



Politechnika Wrocławska

TRANSMISJA W SYSTEMIE FORWARD ERROR CORRECTION

Halszka Ciesielska 248871

Jędrzej Piątek 249453



Spis treści

1. Cel i założenia projektu.....	2
2. Opis wykonanego symulatora.....	3
1. Opis wykorzystanych kodów oraz kanałów	3
2. Opis działania symulatora	3
3. Organizacja eksperymentu symulacyjnego.....	5
4. Wyniki eksperymentu	6
5. Analiza wyników	12
1. Analiza wyników dla BSC	12
2. Analiza wyników dla modelu Gilberta.....	12
3. Analiza nadmiarowości kodów	12
6. Uwagi i wnioski	13
1. Uwagi	13
2. Wnioski	13
7. Spis tabel, wykresów i rysunków.....	14



1. Cel i założenia projektu

Celem projektu było stworzenie symulatora transmisji w systemie FEC. Technika ta polega na dodawaniu nadmiarowości do przesyłanych informacji w taki sposób, aby umożliwić częściową lub nawet całkowitą detekcję i korekcję powstałych błędów. Wykorzystywane metody korekcji błędów są jednak skomplikowane i czasochłonne oraz mogą być zawodne w przypadku dużej liczby błędów w odebranych ciągu danych. W systemie tym nie zachodzi potrzeba informowania o błędzie i przeprowadzania ponownej transmisji informacji w związku z czym wykorzystuje się go przede wszystkim wtedy, gdy ponowna transmisja danych jest niemożliwa do wykonania lub nieopłacalna. Dane przychodzą z jednakowym opóźnieniem, w związku z czym jest to idealna metoda dla systemów czasu rzeczywistego.

Środowiskiem wykonawczym dla naszego projektu był program Octave. Został on przez nas wybrany, ponieważ jest on ogólnodostępnym, darmowym środowiskiem. Ponadto w trakcie kompletowania przez nas wymagań do naszego projektu odnaleźliśmy informację o tym, że tak samo jak Matlab, posiada on pakiet *communications* zawierający potrzebne nam funkcje, takie jak:

- *bchenco* – koduje wiadomość przy użyciu kodowania BCH,
- *bchdeco* – dekoduje wiadomość zakodowaną przy użyciu kodu BCH,
- *bsc* – wysyła dane przez kanał BSC z prawdopodobieństwem P błędu.
- *rsenc* - koduje wiadomość przy użyciu kodowania Reeda-Solomona,
- *rsdec* - dekoduje wiadomość zakodowaną przy użyciu kodu Reeda-Solomona,
- *biterr* – porównuje dwie macierze i zwraca ilość błędnie przesłanych bitów oraz BER,
- *randerr* - generuje macierz losowych błędów.



2. Opis wykonanego symulatora

1. Opis wykorzystanych kodów oraz kanałów



Rysunek 1. Model systemu transmisyjnego.

W projekcie, w celu zasymulowania roli nadajnika, wykorzystane zostały trzy rodzaje kodów: kod potrojeniowy, Reeda-Solomona oraz BCH. Do pracy z kodami Reeda-Solomona oraz BCH użyte zostały funkcje pochodzące z pakietu *communications*. Funkcje kodowania oraz dekodowania z użyciem kodu potrojeniowego zostały zaimplementowane w pełni przez nas, ponieważ nie odnaleźliśmy odpowiedniego pakietu, która zawierałaby interesującą nas implementację funkcji do pracy z kodem potrojeniowym.

Zdecydowaliśmy się, że w trakcie naszych rozważań przeanalizujemy dwa modele kanałów dyskretnych: BSC (Binary Symetric Channel) oraz model Gilberta. Kanał BSC, również był częścią pakietu *communications*, w związku z czym do zasymulowania tego kanału, dla kodów potrojeniowego oraz BCH, użyliśmy funkcji *bsc*. W przypadku kodu Reeda-Solomona użycie tej funkcji sprawiało wiele problemów, w związku z czym zdecydowaliśmy się na stworzenie własnej implementacji kanału BSC poprzez użycie funkcji *randerr* i dobranie jej argumentów w taki sposób, aby jak najlepiej odwzorowywała działanie kanału BSC.

Model Gilberta został zaimplementowany w pełni przez nas, ponieważ ponownie nie odnaleźliśmy właściwej funkcji w pakiecie *communications*. W trakcie tworzenia modelu Gilberta założyliśmy, że prawdopodobieństwo wystąpienia błędu w stanie dobrym wynosi 0,001%, natomiast w stanie złym aż 70%.

2. Opis działania symulatora

Symulator działa w następujący sposób:

1. Pierwszym krokiem jest ustalenie liczności próby poprzez przypisanie odpowiedniej wartości całkowitej nieujemnej do zmiennej *iteracje*.
2. Kolejnym krokiem jest ustalenie wielkości danych poprzez przypisanie do zmiennych *quantity* (liczba wierszy w macierzy danych) oraz *range* (liczba kolumn w macierzy danych) wybranych nieujemnych liczb całkowitych. Po ustaleniu rozmiaru danych następuje ich generacja w postaci macierzy zawierającej bity, z wykorzystaniem stworzonej przez nas funkcji *bitsGenerator*. Założyliśmy, że



3. Następnym, a zarazem najważniejszym krokiem jest ustalenie parametrów kanałów oraz kodów zgodnie z następującą tabelą:

Tabela 1. Nazwy parametrów wraz z ich opisem oraz wykorzystaniem.

Nazwa parametru	Opis parametru	Wykorzystanie
m	Rząd wielkości (potrzebne do wyznaczenia n). Musi być większe od 3.	BCH
t	Zdolność korekcyjna	BCH
$probability$	Prawdopodobieństwo błędów wykorzystywane w kanale BSC dla kodów potrojeniowego oraz BCH	BSC
pg	Prawdopodobieństwo przejścia ze stanu dobrego do stanu złego wykorzystywane w modelu Gilberta	Gilbert
pb	Prawdopodobieństwo przejścia ze stanu złego do stanu dobrego wykorzystywane w modelu Gilberta	Gilbert
$tab[]$	Tablica zawierająca wartości potęg, wykorzystywane do symulacji.	Potrojeniowy

4. Parametr n zależny jest od parametru m , ustawianego ręcznie przez użytkownika i jest wyznaczany w następujący sposób $n = 2^m - 1$. Parametr *errornum*, potrzebny do zasymulowania działania kanału BSC przy kodowaniu Reeda-Solomona zależny jest od prawdopodobieństwa błędu oraz parametru n , a wyznaczany jest w sposób następujący $errornum = \text{floor}(n * probability)$. W końcu parametr *range*, oznaczający liczbę kolumn, dla kodu BCH, zależny jest od parametrów n, m oraz t , a jego wartość wyznaczana jest w następujący sposób: $range = n - t * m$. Taka wartość liczby kolumn wynika z faktu budowy tabeli BCH oraz próby ujednolicenia działania symulatora.
5. Liczba błędów obliczana jest jako średnia arytmetyczna ze wszystkich wykonanych pomiarów dla danych parametrów. Najpierw w pętli sumowana jest całkowita liczba błędów we wszystkich próbach, a następnie wartość ta dzielona jest przez liczbę prób.



3. Organizacja eksperymentu symulacyjnego

Eksperymenty dla każdego kodowania jak i rodzaju kanału przeprowadzaliśmy osobno, co daje nam łącznie 6 różnych testów (2 rodzaje kanału i 3 rodzaje kodowania). Dla każdego rodzaju kanału przyjmowaliśmy 3 różne wartości zakłóceń, dla BSC było to kolejno: 0.01, 0.001, 0.00001 jako prawdopodobieństwo błędu dla dowolnego bitu, a dla modelu Gilberta modyfikowaliśmy parametr p_g , czyli prawdopodobieństwo przejścia ze stanu dobrego do stanu złego wykorzystywane w modelu Gilberta , kolejno wartościami : 1, 0.1, 0.001

W kodowaniu BCH przeprowadzaliśmy eksperymenty dla 3 różnych wartości parametru t . Dla modelu Gilberta było to 4,8 oraz 12, a dla BSC - 6,9,12. Dla testów na BSC rząd wielkości danych to milion, a dla modelu Gilberta dziesiątki tysięcy.

Różnica wynikała z faktu, że w zaimplementowanym przez nas modelu Gilberta fragment z zakłócaniem danych wykonywał się bardzo długo. W ramach generowania danych tworzyliśmy tablicę bitów, o rozmiarach dających nam zamierzone wielkości tablic.

Oznacza to że łącznie przeprowadziliśmy eksperymenty dla 30 różnych instancji. Eksperymenty były przeprowadzane w 50 lub 100 iteracjach, w zależności od wielkości danych.

W każdej iteracji kolejno generowaliśmy losową tablicę, kodowaliśmy ją wybranym sposobem, zakłócaliśmy, rozkodowywaliśmy i porównywaliśmy ilość błędów, po przeprowadzeniu całej serii wyniki dotyczące ilości błędów uśrednialiśmy.

Do przeprowadzenia analizy wykorzystaliśmy wykresy, na których przedstawiliśmy przebiegi funkcji BER i E w zależności od prawdopodobieństwa przekłamania bitu w kanale BSC oraz prawdopodobieństwa przejścia ze stanu dobrego do stanu złego dla kanału Gilberta. Funkcje BER i E określone są w następujący sposób:

$$\text{BER} = (\text{liczba błędnie odebranych bitów danych}) / (\text{liczba odebranych bitów danych})$$

$$E = (\text{liczba poprawnie odebranych bitów danych}) / (\text{liczba wszystkich odebranych bitów})$$

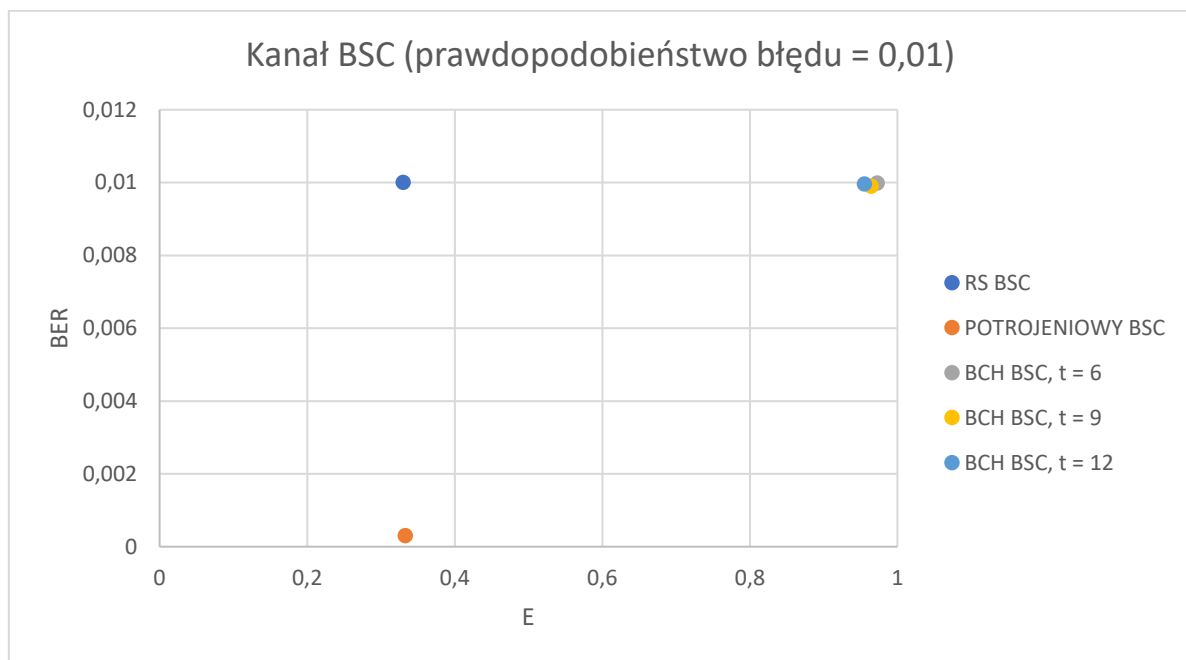
Dzięki wykorzystaniu takiej formy prezentacji wyników jesteśmy w stanie w prosty sposób zauważyć na wykresie, które pomiary dały najlepsze wyniki dla przyjętych przez nas kryteriów. W związku z tym, że w mianowniku współczynnika E znajduje się liczba wszystkich odebranych bitów, w analizie uwzględniona została przed nas również nadmiarowość kodu.



4. Wyniki eksperymentu

Tabela 2. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,01

	Reed-Solomon	Potrojeniowy	BCH, t = 6	BCH, t = 9	BCH, t = 12
E	0,32997	0,333234547	0,972599105	0,963986162	0,955222629
BER	0,010009	0,00029636	0,0099942	0,009901346	0,009962879
Nadmiarowość	16383000	2000000	21600	32400	43200
Rozmiar danych	8190500	1000000	1206900	1196100	1185300
Liczba błędów	81980	296,36	12062	11843	11809

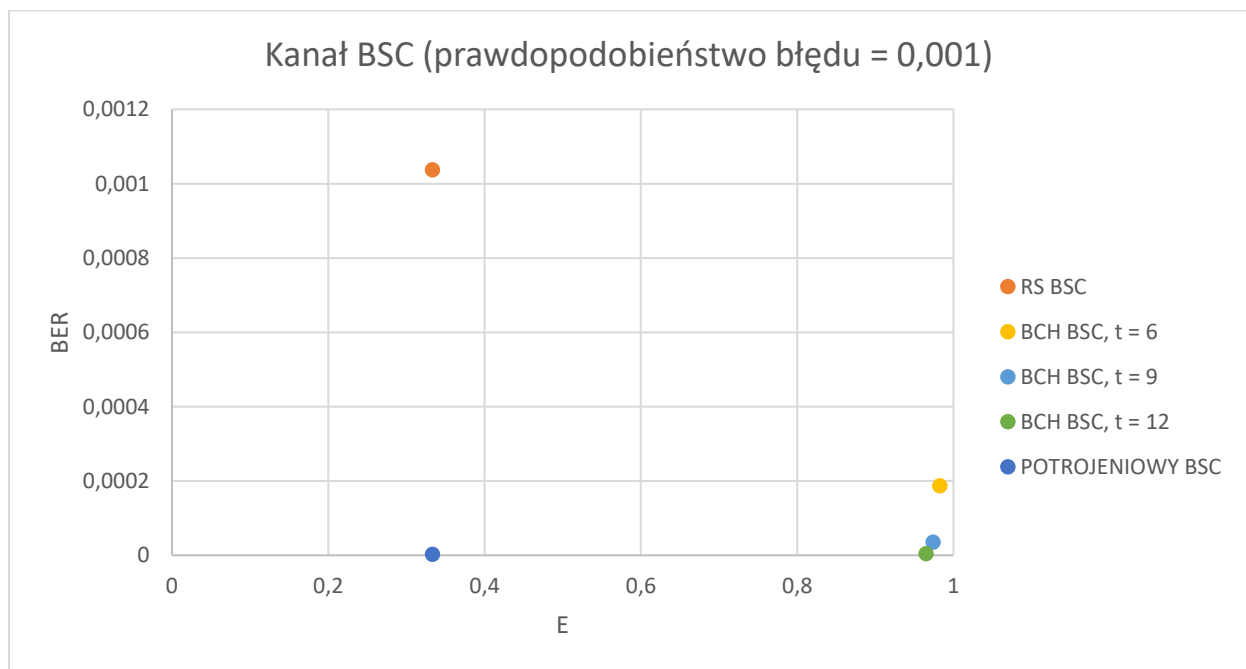


Wykres 1. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,01



Tabela 3. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,001

	Reed-Solomon	Potrojeniowy	BCH, t = 6	BCH, t = 9	BCH, t = 12
E	0,33296	0,333332317	0,982233618	0,973591372	0,964831095
BER	0,001038	0,00000305	0,000187257	3,59502E-05	4,21834E-06
Nadmiarowość	16383000	2000000	21600	32400	43200
Rozmiar danych	8190500	1000000	1206900	1196100	1185300
Liczba błędów	8498	3,05	226	43	5

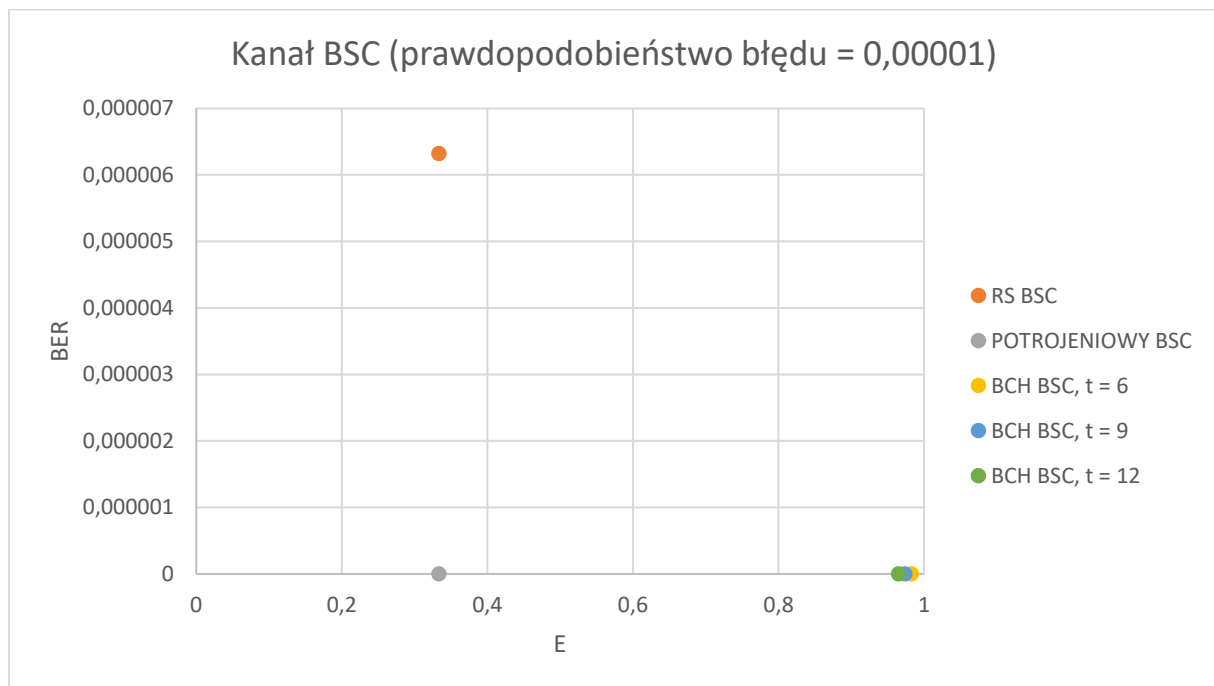


Wykres 2. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,001



Tabela 4. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,00001

	Reed-Solomon	Potrojeniowy	BCH, t = 6	BCH, t = 9	BCH, t = 12
E	0,333304	0,33333333	0,982417582	0,973626374	0,964835165
BER	6,32E-06	0	0	0	0
Nadmiarowość	16383000	2000000	21600	32400	43200
Rozmiar danych	8190500	1000000	1206900	1196100	1185300
Liczba błędów	51,76	0	0	0	0

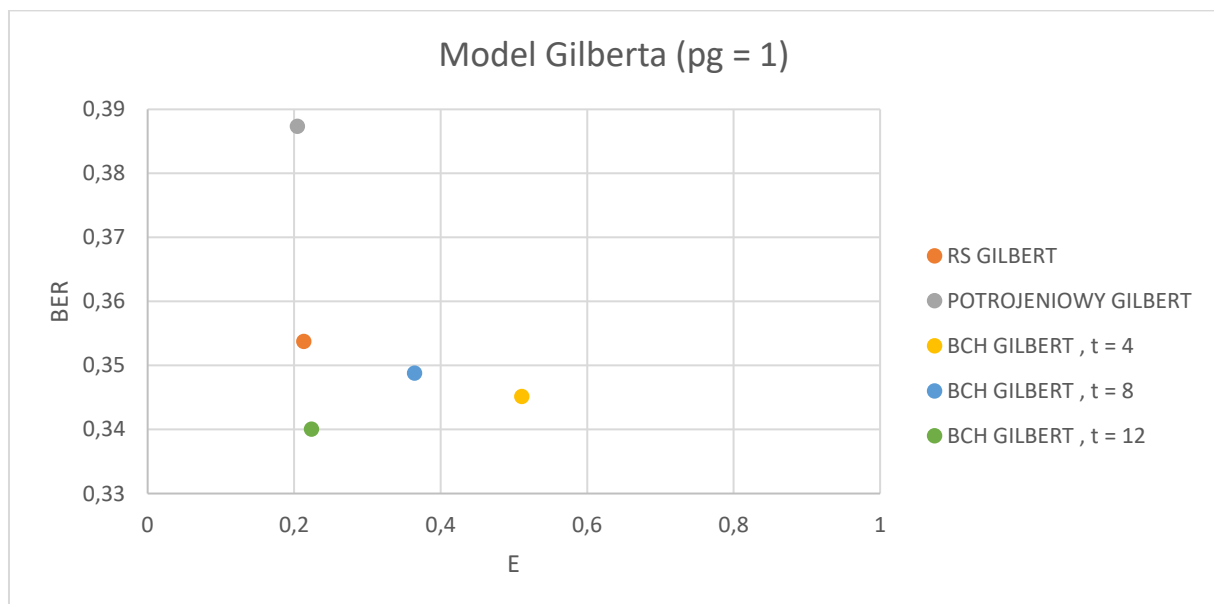


Wykres 3. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,00001



Tabela 5. Wartości dla modelu Gilberta i parametru $pg=1$

	Reed-Solomon	Potrojeniowy	BCH, $t = 6$	BCH, $t = 9$	BCH, $t = 12$
E	0,2131398	0,204213333	0,510472441	0,364055118	0,223440945
BER	0,35376	0,38736	0,345151515	0,348802817	0,340069767
Nadmiarowość	25400	25000	2800	5600	8400
Rozmiar danych	12500	12500	9900	7100	4300
Liczba błędów	4422	4842	3417	2476,5	1462,3

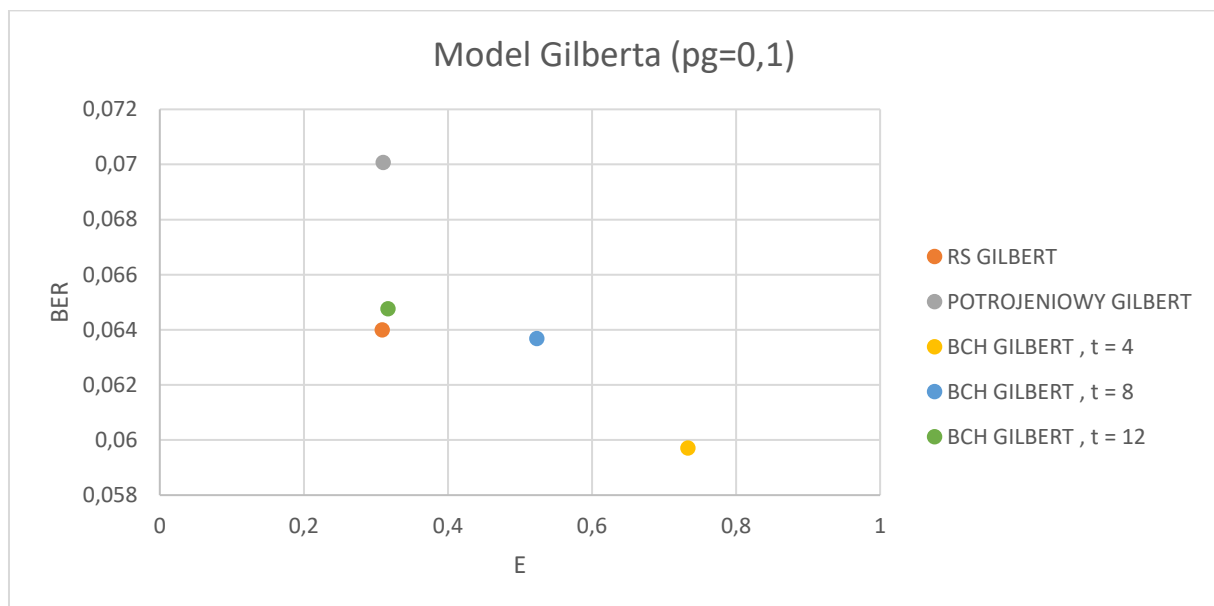


Wykres 4. Zależność BER i E dla modelu Gilberta i parametru $pg = 1$



Tabela 6. Wartości dla modelu Gilberta i parametru $p_g=0,1$

	Reed-Solomon	Potrojeniowy	BCH, $t = 6$	BCH, $t = 9$	BCH, $t = 12$
E	0,3087071	0,309973333	0,732979528	0,523448819	0,316653543
BER	0,064	0,07008	0,059713131	0,063690141	0,064767442
Nadmiarowość	25400	25000	2800	5600	8400
Rozmiar danych	12500	12500	9900	7100	4300
Liczba błędów	800	876	591,16	452,2	278,5

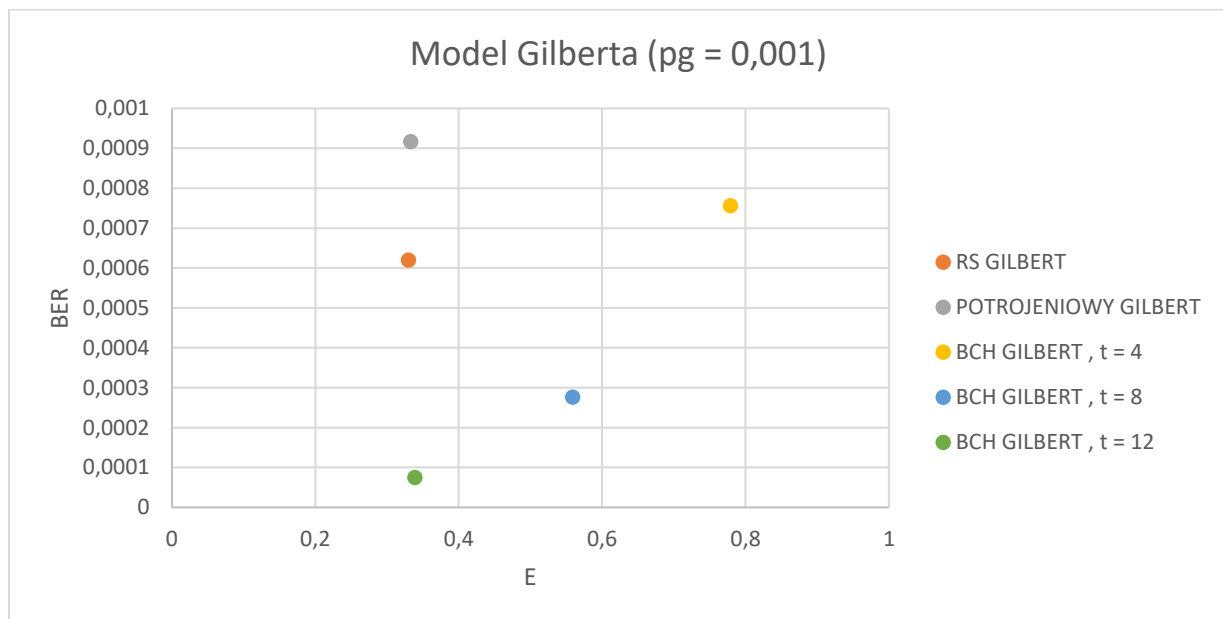


Wykres 5. Zależność BER i E dla modelu Gilberta i parametru $p_g = 0,1$



Tabela 7. Wartości dla modelu Gilberta i parametru $pg=0,001$

	Reed-Solomon	Potrojeniowy	BCH, $t = 6$	BCH, $t = 9$	BCH, $t = 12$
E	0,3296111	0,333028	0,778938583	0,558900787	0,33855748
BER	0,0006192	0,000916	0,000755556	0,000276056	7,44186E-05
Nadmiarowość	25400	25000	2800	5600	8400
Rozmiar danych	12500	12500	9900	7100	4300
Liczba błędów	7,74	11,45	7,48	1,96	0,32



Wykres 6. Zależność BER od E dla modelu Gilberta i parametru $pg = 0,001$



5. Analiza wyników

1. Analiza nadmiarowości kodów

Z badanych przez nas kodów największą nadmiarowością odznacza się zdecydowanie kodowanie Reeda-Solomona. Następnym kodem, którego nadmiarowość jest niewiele mniejsza w porównaniu z kodem Reeda-Solomona jest kod potrojeniowy. Najmniejsza nadmiarowość charakteryzuje natomiast kod BCH.

2. Analiza wyników dla BSC

W każdym przypadku, dla kanału BSC, kodowanie poprzez kod potrojeniowy dawało najmniejszą liczbę błędów, ponieważ kanał BSC charakteryzuje się powstawaniem błędów jednostkowych, z wykrywaniem których kod potrojeniowy radzi sobie bardzo dobrze, ze względu na swój algorytm głosujący. Kod potrojeniowy posiada natomiast duży nadmiar kodowy. Kodowanie poprzez BCH dawało jedyną najlepszy stosunek bitów poprawnie odebranych do ilości bitów całej transmisji (wielkość wiadomości + wielkość kodowania), ponieważ radzi ono sobie również bardzo dobrze z błędami pojedynczymi, jednak posiada znacznie mniejszą nadmiarowość od kodu BSC. Kodowanie Reeda-Solomona okazało się najmniej efektywnym - duża nadmiarowość przy nie aż tak dużej redukcji błędów.

3. Analiza wyników dla modelu Gilberta

Dla modelu Gilberta kod potrojeniowy w każdym z przypadków okazał się być najmniej efektywnym, posiadał największy wskaźnik BER, jak i najgorszy stosunek bitów odebranych poprawnie do wszystkich bitów przesłanych w transmisji. Wynika to z faktu, że po przejściu danych przez kanał w modelu Gilberta powstają błędy wielokrotne, z którymi kod potrojeniowy, ze względu na swoje działanie, nie jest sobie w stanie poradzić. Pod względem poprawności jak i efektywności przesyłania danych najskuteczniejsze okazało się kodowanie BCH, ponieważ radzi sobie ono bardzo dobrze z błędami jednostkowymi. Kodowanie Reeda-Solomona okazało się przeciętne w tym wypadku (w niektórych seriach posiadało lepszy BER od niektórych kodów BCH, jednak ogólnie nigdy nie było najlepszym możliwym kodowaniem).



6. Uwagi i wnioski

1. Uwagi

Ze względu na napotkane problemy przy wykorzystaniu dostępnej w pakiecie *communications* funkcji *bsc* przy kodowaniu Reeda-Solomona, kanał (zakłócanie przesyłanej informacji) został zasymulowany przez nas bez wykorzystania tej funkcji, jednak w sposób najbardziej zbliżony do działania klasycznego kanału BSC.

Na wykresie dla modelu Gilberta przy $p_g = 0,1$ widać lekką anomalię, gdyż pomimo zwiększania nadmiarowości, BER wzrasta zamiast spadać, na innych wykresach to nie występuje.

2. Wnioski

Kod potrojeniowy sprawdza się dobrze w przypadku BSC, gdzie pod względem BER jest najlepszy. Pod względem nadmiarowości jest już trochę gorzej szczególnie w porównaniu do kodowania BCH. Dla modelu Gilberta kod potrojeniowy jest najmniej skutecznych zarówno pod względem BER, jak i nadmiarowości, co wynika z faktu, że kod potrojeniowy nie jest stworzony do korekcji błędów wielokrotnych, jednak okazuje się bardzo skuteczny w przypadku błędów pojedynczych.

Kodowanie BCH sprawdza się dobrze zarówno w przypadku BSC, jak i modelu Gilberta. Posiada zdecydowanie najmniejszą nadmiarowość, jednocześnie uzyskując również niski BER. Dla BSC ustępuje kodowi potrojeniowemu pod względem BER dla kanałów o niskiej jakości, dla kanałów o wyższej jakości wyniki są porównywalne. Dla modelu Gilberta, uzyskuje najlepsze wyniki zarówno pod względem jakości przesyłu, jak i jego efektywności.

Kodowanie RS ma nadmiarowość na poziomie kodu potrojeniowego i zdolności korekcyjne BCH, co czyni go mało efektywnym w tym zestawieniu.



7. Spis tabel, wykresów i rysunków

Tabela 1. Nazwy parametrów wraz z ich opisem oraz wykorzystaniem.	4
Tabela 2. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,01	6
Tabela 3. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,001	7
Tabela 4. Wartości dla kanału BSC i prawdopodobieństwa błędu = 0,00001	8
Tabela 5. Wartości dla modelu Gilberta i parametru $p_g=1$	9
Tabela 6. Wartości dla modelu Gilberta i parametru $p_g=0,1$	10
Tabela 7. Wartości dla modelu Gilberta i parametru $p_g=0,001$	11
Wykres 1. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,01	6
Wykres 2. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,001	7
Wykres 3. Zależność BER i E dla kanału BSC i prawdopodobieństwa błędu = 0,00001	8
Wykres 4. Zależność BER i E dla modelu Gilberta i parametru $p_g = 1$	9
Wykres 5. Zależność BER i E dla modelu Gilberta i parametru $p_g = 0,1$	10
Wykres 6. Zależność BER od E dla modelu Gilberta i parametru $p_g = 0,001$	11
Rysunek 1. Model systemu transmisyjnego.	3