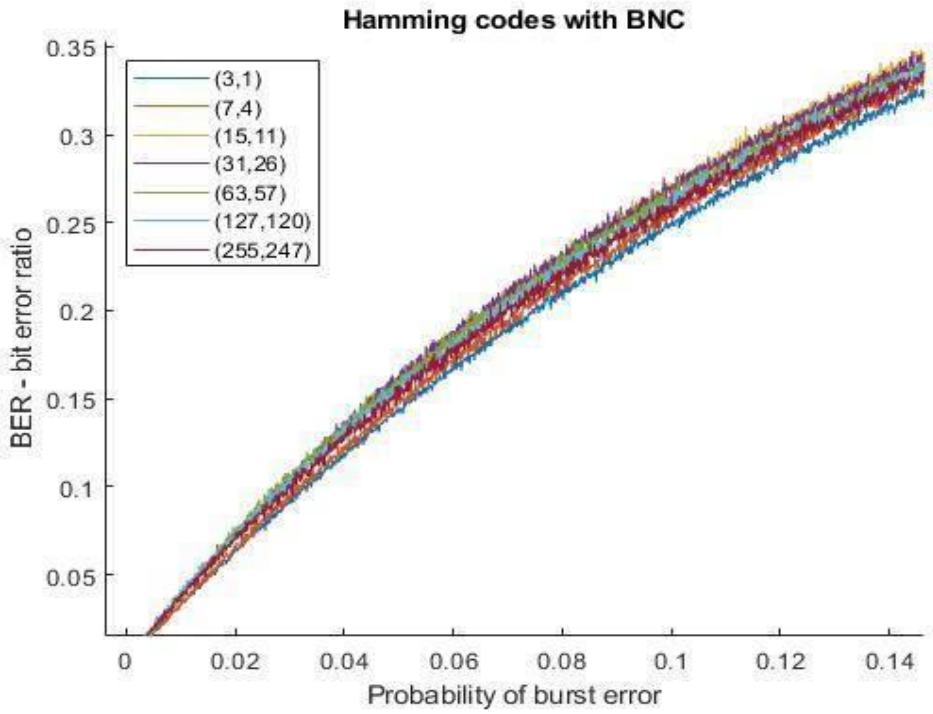


4.2 Zależność BER od typu błędów w kanale dla kodów Hamminga

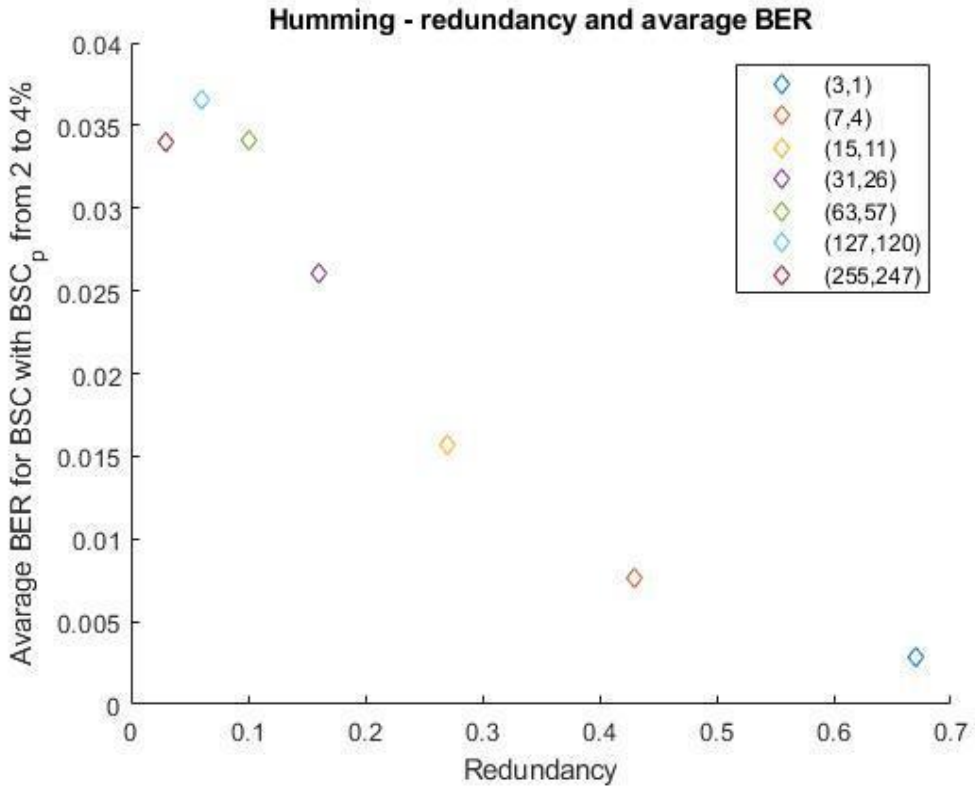
4.2.1 Błędy pojedyncze



4.2.2 Błędy grupowe



4.3 Nadmiarowość



5 Analiza wyników

Obserwując wyniki kodu Hamminga dla błędów pojedynczych zauważyć można, że kształty krzywych są do siebie zbliżone dla różnych wartości n i k. Najbardziej wyróżniającą się krzywą jest ta dla kodu o parametrach n=3, k=1 i dlatego była badana jako osobny kod - kod powtórzeniowy (ang. Repetition Code). Pozostałe krzywe grupują się w jeden zbiór. Zauważyć można, że ich charakterystyczny kształt wskazuje na ciekawe zjawisko - im gorsze właściwości korekcyjne przed prawdopodobieństwem przekłamania = 50% tym lepsze po. Ponadto pewne wartości n i k wykazują lepsze właściwości niż inne w określonych sytuacjach. Na przykład, dla kanału o niskim praw. przekłamania (jaki zazwyczaj jest używany) kombinacja n=7, k=4 zyskuje przewagę na innymi. W przypadku błędów grupowych można zauważyć, że BER zwiększa się proporcjonalnie do prawdopodobieństwa wystąpienia błędu grupowego. W tym przypadku łatwiej określić optymalne wartości n i k. Z wykresu wyraźnie wynika, że ponownie kod Hamminga (7,4) oferuje najlepsze własności korekcyjne.

W przypadku kodu powtórzeniowego (będącym szczególnym przypadkiem kodu Hamminga) zauważyć można, że w stosunku do innych kombinacji n i k jego redundancja jest największa. Sprawia to że narzut informacji jest w jego przypadku najmniej korzystny. Z drugiej jednak strony tak duża nadwyżka informacji przesyłanej wykazuje o wiele mniejsze tendencje do przekłamań, co szczególnie jest to widoczne w przypadku błędów pojedynczych dla prawdopodobieństwa przekłamania mniejszego niż 50%.

W przypadku kodu BCH został przyjęty stały parametr n = 63 i do niego dobierany był zmienny parametr k. Z powyższych wykresów zaobserwować można, że dla kanału realizującego błędy pojedyncze, dla prawdopodobieństwa przekłamania mniejszego niż 40% kody posiadające większą zdolność korekcyjną uzyskują mniejszy współczynnik BER w porównaniu do kodów, których zdolność korekcyjna jest mniejsza. Następnie możemy zauważyć, że dla prawdopodobieństwa przekłamania wynoszącego od 40% do 70% współczynnik BER dla wszystkich testowanych kodów BCH kształtuje się na tym samym poziomie. Dla prawdopodobieństwa przekłamania większego niż 70% sytuacja jest odwrotna w porównaniu do sytuacja dla prawdopodobieństwa przekłamania mniejszego niż 40% i to kody z większą zdolnością korekcyjną mają wyższy współczynnik BER od kodów z mniejszą zdolnością korekcyjną. Dla kanału realizującego błędy grupowe z powyższych wykresów możemy zaobserwować, że kody z większą zdolnością korekcyjną uzyskują współczynnik BER mniejszy od kodów z mniejszą zdolnością korekcyjną dla prawdopodobieństwa wystąpienia błędu grupowego wynoszącego od 0% do 50%.

6 Wnioski i obserwacje

Wykonane zadane projektowe przybliżyło nam obraz problemów napotykanych podczas transmisji informacji oraz zaznajomiło z jednym ze sposobów radzenia sobie z nim – techniką FEC. Ponadto poprzez dokonanie analizy przebiegu transmisji dla zrealizowanych przez nas kodów poznaliśmy wady i zalety każdego z nich.

Prace nad wybranym tematem przebiegały bez większych problemów. Największą ilość pracy poświęciliśmy na implementacje symulatora. Późniejsze etapy obejmowały rozbudowę jego możliwości oraz prowadzenie badań.

Badania te skupiały się głównie na analizowaniu wielkości BER w zależności od wybranych parametrów kanału transmisyjnego. Dzięki nim byliśmy w stanie zaobserwować jakie wartości n i k dla kodów BCH i Hamminga najlepiej zastosować w danych sytuacjach.

Wynikiem prac poświęconych w realizacji tego zadania projektowego jest lepsze zrozumienie praw rządzących transmisją informacji oraz sposobami radzenia sobie z problemami jakie są jej nieodłącznymi elementami.

7 Bibliografia

Mochnacki, Władysław. Kody korekcyjne i kryptografia. Oficyna Wydawnicza Politechniki Wrocławskiej, 1997. https://www.dbc.wroc.pl/Content/442/mochnacki_kody.pdf

Biernat, Janusz. Kodowanie i Szyfrowanie. Akademicka Oficyna Wydawnicza EXIT Andrzej Lang, 2017.

Wikipedia contributors. (2020, April 7). Hamming(7,4). In Wikipedia, The Free Encyclopedia. Retrieved 17:48, April 19, 2020

Maciejewski, H., Jarnicki, J. and Woda, M., 2017. W11 – Kody Nadmiarowe, Zastosowania W Transmisji Danych. [online] Zsk.ict.pwr.wroc.pl. Available at: <http://www.zsk.ict.pwr.wroc.pl/zsk/repository/dydaktyka/ndsc/wyklady/niezawodnosc_w11_12.pdf> [Accessed 19 April 2020].

Gaussianwaves.com. 2020. Hamming Codes – How It Works – Gaussianwaves. [online] Available at: <https://www.gaussianwaves.com/2008/05/hamming-codes-how-it-works/> [Accessed 19 April 2020].

Eduinf.waw.pl. 2020. Bit W Zastosowaniach - ECC. [online] Available at: <https://eduinf.waw.pl/inf/alg/002_struct/0010.php> [Accessed 19 April 2020].

Mathworks.com. 2020. Uniformly Distributed Pseudorandom Integers - MATLAB Randi. [online] Available at: <https://www.mathworks.com/help/matlab/ref/randi.html> [Accessed 20 April 2020].

Mathworks.com. 2020. Error Detection And Correction- MATLAB & Simulink. [online] Available at: <https://www.mathworks.com/help/comm/error-detection-and-correction.html?s_tid=CRUX_lftnav> [Accessed 20 April 2020].

Mathworks.com. 2020. Binary Symmetric Channel - MATLAB Bsc. [online] Available at: <https://www.mathworks.com/help/comm/ref/bsc.html> [Accessed 20 April 2020].

Dde.binghamton.edu. 2020. Binary Additive White-Gaussian-Noise Channel. [online] Available at: <http://dde.binghamton.edu/filler/mct/lectures/25/mct-lect25-bawgnc.pdf> [Accessed 20 April 2020].

Mathworks.com. 2020. Channel Models- MATLAB & Simulink. [online] Available at: <https://www.mathworks.com/help/comm/channel-models.html> [Accessed 20 April 2020].