POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

Niezawodność i Diagnostyka Układów Cyfrowych

Transmisja w systemie FEC

Autorzy: Damian Marciniak
Piotr Kołeczek
Ireneusz Żwirek
Prowadzący projekt: dr inż. Jacek Jarnicki

Wrocław, 2018

Spis treści

1. Cel i założenia projektu

Celem niniejszego projektu było zapoznanie się z techniką kodowania korekcyjnego (FEC – Forward Error Correction) i napisanie oprogramowania symulującego opisany system.

1.1 Założenia projektu

Technologia FEC (Forward Error Correction) jest metodą dodawania nadmiarowego sygnału do przesyłanej w sposób cyfrowy informacji. Dzięki takiemu zastosowaniu techniki transmisyjnej możliwa jest całkowita lub częściowa detecja i korekcja błędów powstałych w wyniku zakłóceń. Dzięki temu nie ma potrzeby wykorzystywania kanału zwrotnego do poinformowania nadawcy o błędzie i konieczności ponownego przesłania informacji. Zatem technologia kodowania korekcyjnego jest wykorzystywana wtedy, gdy retransmisja jest kosztowna, bądź kłopotliwa lub nawet niemożliwa np. ze względu na ograniczenia czasowe.
Technologia transmisyjna FEC zatem opiera się na powielaniu przesyłanych bitów informacji w celu uzyskania w odbiorniku informacji zniekształconej w sposób minimalny, a nawet nienaruszony.

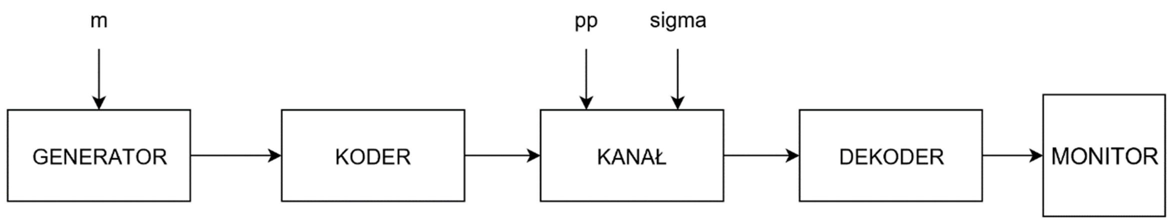
2. Etapy projektu

2.1 Etap pierwszy

Pierwszym etapem niniejszego projektu było zapoznanie się z techniką FEC oraz stworzenie odpowiedniego oprogramowania symulacyjnego w środowisku MATLAB. W tym celu zostały utworzone skrypty:

- **GENERATOR** – jego zadaniem było wygenerowanie ciągu liczb binarnych (traktowanych dalej jako bity) 0 i 1 w postaci tablicy. W tym celu posłużono się funkcją **rand(1, m)**, która wygenerowała losowe liczby rzeczywiste z przedziału [0, 1], gdzie **m** jest parametrem określającym liczbę generowanych liczb. Liczby w tablicy zaokrąglono funkcją round(), aby wartości w tablicy były całkowite (0 i 1) oraz miało to na celu uniknąć niejednoznaczności.
- **KODER** – potrajał każdy bit zawarty w tablicy, stąd jeśli w tablicy wejściowej znajdowało się N bitów, to w tablicy wyjściowej była to wartość trzykrotnie większa – 3N bitów.

- **KANAL** – ta funkcja wprowadza losowe zakłócenia w sygnale na podstawie zadanego parametru sigma (odchylenia standardowego) lub PP (prawdopodobieństwa przekłamania). W tym celu zostały stworzone dwa skrypty, po jednym dla każdego parametru. W przypadku parametru sigma wykorzystano funkcję **normrnd(0, sigma, [1, length(tab)])**, która wprowadzała losowy szum do sygnału wejściowego w oparciu o rozkład normalny Gaussa z zadanym parametrem sigma oraz długością tablicy. Drugi skrypt wprowadzał zakłócenia sygnału w oparciu o parametr pp, generując losowy szum na podstawie jego wartości.
- **DEKODER** – jego zadaniem było usunięcie zakłóceń powstałych przez przepuszczenie sygnału wejściowego przez kanał transmisyjny. Wynikiem jego działania była tablica zdekodowanych bitów. Proces dekodowania polegał na porównaniu wartości trzech kolejnych bitów potrojonych w koderze. W tablicy wynikowej zapisywano wartość bitu, który został wyznaczony na podstawie podejmowanej decyzji przy pomocy poniższego algorytmu głosującego:
  - **jeśli trójka zawiera co najmniej dwa zera, wynikiem jest zero,**
  - **jeśli trójka zawiera co najmniej dwie jedynki to wynikiem jest jeden.**
- **MONITOR** – wyświetlenie wykresów **BER(pp)** i **BER(sigma)** na podstawie wyników symulacji, gdzie parametr BER (Bit Error Rate) jest równy ilorazowi błędnie odebranych bitów do bitów wysłanych. Przykładowo: wysyłając 6 bitów w postaci 010011, a odbierając 000111, parametr BER $= \frac{2}{6} = \frac{1}{3}$ co wynika z błędów w odbiorze powstałych na bicie na pozycji 2 i 4.



Rys. 1 Schemat blokowy kroków składających się na etap 1.

2.2 Etap drugi

Kolejnym etapem było stworzenie skryptu w MATLABIE symulującego koder i dekoder kodu BCH (Bose, Chadhuri, Hocquenghem). W tym celu wykorzystano gotowe funkcje wbudowane w program MATLAB. Za zakodowanie wygenerowanej wiadomości (funkcja **gf()**) przy pomocy kodera BCH wykorzystano **bchenc(msg, n, k)** przyjmującej za parametry:

- **msg** – wygenerowana wiadomość

- **n** – długość kodowa słowa, dla zadanego parametru m (odczytanego z tabeli kodów BCH), długość słowa obliczamy ze wzoru n = 2^m - 1

- **k** – długość wiadomości (liczba bitów wiadomości w słowie kodowym)

Mając wygenerowaną i zakodowaną wiadomość następnym etapem było wprowadzenie losowych błędów do zakodowanej wiadomości poprzez użycie wbudowanej funkcji **randerr(1, n, 1 : t)**, której parametrami są:

- **n** – długość kodowa słowa

- **t** – zdolność korekcyjna (maksymalna liczba błędów, która zostanie poprawiona), wyznaczona została za pomocą funkcji **bchgenoply(n, k).**

Następnie zdekodowano powstałe słowo kodowe z błędami przy pomocy funkcji **bchdec(noisycode, n, k)**, gdzie noisycode jest słowem kodowym z wprowadzonymi losowymi błędami.

Dla zdekodowanego słowa, tworzymy wykresy BER(sigma) i BER(pp), a także wykresy E(sigma) oraz E(pp), gdzie E jest ilorazem prawidłowo odebranych bitów do wszystkich przesłanych bitów.

2.3 Etap trzeci

Kolejny etap skupiał się na utworzeniu wykresów BER(E) przy znanych parametrach PP, sigma, n i k. Celem było znalezienie **Zbioru Pareto**, dla którego parametry BER i E były optymalne. W tym celu wygenerowano w MATLABIE skrypt tworzący odpowiednie wykresy i w programie PAINT wykreślono odpowiedni zbiór punktów.

2.4 Etap czwarty

Ostatnim etapem było stworzenie niniejszego sprawozdania w celu podsumowania wykonanych działań, które pozwoliły na zapoznanie się z techniką kodowania

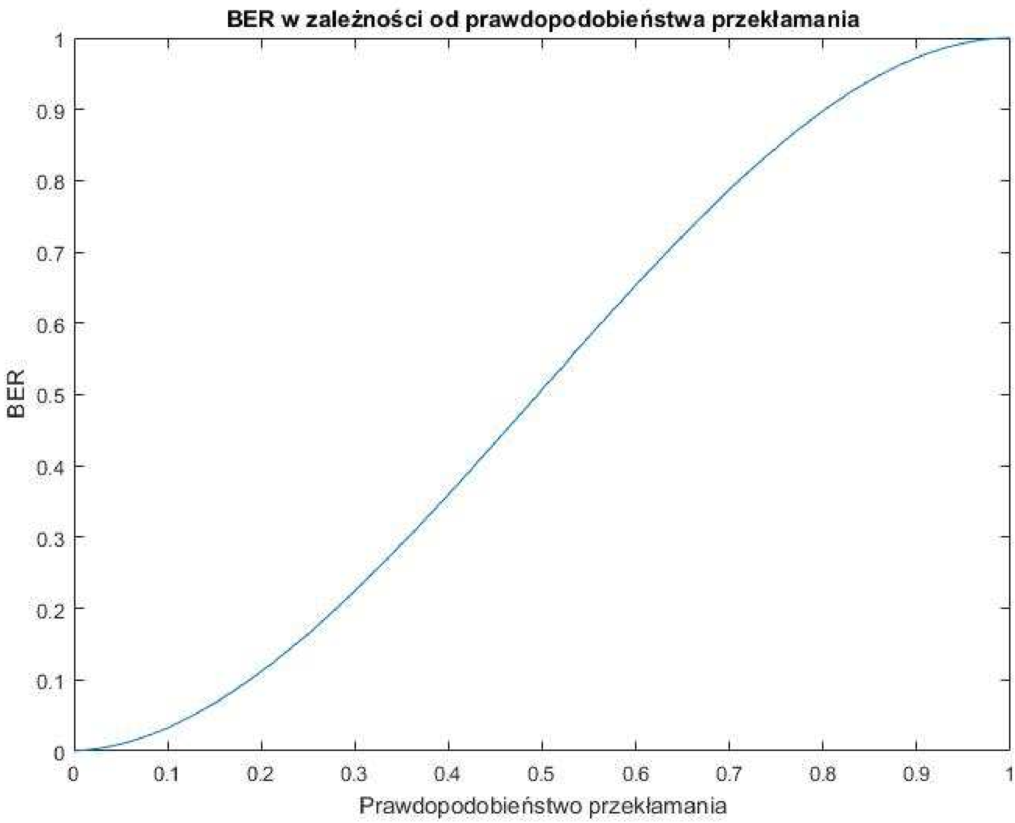
FEC, a także dały ogólne pojęcie na temat modelowania losowości i symulacji systemów transmisji informacji.

3. Wyniki eksperymentu

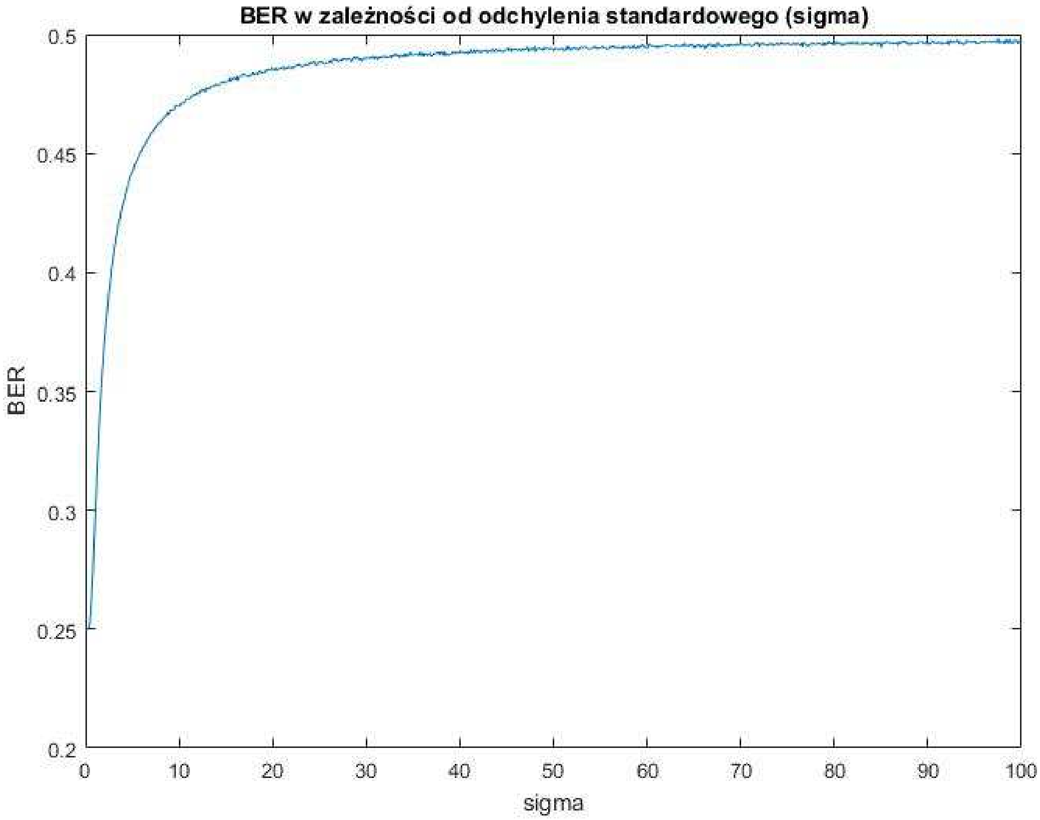
Poniższe charakterystyki są zależne od następujących parametrów:

- Sigma – odchylenie standardowe
- Prawdopodobieństwo przekłamania (pp) – określa jaka część bitów może zostać przekłamana
- BER - iloraz błędnie odebranych bitów do bitów wysłanych
- E - iloraz prawidłowo odebranych bitów do wszystkich przesłanych bitów

3.1 Etap 1 - BER(pp), BER(sigma)

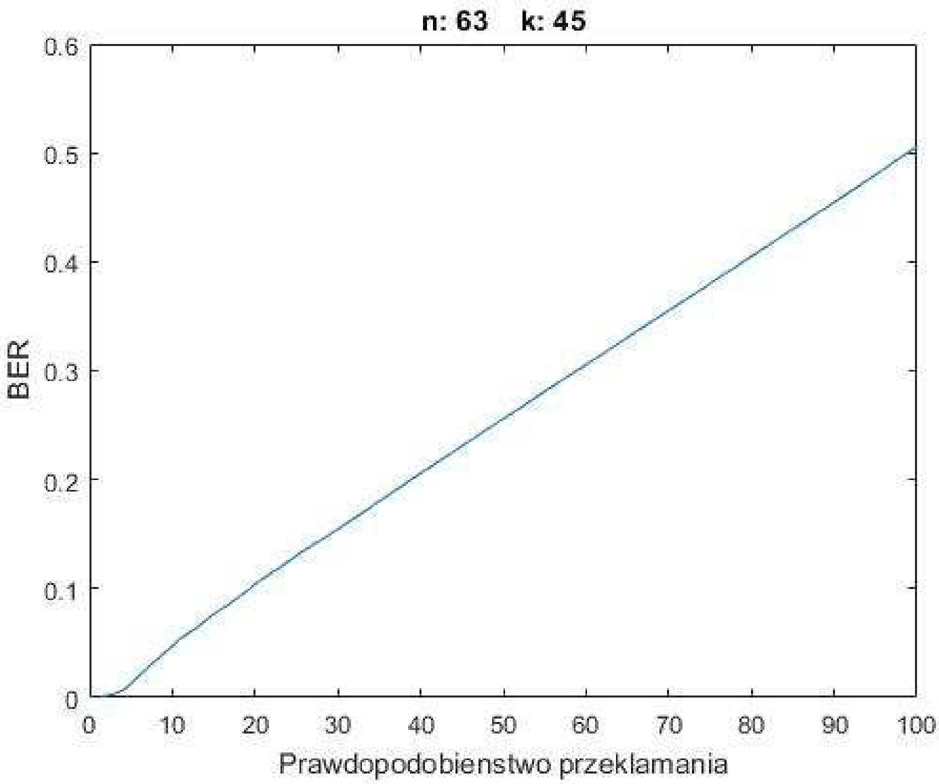


Rys. 2 Wykres Bit Error Rate w zależności od prawdopodobieństwa przekłamania.

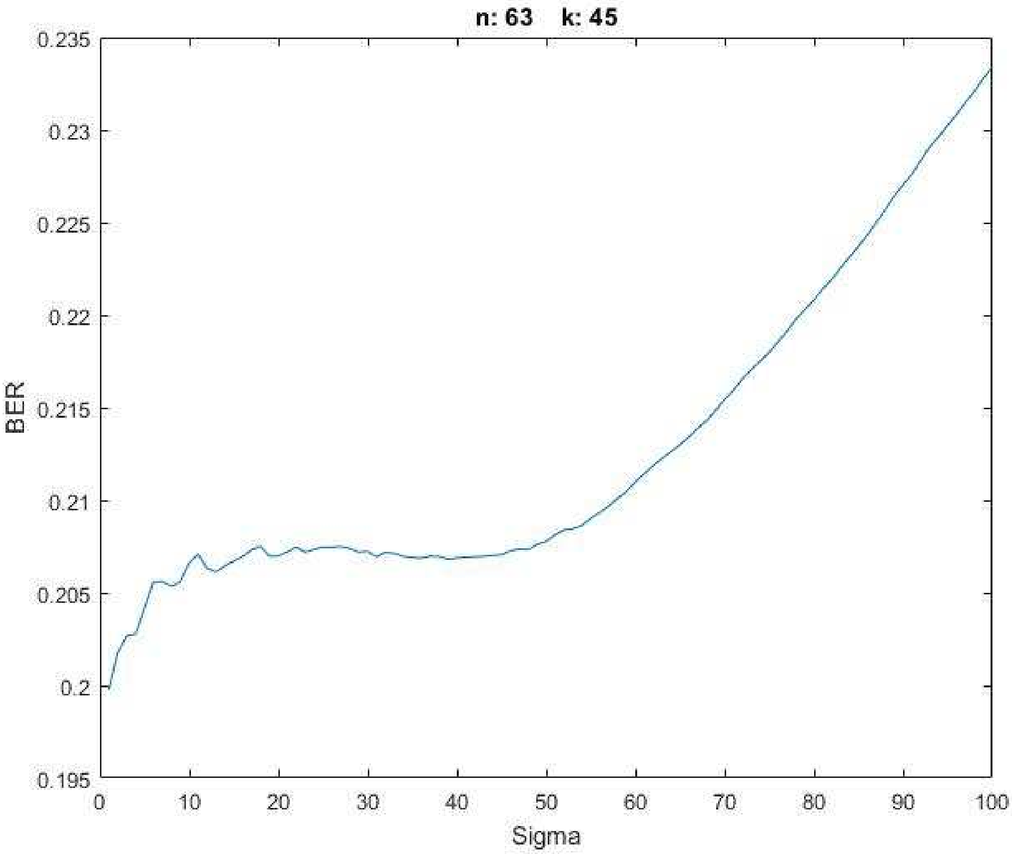


Rys. 3 Wykres Bit Error Rate w zależności od odchylenia standardowego.

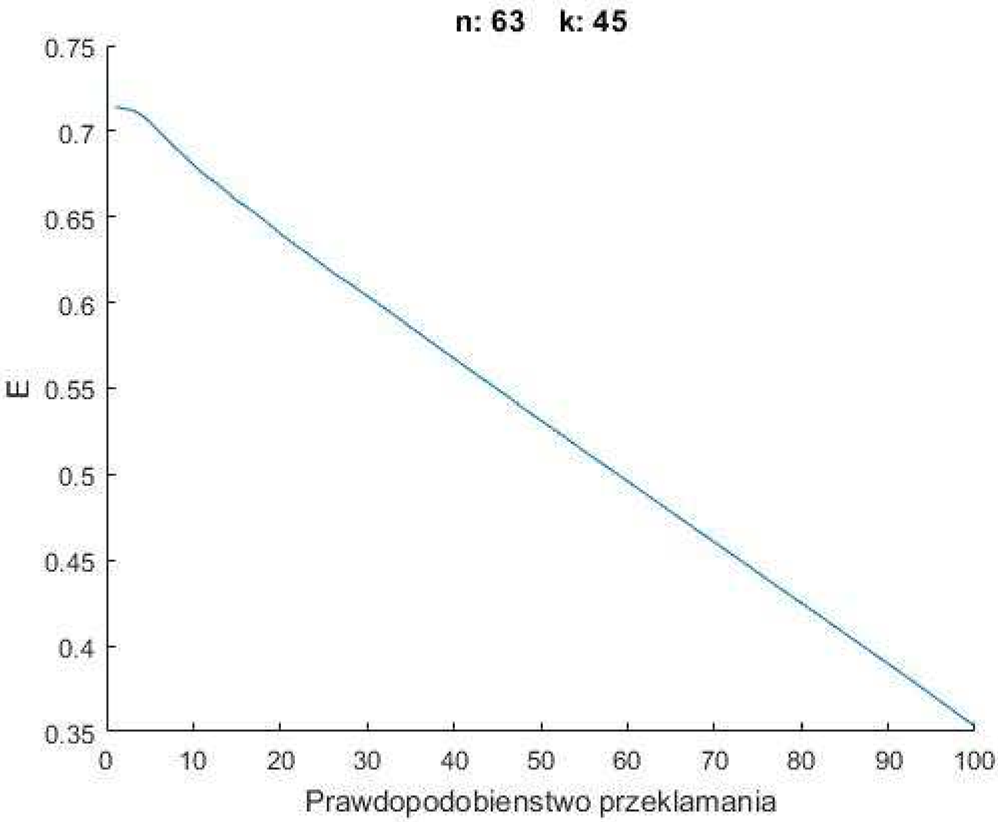
3.2 Etap 2 - BER(pp), BER(sigma), E(pp), E(sigma)



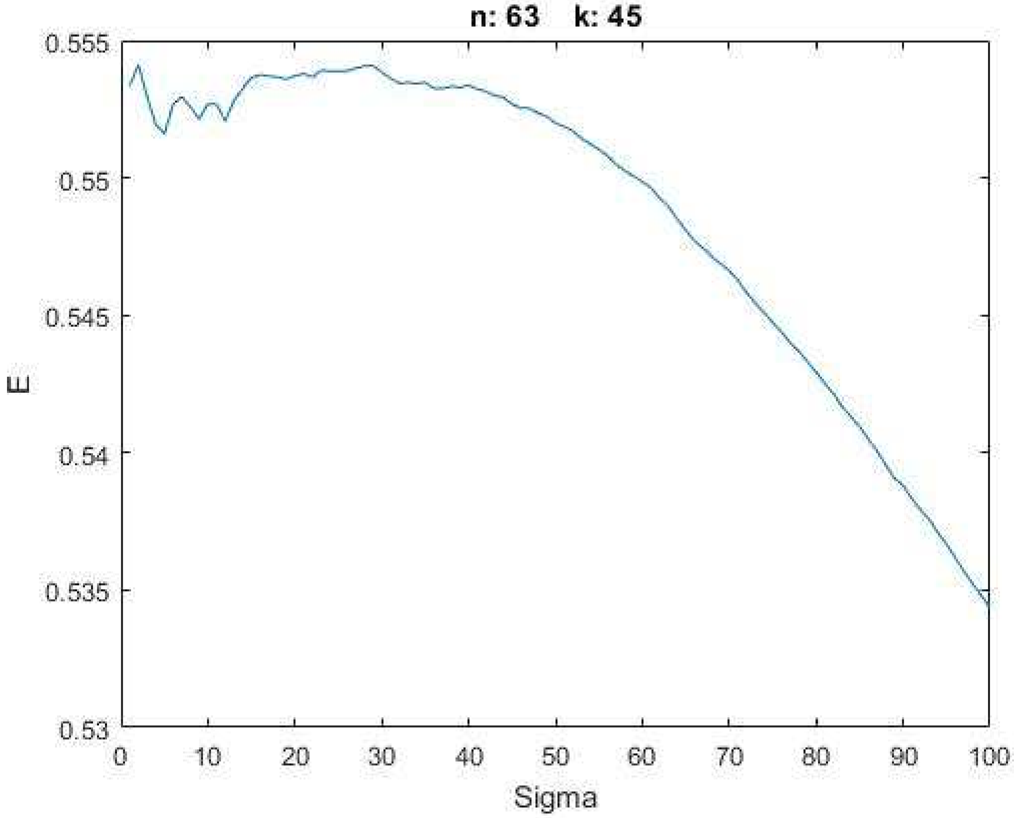
Rys. 4 Wykres zależności BER od prawdopodobieństwa przekłamania dla określonych parametrów n i k.



Rys. 5 Wykres zależności BER od sigmy dla określonych parametrów n i k.

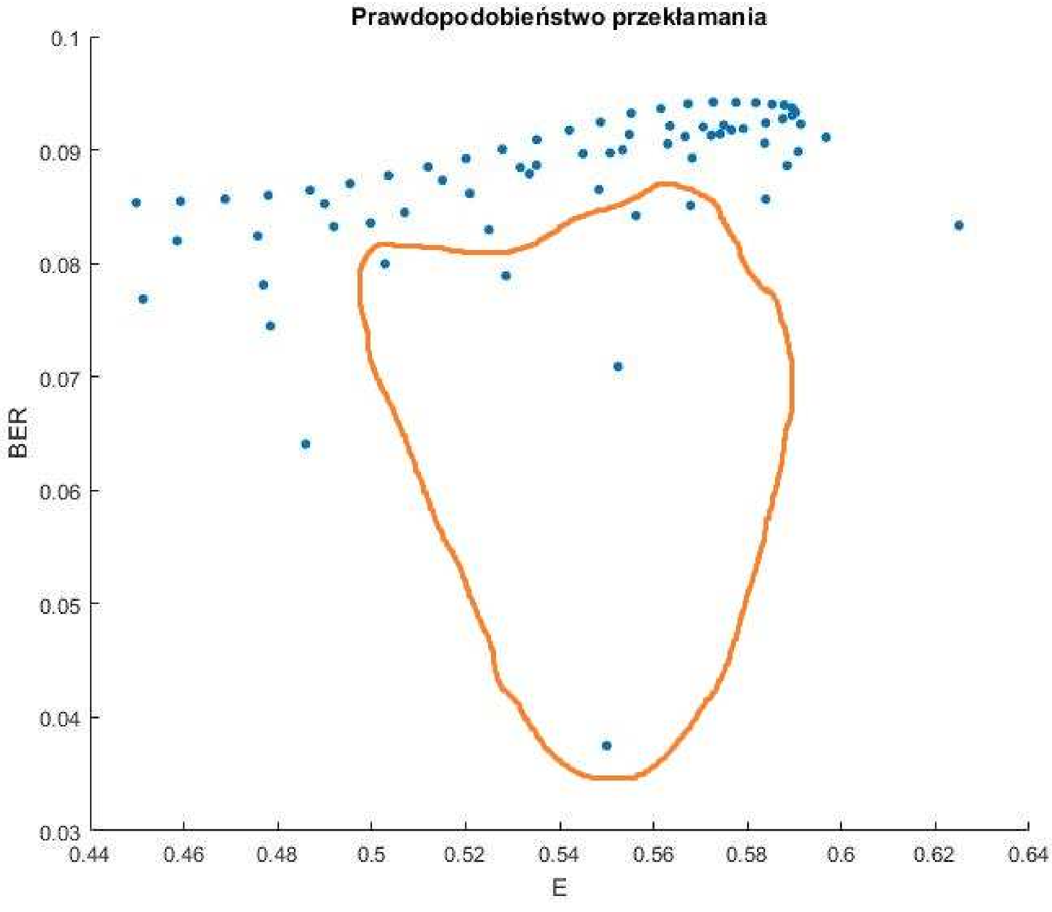


Rys. 6 Wykres zależności E od p. przekłamania dla określonych parametrów n i k.

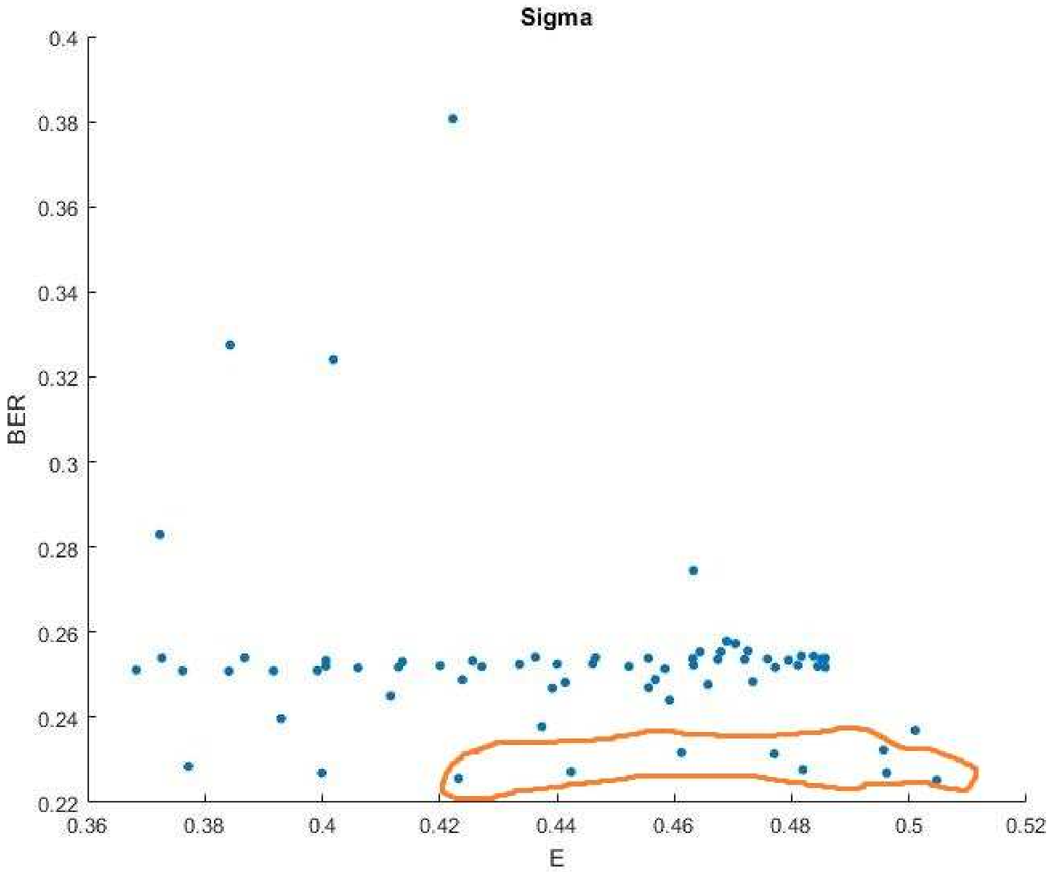


Rys. 7 Wykres zależności E od sigmy dla określonych parametrów n i k.

3.3 Etap 3 - BER(E) dla PP, BER(E) dla sigmy.



Rys. 8 Wykres zależności BER od E z wykreślonym Zbiorem Pareto dla p. przekłamania.



Rys. 9 Wykres zależności BER od E z wykreślonym Zbiorem Pareto dla sigmy.

4. Analiza i interpretacja wyników.

Uzyskane wyniki ukazują badany problem, nawiązujący do systemu FEC. Parametrami, które były rozpatrywane w zależności od różnych zmiennych była wartość **BER** (Bit Error Rate) zwana stopą błędów oraz wartość **E**, czyli stopa prawidłowych bitów.

W dziedzinie pradopodobieństwa przekłamania sygnału w sposób losowy można łatwo zauważyć, że posiada ona charakterystykę liniowo rosnącą. Jest to w pełni zrozumiałe, ponieważ BER (jak już wcześniej zostało wspomniane) nazywamy ilorazem błędnie otrzymanych bitów do wszystkich wysłanych. Zatem jeśli wzrasta prawdopodobieństwo możliwości przekłamania wysyłanego sygnału to automatycznie liczba przekłamanych bitów także rośnie.

Podobnie ma się sytuacja z parametrem **E.** Jej zmiana także jest liniowa w dziedzinie prawdopodobieństwa przekłamania z tą różnicą, że jest ona liniowo malejąca. Zatem jeśli prawdopodobieństwo przekłamania rośnie, to liczba prawidłowo otrzymanych bitów maleje – jest to logiczne.

Sytuacja ma się trochę inaczej, gdy w dziedzinie umiejscowimy parametr sigma, a zatem odchylenie standardowe. Odchyleniem standardowym nazywamy rozrzut wszystkich wyników od wartości średniej. Im wyższą wartość posiada ten parametr statystyczny, tym większy rozrzut mają wyniki poszczególnych badań/operacji.

Dla wykresów, w których dziedzina stanowiła sigma, a więc odchylenie standardowe charakterystyka liniowości została zaburzona. To jest zrozumiałe – wyniki cechujące się rozrzutem o odpowiedniej wielkości zawsze – przy wielokrotnej próbie i dużej liczbie populacji danych - będą się różniły od wartości średniej w sposób większy bądź mniejszy.

Uzyskując wykresy BER w dziedzinie wartości parametru E oszacowany został Zbiór Pareto dla przypadku Prawdopodobieństwa Przekłamania oraz dla sigmy. Zostały one zakreślone na wykresach i zbiory tych punktów charakteryzują optymalne wyniki, w których to stopa błędów powstała w wyniku transmisji danych jest niska i zarazem stopa właściwie odebranych bitów jest satysfakcjonująca. Najlepsze efekty uzyskano m.in. dla par **n** i **k** z tabeli parametrów kodu BCH**: (63,18), (31,11), (7,4), (15,7), (63, 24)** oraz dla prawdopodobieństwa przekłamania m.in. **(15,11), (63,51), (255, 187), (255, 179), (255, 191)** dla sigmy.

5. Uwagi i wnioski.

Zadanie projektowe miało na celu oswoić studenta z rozpatrywanym problemem i metodą kodowania korekcyjnego w przypadku transmisji danych. Poszczególne etapy przebiegały bez większych problemów i sprowadzały się głównie do utworzenia oraz stopniowego rozszerzania pewnego umownego modelu, który jest symulacją rzeczywistości. W sposób zadany generowano sygnał w postaci ciągu bitów, który później w sposób losowy był modyfikowany na drodze dodawania szumów oraz kodowania i dekodowania sygnału. Wartym do odnotowania jest fakt, że nie istnieją takie parametry korekcyjne (kodów BCH), które okazałyby się zdecydowanie lepsze od innych wartości. Istnieją lepsze, efektywniejsze, aczkolwiek nie możemy żadnego przypadku wyszczególnić. W całym układzie musi być reprezentowana pewna losowość zdarzeń, a w takim przypadku konieczne jest liczenie się z podejmowaniem decyzji, które są uwarunkowane kosztem innych czynników, często mniej korzystnych dla nas w innych kryteriach i dziedzinach. Zatem by uzyskać optymalny wynik, trzeba liczyć się z pewnymi ograniczeniami, na które nie mamy większego wpływu.