Politechnika Wrocławska
Wydział Elektroniki

NIEZAWODNOŚĆ I DIAGNOSTYKA
UKŁADÓW CYFROWYCH

SCRAMBLING

Autorzy:
Alicja Myśliwiec 248867
Daria Milczarska 248894
Dominik Kurowski 248840

Termin zajęć:
Wtorek 11.15-13.00 TN

Prowadzący:
dr inż. Jacek Jarnicki

Spis treści

1 Wstęp

Tematem naszego projektu jest scrambling oraz zasady działania scramblerów i descramblerów. Do tworzenia programów na poszczególne etapy wykorzystaliśmy środowisko Octave. Projekt składał się z trzech etapach omówionych poniżej.

1.1 Etap 1

W pierwszym etapie napisaliśmy program symulujący działanie scrmablera stosowanego w telewizji cyfrowej w standardzie DVB. Należało napisać program symulujący działanie scramblera stosowanego w telewizji cyfrowej w standardzie DVB. Napisany program powinien przekształca ciąg bitów o zadanej długości w inny ciąg, o tej samej długości. Przetestowaliśmy program dla trzech przepadków ciągu wejściowego:

- losowego ciągu bitów
- ciągu samych zer
- ciągu samych jedynek

Wyniki symulacji należy przedstawiliśmy przy pomocy histogramów pokazujących częstość występowania ciągów takich samych bitów o różnej długości pamiętając, że liczba transmitowanych bitów m powinna być tak duża, aby osiągnąć stabilność wyników.

1.2 Etap 2

Podobnie jak w poprzednim etapie zasymulowaliśmy działanie scramblera i descramblera wg. standardu V.34. Od scramblera DVB różnią się one zasadą działania oraz tym, że scrambler i descrambler są inne. Dla scramblera V.34 wykonaliśmy odpowiednie symulacje i utworzyć histogramy, pokazujące częstości występowania ciągów takich samych bitów danych składających się z ciągu zer, ciągu jedynek i ciągu losowego. Porównaliśmy jakościowo histogramy dla scramblerów DVB i V.34 i sformułowaliśmy odpowiednie wnioski. Następnie zmodyfikowaliśmy symulator wprowadzając pomiędzy scramblerem i descramblerem kanał transmisyjny, w którym niektóre bity są przekłamywane z prawdopodobieństwem p Zaobserowwaliśmy również skutki pojawienia się pojedynczego błędu transmisji. Dla obu scramblerów (DVB i V.34) przeprowadziliśmy właściwe symulacje i narysowaliśmy wykresy pokazujące, jak zmienia się wskaźnik BER w zależności od prawdopodobieństwa przekłamania bitu w kanale p. Na koniec porównaliśmy oba scramblery i sformułowaliśmy wnioski.

1.3 Etap 3

Zadanie ostatniego etapu projektu polegało na zbadaniu, jak proces scramblingu wpływa na zjawisko utraty synchronizacji transmisji. W tym celu stworzyliśmy prosty model, pozwalający na opisanie procesu utraty i odzyskiwania synchronizacji pomiędzy nadajnikiem i odbiornikiem. Jakość synchronizacji w takim systemie można scharakteryzować przy pomocy wskaźnika wyrażającego udział czasu pozostawania systemu w stanie synchronizacji do całego czasu który upłynął. Należało opracować symulator pozwalający na zamodelowanie procesu synchronizacji dla trzech przypadków:

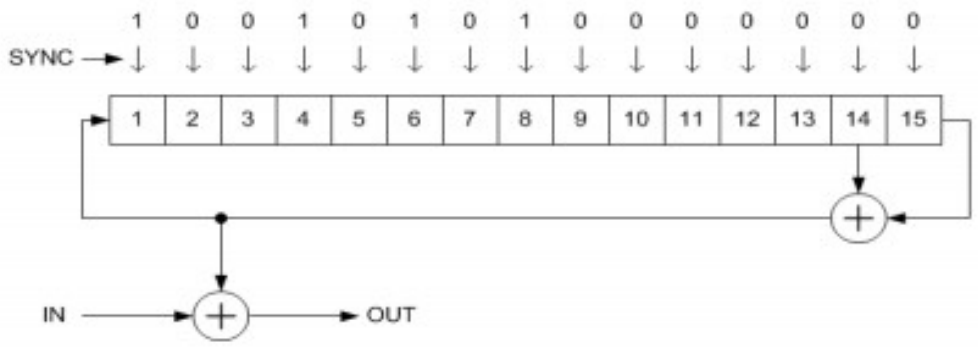
- bez scramblera
- DVB
- V.34

z możliwością zmiany prawdopodobieństwa przekłamania bitu w kanale transmisyjnym. Dla każdego z przypadków wyznaczyliśmy wskaźnik A i porównaliśmy wyniki. Zbadaliśmy również, jak zmienia się wskaźnik A w zależności od prawdopodobieństwa p przekłamania bitu w kanale transmisyjnym oraz wartości długości sekwencji zer k i liczby l odpowiadającej czasowi powrotu synchronizacji.

2 Scrambler/Descrambler DVB oraz V.34

2.1 Scrambler

Scrambler służy do randomizacji ciągów zer i jedynek. Generowany jest ciąg pseudolosowy, który następnie mieszany jest z danymi wejściowymi. Nie wszystkie ciągi bitów są łatwe do przekazania, szczególnie wymagające są te składające się z samych zer lub samych jedynek. Takim przykładem jest kod NRZ, w którym faza może przyjmować jedną z dwóch wartości przesuniętych względem siebie o 180 stopni reprezentując logiczne 0 i 1. W tym wypadku brak przeźroczystości kodowej wynika z braku synchronizacji odbiornika, która występuje w przypadku długiej sekwencji składających się z samych zer. Metoda scramblingu opiera się na założeniu, że pewne ciągi bitowe są bardziej prawdopodobne niż inne, ale trudniejsze do przesyłania. Zadaniem scramblera jest randomizacja ciągu na łatwiejszy do przekazania. Ciąg ten powstaje przy użyciu bramki XOR, która sumuje kod z pseudolosowymi wartościami. W wyniku takiej operacji powstaje maksymalnie długi ciąg, który jest potem przesyłany kanałem transmisyjnym. Scrambling wykorzystuje się nie tylko do usuwanie z ciągu bitów przygotowywanych do transmisji długich ciągów zer lub jedynek. Innym zastosowaniem scramblerów jest kryptografia, na przykład zabezpieczanie płatnych programów telewizyjnych przed niepowołanym dostępem. Na rysunku przedstawiony jest schemat scramblera składającego się z dwóch bramek XOR i rejestru przesuwnego z przykładowym słowem początkowym.



2.2 Scrambler DVB

Scrambler DVB składa się z dwóch bramek XOR i 15 bitowego rejestru przesuwnego, inicjalizowanego słowem początkowym 10001010100000000. Rejestr przesuwny i połączona z jego 14 i 15 pozycją bramka XOR realizuje generator pseudolosowy, którego wyjście podawane jest na wejście drugiej bramki XOR, która przekształca bit wejściowy w bit wyjściowy. Dekodowanie odebranego ciągu bitów w odbiorniku realizowane jest przy pomocy drugiego identycznego układu, co może być bardzo pomocne przy testowaniu poprawności działania symulatora, bowiem ciąg wejściowy podany kolejno na dwa takie układy nie powinien się zmienić.

2.3 Descrambler

Descrambler dekoduje informacje do postaci pierwotnej. W jednym i drugim urządzeniu używa się rejestrów przesuwnych. Realizacja dekodowania odebranego ciągi bitów w odbiorniku przy pomocy takiego układu pozwala przetestować poprawność działania symulatora, ponieważ ciąg powinien pozostać taki sam.

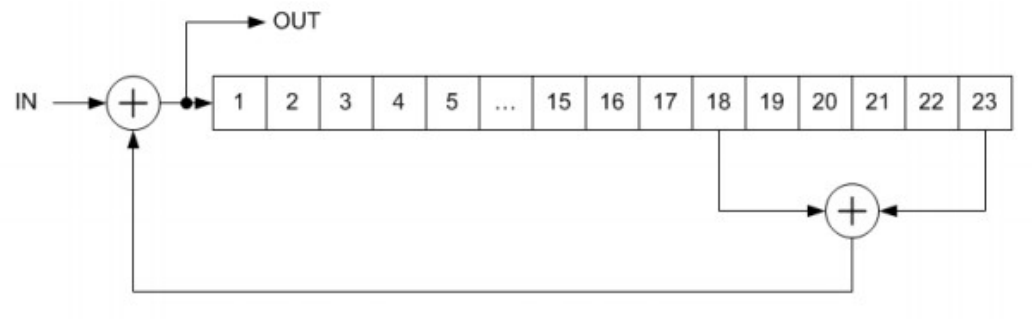
2.4 V.34

V.34 to standard w komunikacji rekomendowany przez ITU-T, charakteryzujący się pełną transmisją dwukierunkową pomiędzy dwoma modemami wdzwanianymi (połączenie wdzwaniane, ang. *dial-up*) przy przepustowości łącza do 28,8 kbit/s. Inne dodatkowo zdefiniowane przepustowości to: 24,0 kbit/s, 19,2 kbit/s a także przepustowości, które są dopuszczalne przez zalecenia V.32 i V.32bis. Standard ten jest następcą nieoficjalnego standardu V.FC (V.Fast Class) opracowanego przez firmy Hayes i Rockwell International, który był pierwszym szeroko dostępnym protokołem

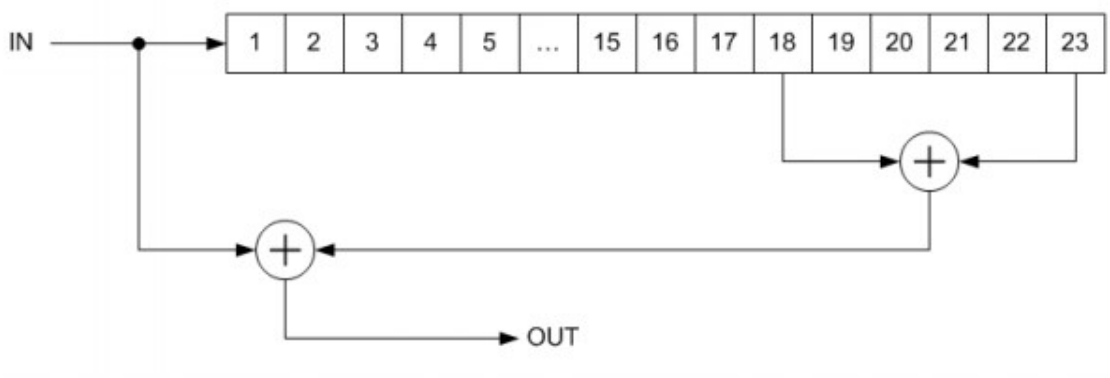
oferującym przepustowość 28,8 kbit/s. Większość modemów standardu V.34 obsługuje także protokół V.FC jakkolwiek nie wszystkie nowoczesne modemy obsługują je oba.

2.5 Scrambler i descrambler wg. standardu V.34

Scrambler i descrambler wg. standardu V.34 to urządzenia różniące się zasadą działania od scramblera i descramblera przedstawionych w etapie pierwszym. Mamy tutaj do czynienia z dwoma różnymi układami. Scrambler i descrambler różnią się od siebie. Poniżej na rysunkach przedstawione zostały schematy danych urządzeń.



Scrambler V.34

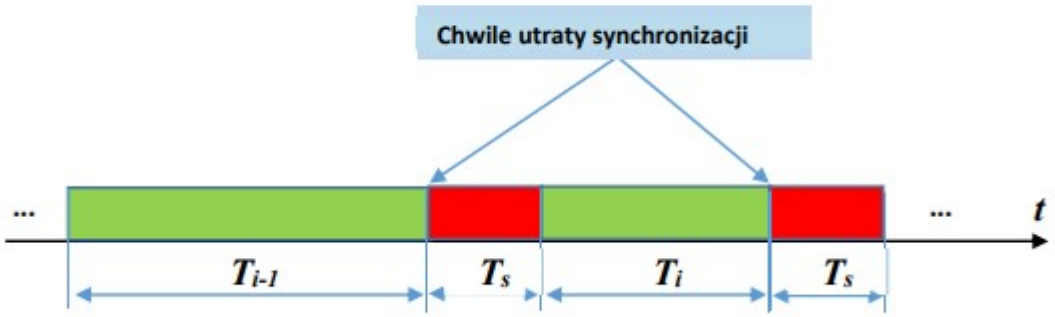


Decrambler V.34

3 Opis przyjętego modelu symulacyjnego systemu

W ramach projektu badaliśmy proces scramblingu oraz jak wpływa on na zjawisko utraty synchronizacji transmisji. W tym celu przyjęliśmy prosty model pozwalający na opisanie procesu utraty i odzyskiwania synchronizacji pomiędzy nadajnikiem i odbiornikiem. System transmisji danych w postaci ciągu bitów składa się z nadajnika, kanału transmisyjnego, w którym z

prawdopodobieństwem p mogą występować przekłamania i odbiornika. W nadajniku został zastosowany scrambler, a w odbiorniku odpowiedni descrambler. Odbiornik po odebraniu sekwencji k kolejnych zer traci synchronizację, co powoduje, że przez czas odpowiadający transmisji l kolejnych bitów nie może prawidłowo odbierać danych. Mechanizm procesu przywrócenia synchronizacji nie jest modelowany w projekcie, jednak powinien następować po czasie odpowiadającym transmisji l bitów.



Rysunek przedstawia naprzemiennie okresy synchronizacji transmisji o losowym czasie trwania (kolor zielony) i okresy przywracania synchronizacja o stałej długości (kolor czerwony). Jakość synchronizacji w takim systemie można scharakteryzować przy pomocy wskaźnika wyrażającego udział czasu pozostawiania systemu w stanie synchronizacji do całego czasu, który upłynął. Postać wskaźnika wygląda następująco:

$$A = \frac{\sum_{i=1}^{i=n} T_i}{n \cdot T_s + \sum_{i=1}^{i=n} T_i}$$

gdzie n jest liczbą przypadków wystąpienia utraty synchronizacji w czasie wykonanej symulacji.

Realizacja projektu polegała na opracowaniu symulatora pozwalającego na zamodelowanie wyżej opisanego procesu synchronizacji dla trzech przypadków:

- Bez scramblera
- DVB
- V.34

z możliwością zmiany prawdopodobieństwa przekłamania bitu w kanale transmisyjnym.

Dla każdego z przypadków należało wyznaczyć zdefiniowany powyżej wskaźnik A i porównać wyniki oraz zbadać jak zmienia się ten wskaźnik w zależności od prawdopodobieństwa p przekłamania bitu w kanale transmisyjnym oraz wartości długości sekwencji zer k i liczby l odpowiadającej czasowi powrotu synchronizacji.

4 Opis programu symulatora

Do wykonania symulacji zostały użyte funkcje stworzone we wszystkich etapach projektu. Do badania ilości utrat synchronizacji została stworzona i zastosowana funkcja przedstawiona poniżej.

```
%--------------------------------Badanie ilosci utrat synchronizacji-------------------------------------
function synchronized_bits = Synchronization (data, k, I)
  counter = 0; % Licznik wystapien po sobie bitow '0'
  synchronized_bits = 0; % Bity odebrane przy prawidlowej synchronizacji
  i = 1; % Indeks kolejnego bitu danych
  while i < length(data)
    if data(i) == 0 % Sprawdz czy dany bit jest rowny '0'
      counter = counter + 1; % Jezeli tak to inkrementuj licznik wystepujacych po sobie '0'
    else
      counter = 0; % W przeciwnym wypadku zeruj licznik
    endif

    if counter == k % Jezeli licznik jest rowny liczbie bitow zerowcyh potrzebnych do desynchronizacji
      counter = 0; % Zeruj licznik
      i = i + I; % Pomin nastepne I bitow
      continue; % Pomin dalsza czesc petli
    endif

    synchronized_bits = synchronized_bits + 1; % Zwieksz licznik bitow odebranych przy prawidlowej synchronizacji
    i = i + 1; % Przejdz do nastepnego indeksu bitu danych
  endwhile
endfunction

```

Poniżej przedstawiony jest generator bitów o zadanej długości oraz inicjalizacja rejestrów dla scramblerów.

```
clear;
%---------------------------Symulacja bez przeklaman na kanale transmisyjnym----------------------%
N = 16384; % Ilosc bitow wejsciowych
register_DVB = [1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0]; % Inicjalizacja rejestru dla DVB
register_V34 = [1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0]; % Inicjalizacja rejestru dla v.34
%---------------------------Losowe dane na wejsciu---------------------------%
%Generator ciagu bitow o zadanej dlugosci
for i = 1:N
  temp = rand();
  if temp > 0.5 %Przyjmujemy, ze dla wartosci powyzej 0.5, bit bedzie mial wartosc 1
    data(i) = 1;
  else          %Przyjmujemy, ze dla wartosci ponizej 0.5, bit bedzie mial wartosc 0
    data(i) = 0;
  end
endfor

```

4.1 Zależność A od p

```
%--------------------------Zaleznosc A od p--------------------------%
A1 = A2 = A3 = [];
k = 9; % Dlugosc sekwencji zer
I = 256; % Czas powrotu synchronizacji
p_range = 0:0.05:0.5; % Prawopodobienstwa wystapienia przeklamania
for p = p_range
  % Bez scramblera
  scrambled_data = data;
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A1 = [A1, synchronized_bits / N];

  % Scrambler DVB
  scrambled_data = Scramble_DVB(data, register_DVB);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A2 = [A2, synchronized_bits / N];

  % Scrambler v.34
  scrambled_data = Scramble_V34(data, register_V34);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A3 = [A3, synchronized_bits / N];
endfor
p_range = p_range*100;

figure(1)
plot(p_range, A1,'LineWidth', 2)
hold on
plot(p_range, A2,'LineWidth', 2)
plot(p_range, A3,'LineWidth', 2)
hold off

title('Zaleznosc wspolczynnika A od prawdopodobienstwa przeklamania bitu, k = 9, I = 256')
xlabel('Prawdopodobienstwo przeklamania bitu w kanale transmisyjnym [%]')
ylabel('Wspolczynnik pozostania w stanie synchronizacji')
legend({'Bez scramblera', 'Scrambler DVB', 'Scrambler v.34'},'Location','southwest')
legend('boxoff')
axis([0 50])
```

4.2 Zależność A od k

```
%--------------------------Zaleznosc A od k--------------------------%
A1 = A2 = A3 = [];
k_range = [4,5,6,7,8,9,10,11,12,13]; % Dlugosc sekwencji zer
I = 256; % Czas powrotu synchronizacji
p = 0; % Prawopodobienstwo wystapienia przeklamania
for k = k_range
  % Bez scramblera
  scrambled_data = data;
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A1 = [A1, synchronized_bits / N];

  % Scrambler DVB
  scrambled_data = Scramble_DVB(data, register_DVB);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A2 = [A2, synchronized_bits / N];

  % Scrambler v.34
  scrambled_data = Scramble_V34(data, register_V34);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A3 = [A3, synchronized_bits / N];
endfor

figure(2)
plot(k_range, A1,'LineWidth', 2)
hold on
plot(k_range, A2,'LineWidth', 2)
plot(k_range, A3,'LineWidth', 2)
hold off

title('Zaleznosc wspolczynnika <A> od ilosci kolejnych zer <k>, p = 0%, I = 256')
xlabel('Ilosc nastepujacych po sobie zer potrzebnych do desynchronizacji')
ylabel('Wspolczynnik pozostania w stanie synchronizacji')
legend({'Bez scramblera', 'Scrambler DVB', 'Scrambler v.34'},'Location','southeast')
legend('boxoff')
axis([4 13])
```

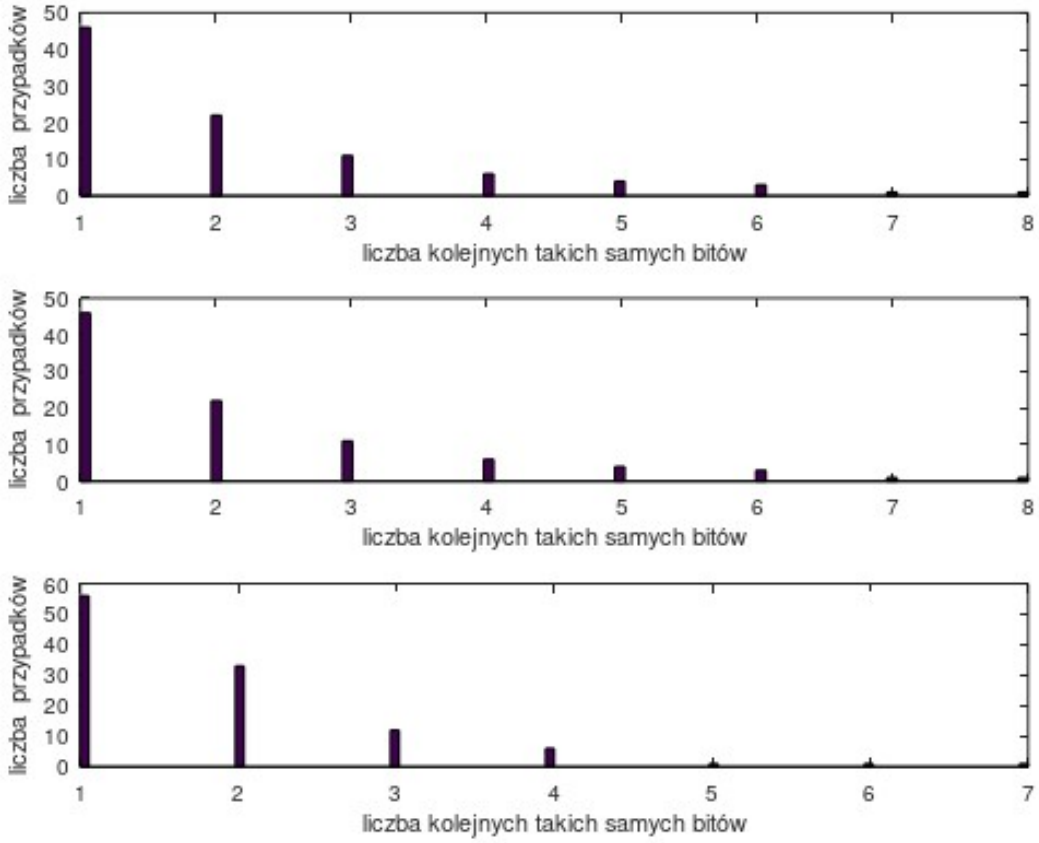
4.3 Zależność A od I

```
%--------------------------Zaleznosc A od I--------------------------%
A1 = A2 = A3 = [];
k = 9; % Dlugosc sekwencji zer
I_range = [256,384,512,640,768,896,1024,1152,1280,1408,1536]; % Czas powrotu synchronizacji
p = 0; % Prawopodobienstwo wystapienia przeklamania
for I = I_range
  % Bez scramblera
  scrambled_data = data;
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A1 = [A1, synchronized_bits / N];

  % Scrambler DVB
  scrambled_data = Scramble_DVB(data, register_DVB);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A2 = [A2, synchronized_bits / N];

  % Scrambler v.34
  scrambled_data = Scramble_V34(data, register_V34);
  transmitted_data = Misrepresentation(scrambled_data, p); % Powstanie zaklaman na kanale transmisyjnym
  synchronized_bits = Synchronization(transmitted_data, k, I); % Zliczanie ilosci desynchronizacji przy odbiorze danych
  A3 = [A3, synchronized_bits / N];
endfor

figure(3)
plot(I_range, A1,'LineWidth', 2)
hold on
plot(I_range, A2,'LineWidth', 2)
plot(I_range, A3,'LineWidth', 2)
hold off

title('Zaleznosc wspolczynnika <A> od czasu potrzebnego na odzyskanie synchronizacji <I>, p = 0%, k = 9')
xlabel('Czas potrzebny na odzyskanie synchronizacji [w bitach]')
ylabel('Wspolczynnik pozostania w stanie synchronizacji')
legend({'Bez scramblera', 'Scrambler DVB', 'Scrambler v.34'},'Location','southwest')
legend('boxoff')
axis([256 1536])
```

5 Wyniki symulacji

5.1 Etap 1

W pierwszym etapie projektowaliśmy scrambler oraz descrambler DVB. Symulacja polegała na sprawdzeniu liczby kolejnych takich samych bitów po przeprowadzeniu scramblingu dla trzech przypadków ciągu bitów na wejściu: same zera, same jedynki, losowy ciąg.

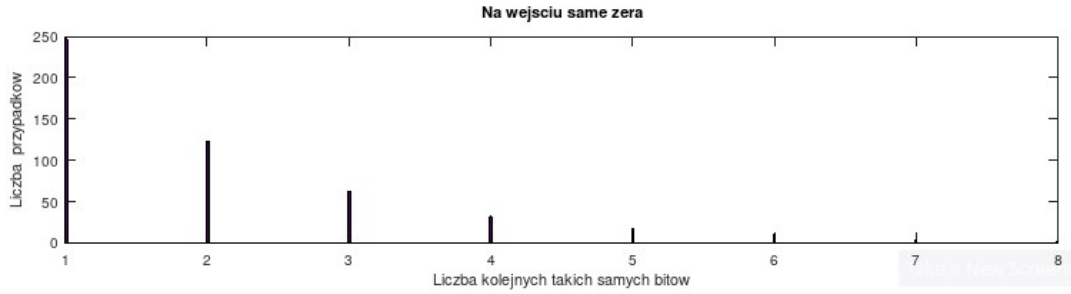


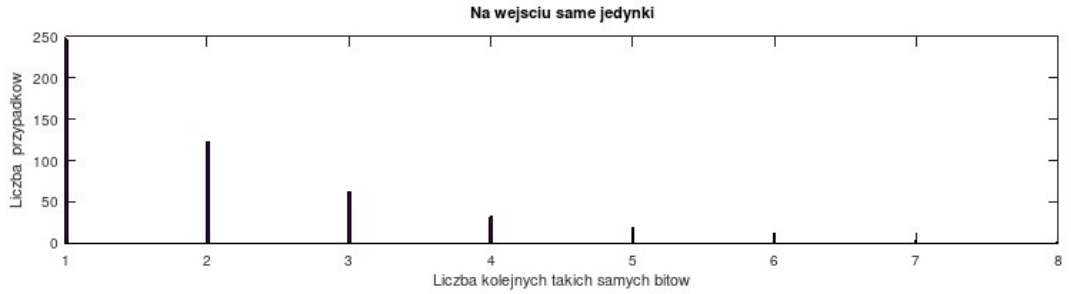
5.2 Etap 2

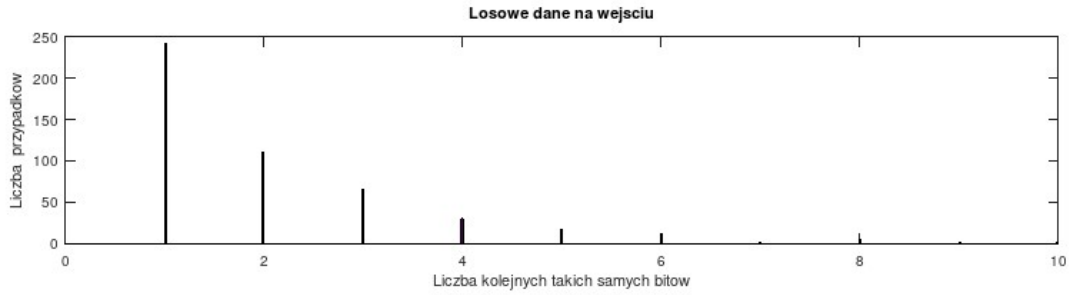
W drugim etapie zaprojektowaliśmy scrambler oraz descrambler w standardzie V.34 i porównywaliśmy jego efektywność do DVB dla trzech przypadków symulacji: bez przekłamań na kanale transmisyjnym, z przekłamaniami na kanale transmisyjnym, z przekłamaniami na pojedynczym bicie. W ramach tego etapu porównywaliśmy również wskaźnik BER dla obu scramblerów, gdy na wejściu został podany losowy ciąg bitów.

Wyniki przedstawione są w następujący sposób: pierwszy wykres dotyczy scramblera DVB, drugi dotyczy scramblera V.34.

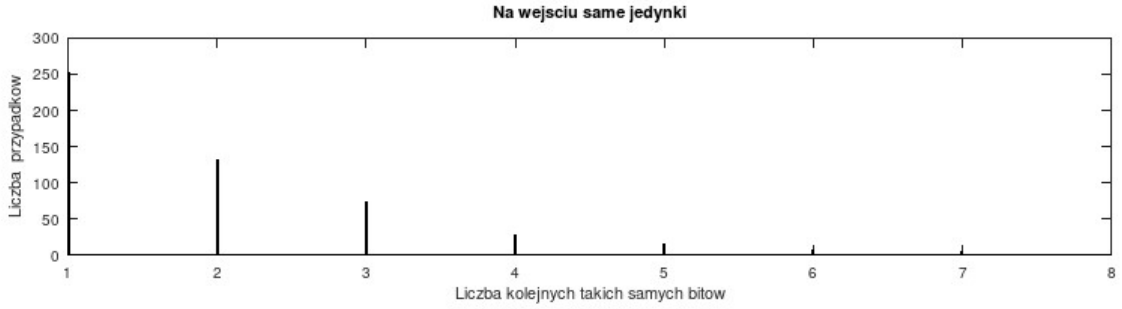
Symulacja bez przekłamań na kanale transmisyjnym

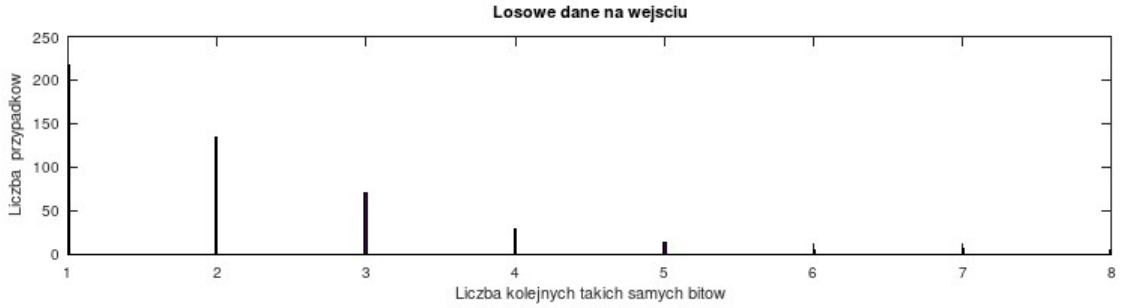




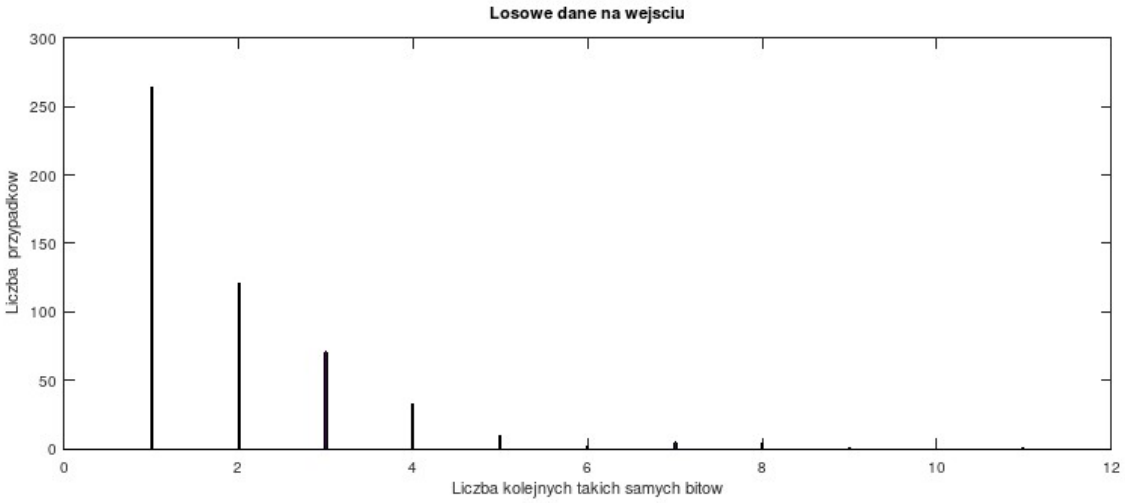


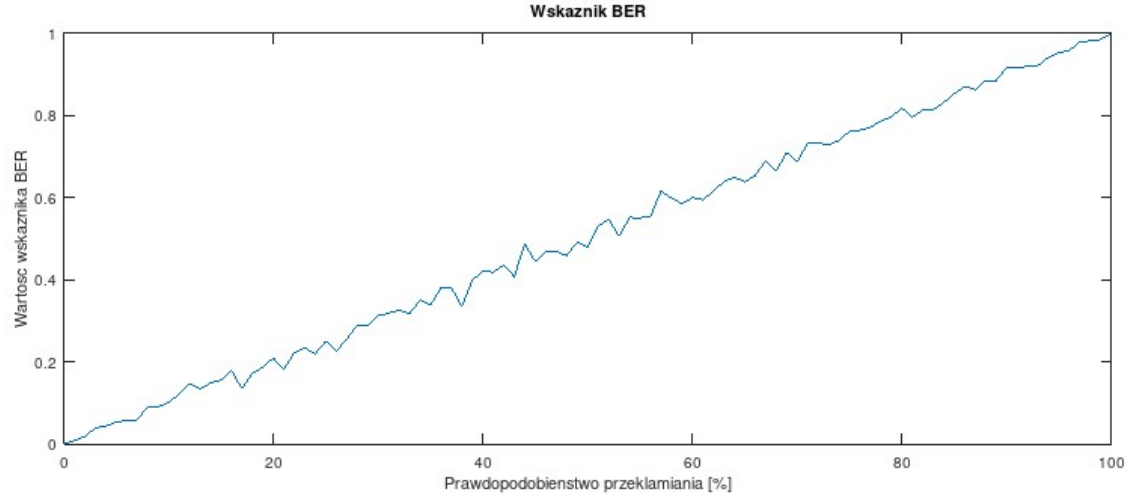

Na wejsciu same zera
Liczba przypadkow
Liczba kolejnych takich samych bitow

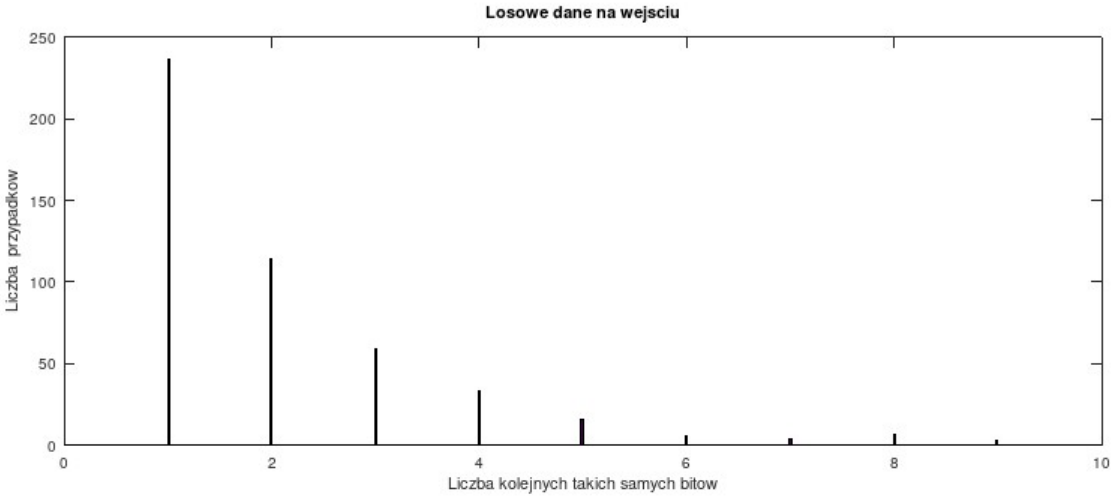

Na wejsciu same jedynki
Liczba przypadkow
Liczba kolejnych takich samych bitow

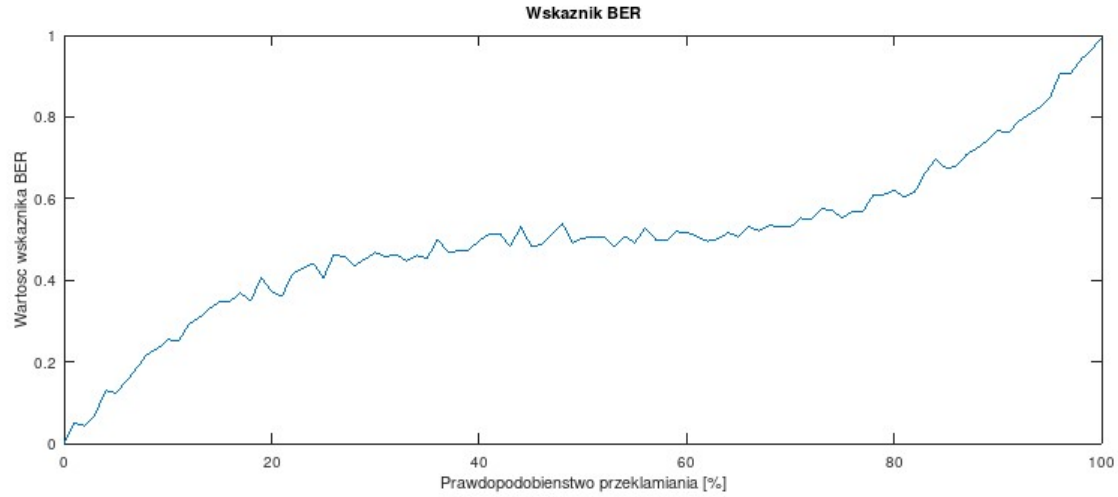

Losowe dane na wejsciu
Liczba przypadkow
Liczba kolejnych takich samych bitow

Symulacja z przekłamaniami na kanale transmisyjnym



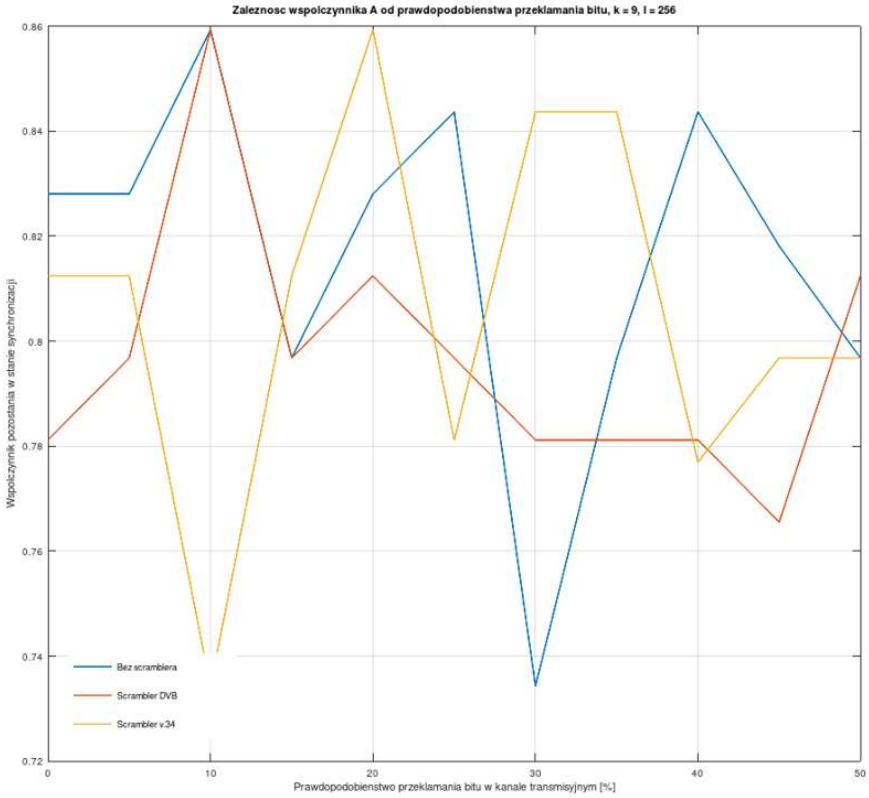


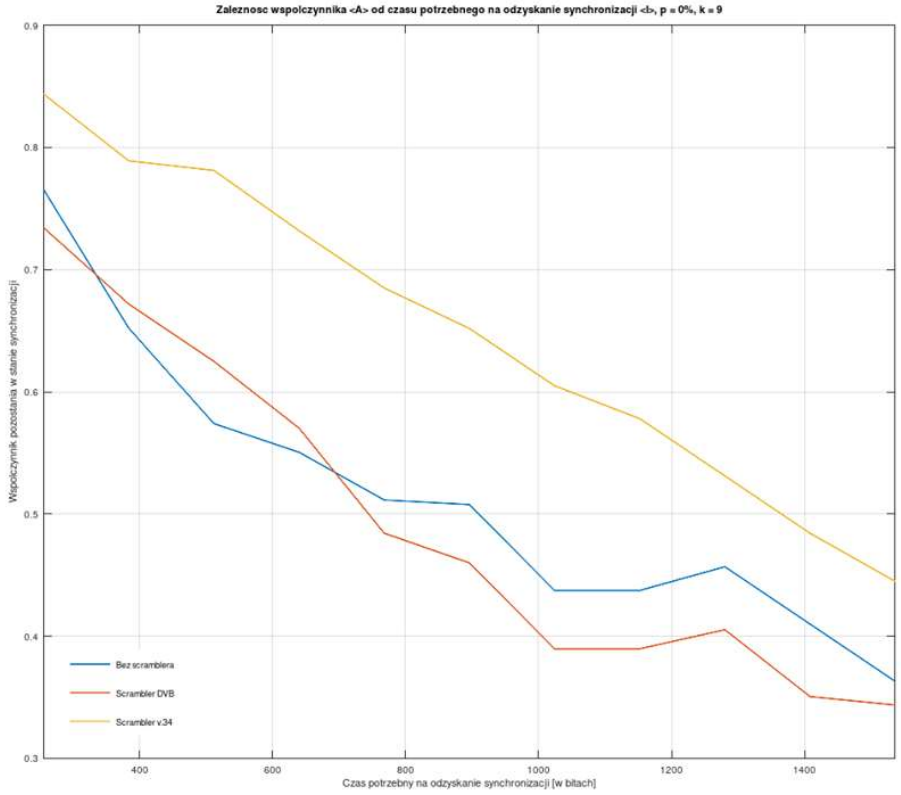


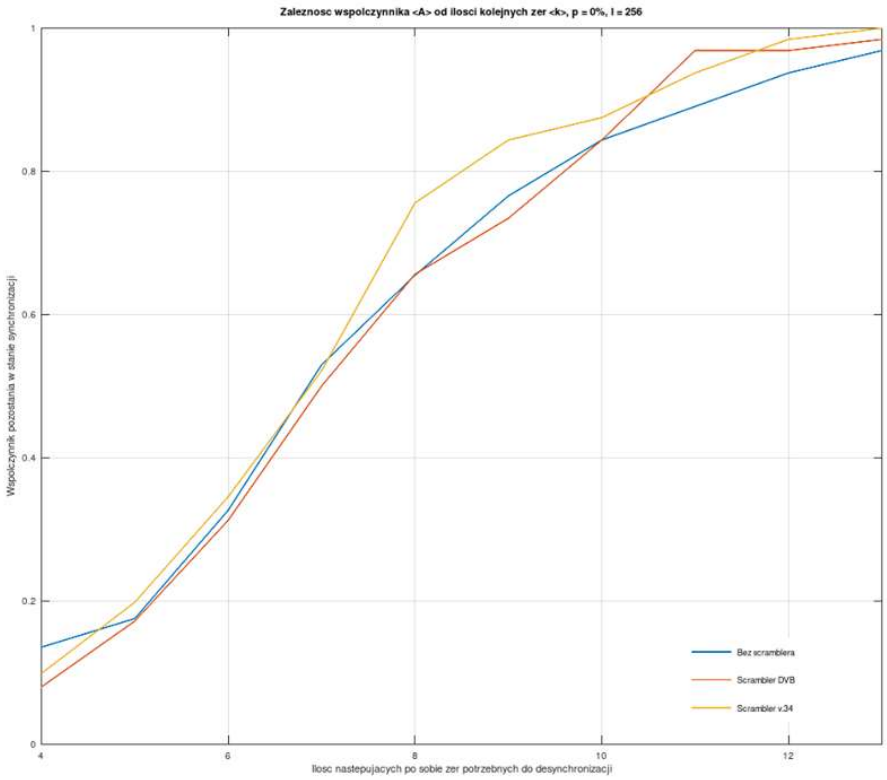


5.3 Etap 3

W trzecim i ostatnim etapie projektu tworzyliśmy symulator utraty synchronizacji. Dla każdego z trzech przypadków: bez scramblera, scrambler DVB, scrambler V.34 należało wyznaczyć zdefiniowany wskaźnik A i porównać wyniki oraz zbadać jak zmienia się ten wskaźnik w zależności od prawdopodobieństwa p przekłamania bitu w kanale transmisyjnym oraz wartości długości sekwencji zer k i liczby l odpowiadającej czasowi powrotu synchronizacji.


Zaleznosc wspolczynnika A od prawdopodobienstwa przeklamania bitu, k = 9, l = 256
Wspolczynnik pozostania w stanie synchronizacji
Prawdopodobienstwo przeklamania bitu w kanale transmisyjnym [%]
Bez scramblera
Scrambler DVB
Scrambler v.34
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0
10
20
30
40
50


Zaleznosc wspolczynnika <A> od czasu potrzebnego na odzyskanie synchronizacji <t>, p = 0%, k = 9
Wspolczynnik pozostania w stanie synchronizacji
Czas potrzebny na odzyskanie synchronizacji [w bitach]
Bez scramblera
Scrambler DVB
Scrambler v.34
0.9
0.8
0.7
0.6
0.5
0.4
0.3
400
600
800
1000
1200
1400



6 Wnioski

Przy porównywaniu scramblerów za pomocą ciągu samych jedynek, samych zer oraz losowych wartości, lepiej wypada scrambler wg. standardu V.34, ponieważ ciągi bitów o takiej samej wartości są krótsze, jednak te różnice są nieznaczne. Scrambler DVB tworzy gorszy rozkład jakościowy. Wskaźnik BER = (liczba błędnie odebranych bitów)/ (liczba odebranych bitów), zatem powinien być on jak najmniejszy. Jeżeli prawdopodobieństwo przekłamania jest duże, to wskaźnik BER jest generalnie wysoki. Przekłamanie na pojedynczym bicie (bit o indeksie 100 z 1000) daje wskaźnik BER: 0.001 dla DVB, 0.003 dla v.34 (dla programu 3). Przewagę scramblera wg. standardu V.34 widzimy porównując wskaźnik BER. Nawet dla dużego prawdopodobieństwa przekłamania sygnału rzędu 70% nadal utrzymuje wskaźnik BER na poziomie 50%, w przeciwieństwie do scramblera DVB, gdzie wskaźnik BER rośnie liniowo.

Otrzymane wyniki jednoznacznie pokazują, jak istotny jest rozwój mechanizmów przeciwdziałających skutkom zakłóceń w technologiach telekomunikacyjnych. Sygnał niosący informację ma jakąkolwiek wartość jedynie, gdy nie został całkowicie zniekształcony w kanale transmisyjnym lub gdy to odkształcenie jest odwracalne z poziomu odbiornika. Dzięki standardom takim jak DVB lub v.34 możemy cieszyć się stabilnością podczas korzystania z telewizji bądź internetu, mimo nieuniknionych, szkodliwych wpływów zewnętrznych.

7 Spis opracowanych programów i funkcji

- Scramble_DVB.m
- Descramble_DVB.m
- Main.m
- Scramble_V34.m
- Descramble_V34.m
- BitSequence.m
- Misrepresentation.m
- Program1.m
- Program2.m
- Program3.m
- Synchronization.m
- Etap3.m

8 Literatura

- https://pl.wikipedia.org/wiki/Scrambler
- https://pl.wikipedia.org/wiki/Digital_Video_Broadcasting
- https://pl.wikipedia.org/wiki/V.34
- https://pl.qwe.wiki/wiki/Scrambler
- https://pl.wikipedia.org/wiki/Kana%C5%82_%C5%82%C4%85czno%C5%9Bci
- https://pl.wikipedia.org/wiki/Telekomunikacja
- https://scholar.google.pl/scholar?q=niezawodno%C5%9B%C4%87+i+diagnostyka&hl=pl&as_sdt=0&as_vis=1&oi=scholart
- http://telewizor.eu/cyfrowa.html
- https://eduinf.waw.pl/inf/alg/002_struct/0010.php
- https://eduinf.waw.pl/inf/utils/010_2010/0004.php
- https://en.wikipedia.org/wiki/Common_Scrambling_Algorithm
- https://pl.wikipedia.org/wiki/Advanced_Encryption_Standard
- https://pl.wikipedia.org/wiki/Kryptologia
- https://pl.wikipedia.org/wiki/Kryptoanaliza