Sprawozdanie

Niezawodność i diagnostyka układów cyfrowych – projekt

Temat: Scrambling

Grupa IV:

-Beata Szeląg, 218139

-Szymon Wojciechowski, 218154

-

-

-

[dopiszcie się,podajcie inkdesy]

Termin:

Poniedziałek, tydzień parzysty, 14:00-16:00

1. **Wstęp**

Podczas transmisji danych najważniejszym jest, aby dane w odbiorniku były jak najbardziej zbliżone do tych, przesłanych przez nadajnik. Związane jest to z synchronizacją danych – gdzie istotnym jest rozwiązanie problemu pojawiających się ciągów zer bądź jedynek. Jednym z rozwiązań jest scrambling – dzięki któremu część bitów, wchodzących w skład ciągu wejściowego zostaje zmieniona. Operacją odwrotną, przywracającą pierwotny sygnał jest descramblowanie, które odbywa się w dekoderze. Jednak scrambling jest również traktowany jako rodzaj szyfrowania, używany w kryptografii, na przykład przez producentów płatnych programów telewizyjnych.

W powyższym sprawozdaniu przedstawimy sposób działania dwóch rodzajów scramblerów, omówimy proces synchronizacji nadajnika i odbiornika, przedstawimy symulacje generatora pseudolosowego oraz wspomnianych już nadajnika i odbiornika. Ponadto zbadamy dobroć całego układu.

1. **Scramblery**
   1. Zastosowanie

Podczas transmisji, gdy w nadajniku pojawią się dane w formie ciągów okresowych, których okres jest równy czasowi trwania pojedynczego bitu – a więc występują one w postaci ciągu zer bądź jedynek – obserwujemy problemy przy procesie synchronizacji. Jest to spowodowane tym, że sygnał, jaki w tym przypadku zostałby przez nas odebrany, nie zawierałby informacji dotyczącej czasu trwania takiego bitu danych. [2] W związku z tym, jakakolwiek próba synchronizacji pracy nadajnika i odbiornika byłaby utrudniona – przez co w efekcie najprawdopodobniej otrzymalibyśmy zniekształcony ciąg danych. Aby temu zapobiec stosuje się scrambleryzację – która polega na uporządkowaniu danych z nadajnika, tak, aby w jak największym stopniu uniknąć długich, jednolitych ciągów danych.

Rys 2.1.1. Proponuję tutaj wykres zliczania jedynek i zer.

* 1. Zasada działania

Uporządkowanie danych odbywa się za pomocą wielomianu pierwotnego, oznaczanego w literaturze jako h(x). [2]

*(x) = x M⊕ CM-1 x M-1⊕ .......... ⊕ C1 x⊕ 1 (1)*

Wielomian ten, z definicji, nie jest podzielny przez żaden inny wielomian, dzieli się bez reszty przez *xn ⊕ 1* dla *n = 2M-1* , lecz nie dzieli się dla *n < 2M-1* . Testując scramblery DVB skorzystaliśmy z wielomianu:

*h(x) = 1 + x14 + x15 (2)*

Odpowiada to operacji XOR między bitami x14 oraz x15.

Dla scramblera V34 wielomian pierwotny wygląda następująco [1] :

*h(x) = 1 + x-18 + x-23 = 1 + x5 + x23 (3)*

Odpowiada to operacji XOR między bitami x5 oraz x23.

Implementując obie te funkcje w programie Matlab, wykorzystaliśmy zależność, według której logiczna operacja XOR została zamieniona na arytmetyczną operację modulo.

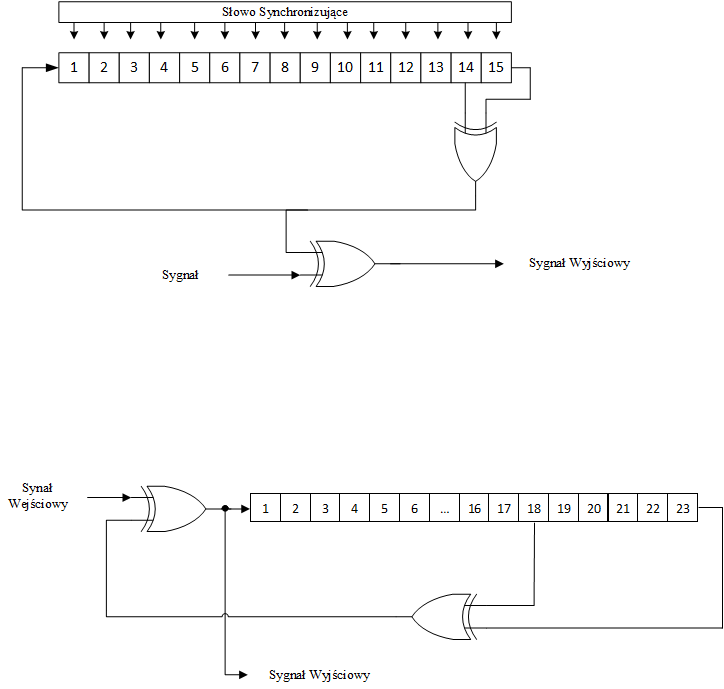
Jak wcześniej zostało wspomniane, wykonaliśmy symulatory zarówno dla scramblera używanego w DVB jak i V34 – a więc dla scramblerów addytywnego i multiplikacyjnego.

* + 1. Scrambler addytywny (używany w DVB)

Bit danych pochodzących z nadajnika jest jednym z argumentów funkcji XOR. Drugim jest wynik operacji wynikającej z generatora pseudolosowego. Wynik tej operacji jest przesyłany przez kanał transmisyjny.

Działanie generatora pseudolosowego: dwa ostatnie bity klucza (w wielomianie pierwotnym oznaczane jako x14, x15) wchodzą w skład operacji XOR. [3] Wykonujemy rotację w prawo o jedną pozycję - w rezultacie tracimy najmłodszy bit i pojawia się miejsce na najstarszym bicie, które zostanie zajęte przez wyliczony wcześniej wynik operacji XOR.

Na *Rys.2.2.1.1* widoczny jest logiczny schemat wyżej przedstawionego algorytmu:

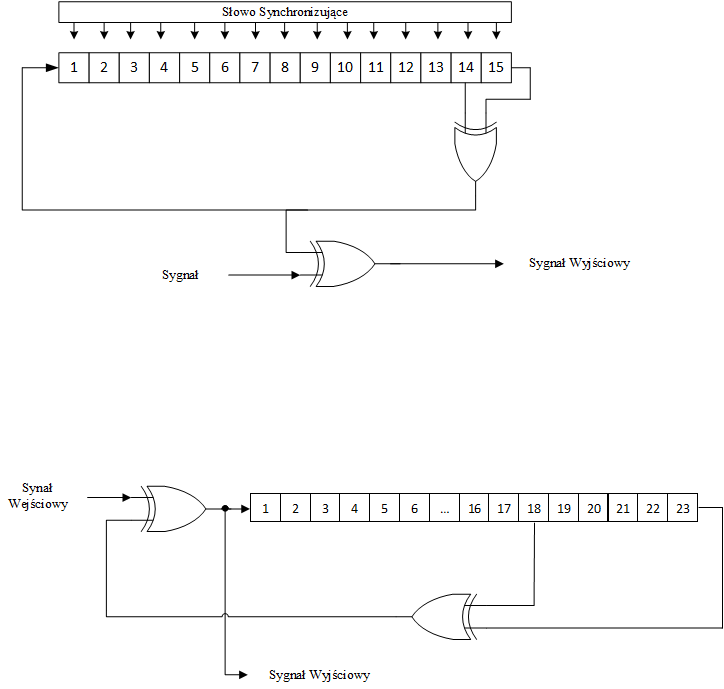


*Rys.2.2.1.1. Scrambler addytywny*

* + 1. Scrambler multiplikacyjny (używany w V-34)

Bity klucza: x23, x5 wchodzą w skład operacji XOR. Jej wynik jest również poddany operacji XOR, gdzie drugim argumentem jest bit danych z nadajnika. Ten wynik trafia zarówno do kanału transmisyjnego, jak i po rotacji (analogicznej jak przy DVB) staje się najstarszym bitem w kluczu. Takie postępowanie wymusza stworzenie osobnego algorytmu deszyfrującego (w odróżnieniu od scramblera addytywnego, gdzie szyfrowanie i deszyfrowanie jest identyczne). [2]

Różnicą między scramblerem V34, a descramblerem jest to, że dla tego drugiego, do najstarszego bitu klucza trafia nie wynik przekazywany do odbiornika, lecz dana wejściowa. Na *Rys.2.2.2.1* widoczny jest logiczny schemat wyżej przedstawionego algorytmu:



*Rys.2.2.2.1 Scrambler multiplikacyjny*

1. **Symulacja wybranych scramblerów**
   1. DVB

Zgodnie z opisem scramblera addytywnego w punkcie 2.2.1 jego działanie opiera się na dodawaniu (zarówno w nadajniku, jak i odbiorniku) ciągu, który został wygenerowany przez wielomian h(x), szerzej opisany również we wspomnianym punkcie. Taki algorytm wykorzystuje własność, mówiącą, że dodając ten sam ciąg do siebie, wykonując operację modulo 2 – otrzymujemy ciąg zerowy. Własność ta jest przedstawiona na *Rys.3.1.3.*

Jednak takie postępowanie wymusza na nas synchronizację nadajnika i odbiornika – która będzie oparta o słowo synchronizujące. Wtedy też, jeśli układ wykryje takie słowo w przesyłanym ciągu – to klucze znajdujące się w nadajniku i odbiorniku zostaną odnowione. Dzięki temu, jeśli w trakcie transmisji wystąpił błąd – to będziemy mieli pewność, że dalsze przesyłanie danych (po synchronizacji) będzie już poprawne.

Efekt scramblowania widoczny jest na *Rys.3.1.1.* oraz *Rys.3.1.2*, gdzie *Rys.3.1.1* pokazuje sygnał przed scramblowaniem, *Rys.3.1.2* – po tym procesie.

Funkcja szyfrująca:

function [out] = DVB (key, in)

for i = in

temp = xor (key(end), key(end - 1));

key = shift(key,1);

key(1) = temp;

out(end+1) = xor (temp, i);

endfor

endfunction

Deszyfrowanie korzysta z tej samej funkcji, nazwanej tutaj DVB.

*Rys.3.1.1: Sygnał przed scramblowaniem*

*Rys.3.1.2: Sygnał po scramblowaniu*

*Rys.3.1.3: Porównanie sygnału z nadajnika i odbiornika [to zerowanie się]*

* 1. V-34

Zgodnie z opisem scramblera multiplikacyjnego w punkcie 2.2.2 jego działanie opiera się na dodawaniu (zarówno w nadajniku, jak i odbiorniku) ciągu, który został wygenerowany przez wielomian h(x), szerzej opisany w punkcie 2.2.

Taki sposób generowania ciągu, w odróżnieniu od scramblera addytywnego, nie wymusza stosowania ciągu synchronizującego (dlatego w literaturze obcojęzycznej scramblery te są często nazywane „samo-synchronizującymi”). [1].

Jednak takie szyfrowanie ciągu danych nie jest równoważne deszyfrowaniu – dlatego stworzyliśmy 2 funkcje: v34 – szyfrującą oraz v34d – deszyfrującą.

Długość użytego tutaj klucza to 23 (co jest determinowane przez wielomian h(x)), podczas gdy, dla przypomnienia, w DVB było to 15.

Funkcja szyfrująca:

function [out] = v34 (key, in)

for i = in

temp = xor (i, xor (key(end), key(5)));

key = shift(key,1);

out(end+1) = temp;

key(1) = temp;

endfor

endfunction

Objaśnienie: key(end) to ostatni element klucza – czyli x23

Funkcja deszyfrująca:

function [out] = v34\_d (key, in)

for i = in

temp = xor (i, xor (key(end), key(5)));

key = shift(key,1);

out(end+1) = temp;

key(1) = i;

endfor

endfunction

Różnicą między obiema funkcjami jest definiowanie nowej wartości pierwszej pozycji klucza – przy szyfrowaniu jest to wynik operacji XOR, przy deszyfrowaniu – jest to kolejny element z zakresu [1, długość ciągu].

Efekty obu funkcji widoczne są na *Rys.3.2.1, Rys.3.2.2* oraz *Rys.3.2.3, Rys.3.2.4*, gdzie pierwsze dwa wykresy dotyczą procesu scramblowania, dwa kolejne – deszyfrowania. Ostatni rysunek, Rys.5, pokazuje nam na ile zgodne są sygnały z nadajnika i odbiornika.

*Rys. 3.2.1: Sygnał przed scramblowaniem*

*Rys. 3.2.2: Sygnał po scramblowaniu*

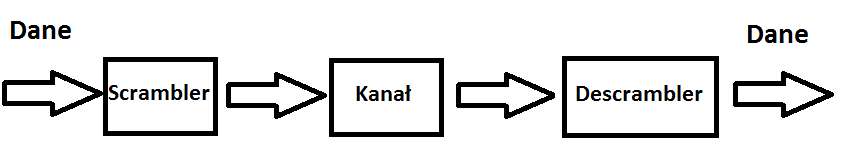
*Rys. 3.2.3: Sygnał przed scramblowaniem*

*Rys. 3.2.4: Sygnał po scramblowaniu*

*Rys. 3.2.5: Porównanie sygnału z nadajnika i odbiornika [to zerowanie się]*

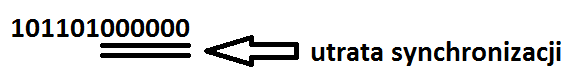
1. **Analiza symulacyjna działania systemu transmisji z możliwością utraty synchronizacji** 
   1. Algorytm działania i model systemu transmisji

Przy transmisji sygnałów cyfrowych istotna jest synchronizacja nadajnika i odbiornika. Długie ciągi jedynek, a szczególnie długie ciągi zer mogą powodować utratę synchronizacji czasowej nadajnika z odbiornikiem. Brak synchronizacji uniemożliwi odbiornikowi odczytywanie poprawnych danych.



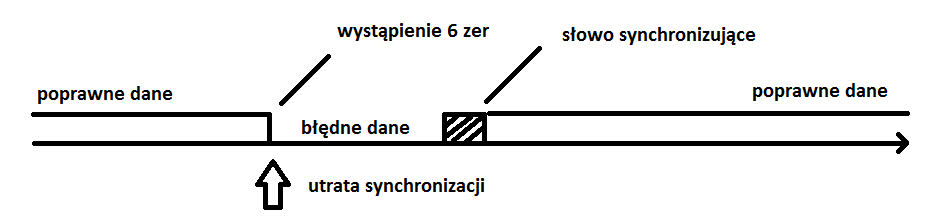
*Rysunek 4.1.Schemat operacji nadania oraz odebrania sygnału.*

Dla symulacji przyjmujemy założenie że do przerwania synchronizacji dochodzi w przypadku gdy w sygnale nadawanym natrafimy na ciąg zer o odpowiedniej długości. W naszych badaniach założyliśmy że ciągi te mają długość 5,6,7 oraz 10.



*Rysunek 4.2. Przypuszczalny ciąg po którym następuje utrata synchronizacji.*

W celu zachowania zsynchronizowanej transmisji i odbioru należy użyć unikalnego słowa synchronizującego. Słowo to jest wzorem umieszczanym w strumieniu danych w jednakowych odstępach (tzn. w każdej paczce). Zadaniem odbiornika jest znalezienie tych słów w sygnale i dokonanie odświeżenia procesu descramblingu po jego wykryciu. Słowo musi być na tyle długie i unikalne, aby uniknąć możliwych pomyłek. Gdyż prowadzi to do utraty danych. Może się zdarzyć, że w strumieniu danych znajdzie się ciąg taki sam jak słowo synchronizujące, przez co zostanie błędnie zinterpretowany. Podobnie wybieramy długość paczek. Nie opłaca się wysyłać krótkich paczek, ponieważ wysyłamy wiele słów synchronizujących co wpływa niekorzystnie na długość transmisji, kiedy szansa na wystąpienie desynchronizacji jest stosunkowo mała.



*Rysunek 4.3.Przykładowa utrata synchronizacji.*

Z kolei wysyłając zbyt długie paczki ryzykujemy utratę danych kiedy synchronizacja wystąpi stosunkowo wcześnie.

* 1. Opis symulatora

Do przeprowadzenia badań utworzyliśmy następujące symulacje:

Generujemy losowy sygnał cyfrowy w postaci tablicy. Sygnał ten jest przekazywany do nadajnika który wykonuje procedurę podziału sygnału, następnie dochodzi do scramblowania owej porcji danych oraz dodanie słowa synchronizacyjnego na końcu tej struktury nazwanej paczką. Dla każdej takiej operacji klucz scramblujący jest przywracany do stanu początkowego. Paczki te łączone są kolejno w sygnał nadawany a następnie przekazywane do odbiornika w którym dochodzi do descramblingu do odbiornika w którym dochodzi do descramblingu.   
Funkcja descramblera jest odświeżana za każdym razem gdy dochodzi do wykrycia klucza synchronizującego.

Aby uzyskać symulacje utraty synchronizacji w nadajniku dokonujemy sprawdzenia kolejnych bitów zescramblowanego sygnału gdy natrafimy na oczekiwaną ilość zer to dokonujemy przesunięcie sygnału o jedną pozycję w prawo zapętlając.

Badanie transmisji sygnału nadajnika DVB oraz v34 przeprowadzono przy następujących założeniach:

Długość sygnału: 500

Wielkość paczki: 100

Ilość zer desynchronizujacych: 6

Wielkość słowa synchronizującego: 10

Badanie stosunku dobroci w zależności od długości paczki dla nadajnika DVB oraz v34 przeprowadzono przy następujących założeniach:

Długość sygnału: 500

Wielkość paczki jest zwiększana od 1 bitu aż do pełnej długości sygnału.

Ilość zer desynchronizujacych kolejno: 5,7,10

Wielkość słowa synchronizującego: 10

Stosunek dobroci:

(4)

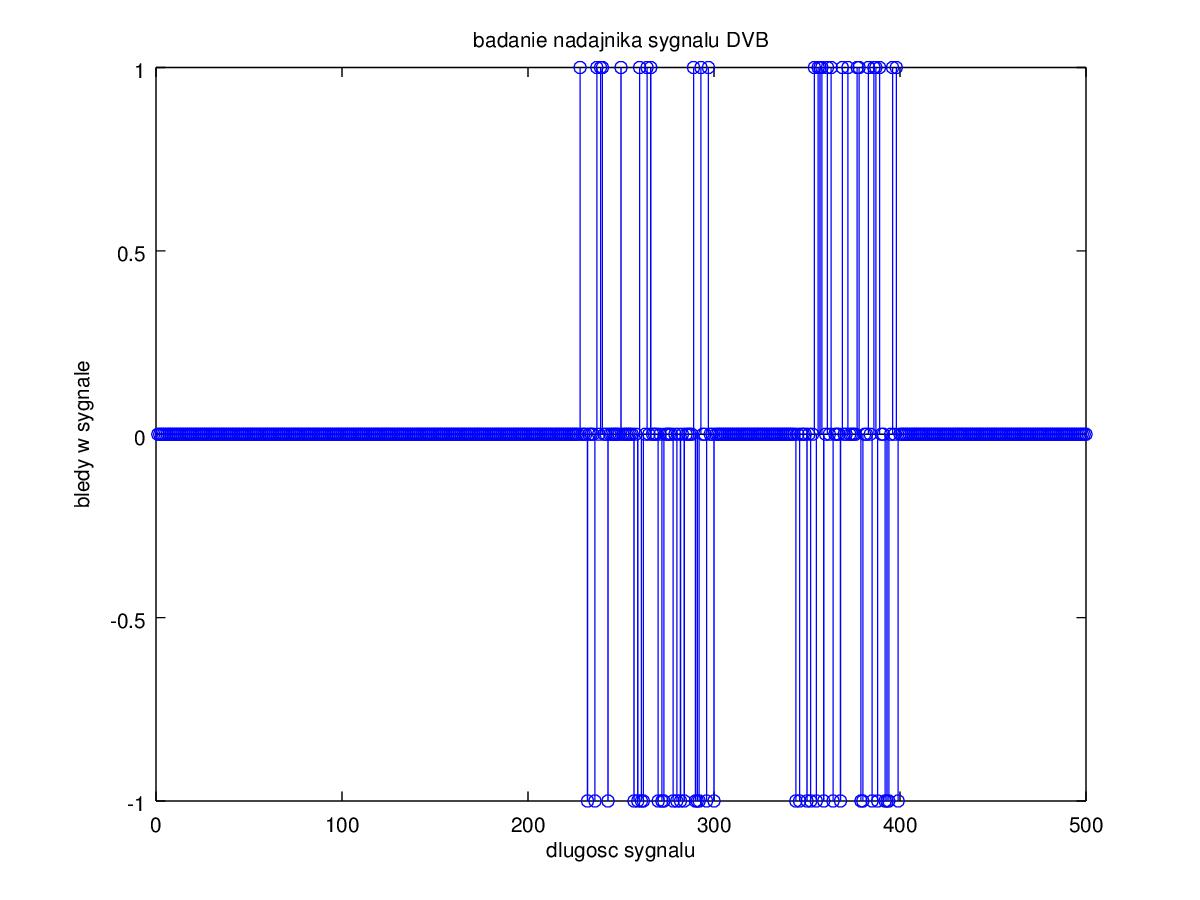
Gdzie:

A - stosunek dobroci

Dobrze –suma prawidłowo przesłane bity.

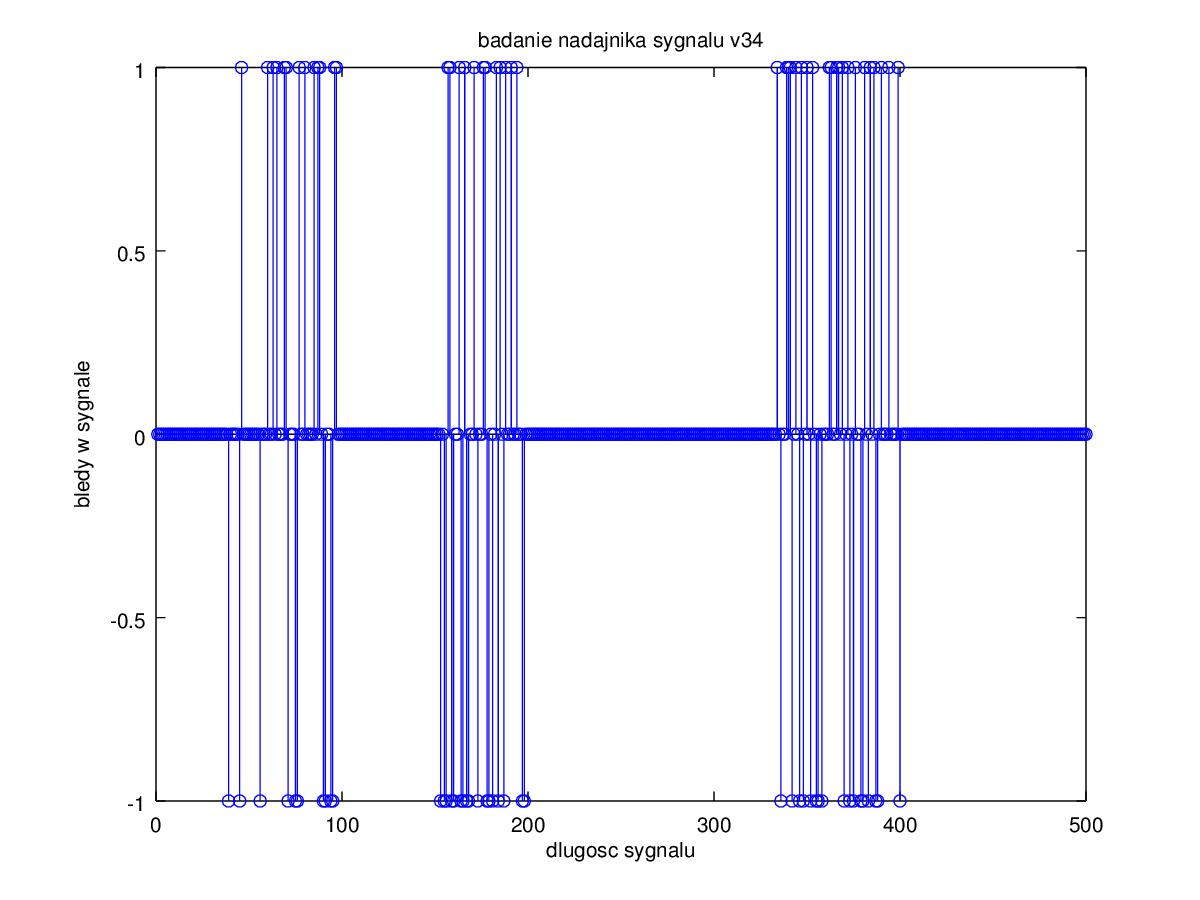
Źle – suma źle przesłanych bitów oraz suma bitów wszystkich zajmowanych przez kluch synchronizujący.

* 1. Wyniki symulacji, ich analiza



*Wykres 4.1.* *Badanie nadajnika sygnału DVB*

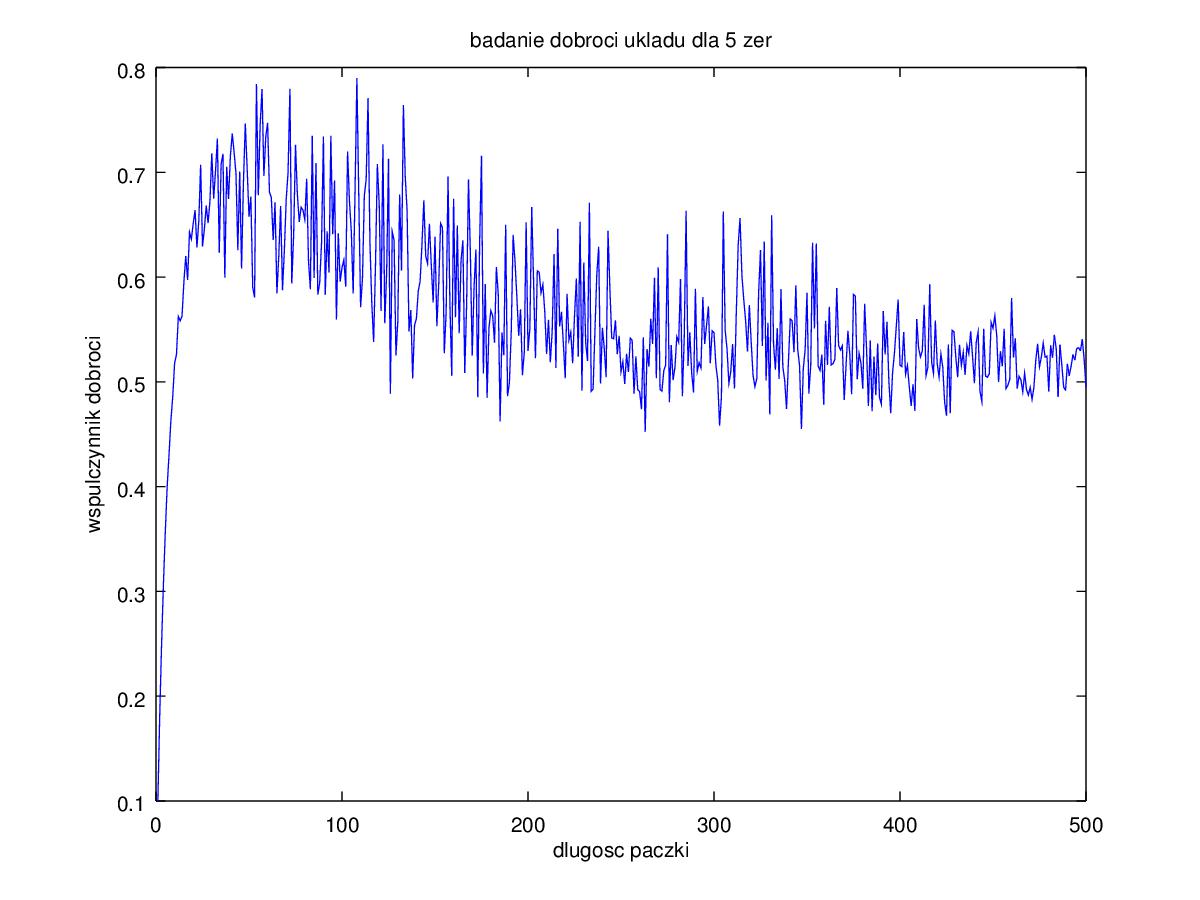
Wyniki badania nadajnika DVB przedstawione są na *Wykres 4.1.* przedstawiona została różnica sygnałów: oryginalnego, oraz odebranego dla scramblera typu DVB. Wyraźnie widoczne jest uszkodzenie sygnału w okolicach bitu 220, które zostaje naprawione od bitu 300. Taka sama sytuacja ma miejsce około bitu 345 a jego naprawę widać już od 400 bitu. W analizowanym przykładzie uszkodzone zostało około 27% sygnału, natomiast dodatkowych bitów, które musiały być przesłane było 50.



*Wykres 4.2.* *Badanie nadajnika sygnału v34*

Wyniki badania nadajnika v34 przedstawione są na *Wykres 4.2.* przedstawiona została różnica sygnałów: oryginalnego, oraz odebranego dla scramblera typu v34. Wyraźnie widoczne są trzy utraty synchronizacji. Do uszkodzenie sygnału dochodzi w okolicach bitu 40, które zostaje naprawione od bitu 100. Następnie sygnał tracony jest na 150 bicie a jego naprawę widać już od 200 bitu. Do ostatniego przerwania dochodzi jak ukazuje to *Wykres 4.2.* na bicie 330 z naprawą na 400. W analizowanym przykładzie uszkodzone zostało około 36% sygnału, natomiast dodatkowych bitów, które musiały być przesłane w transmisji było 50.

Aby dowiedzieć się jaki wpływ ma długość paczki w stosunku do ilości prawidłowo przesłany bitów dokonaliśmy następujące badania:

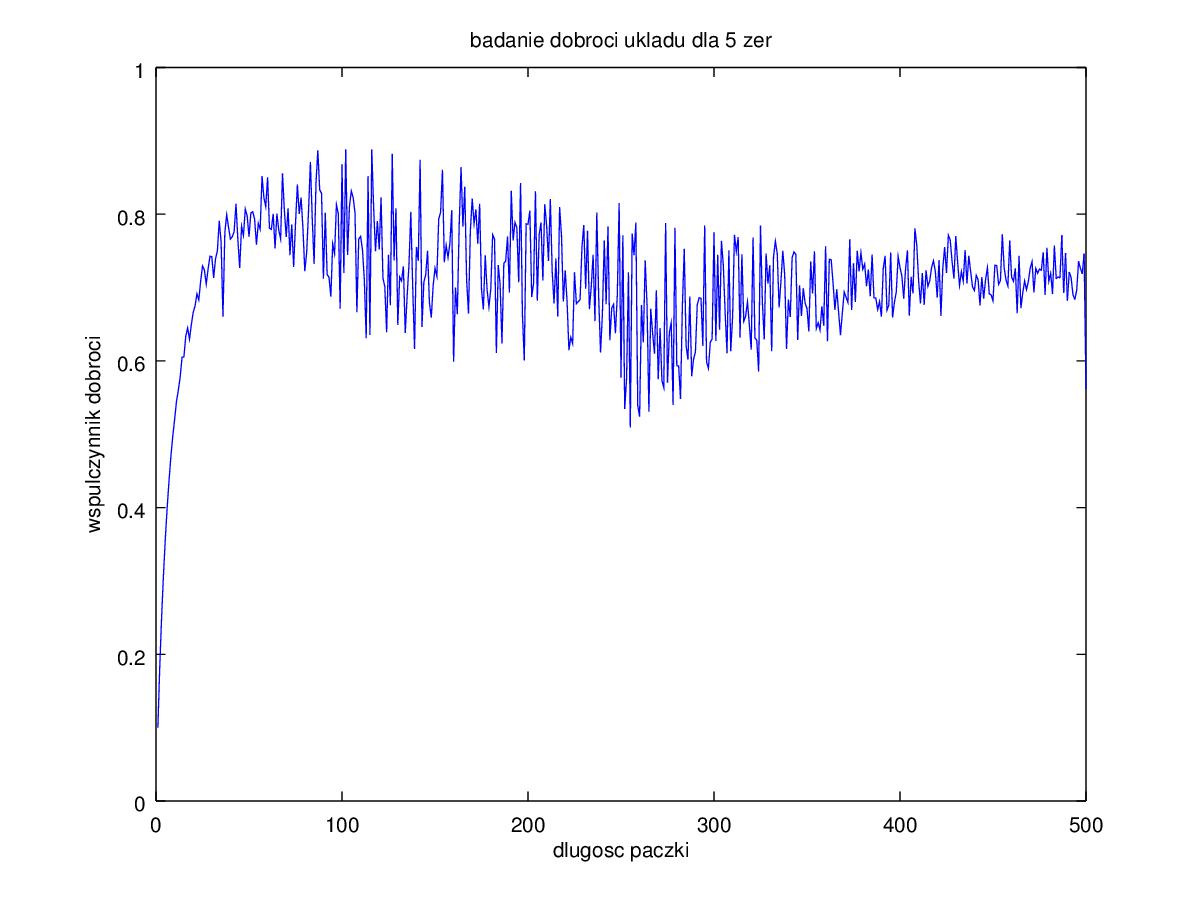


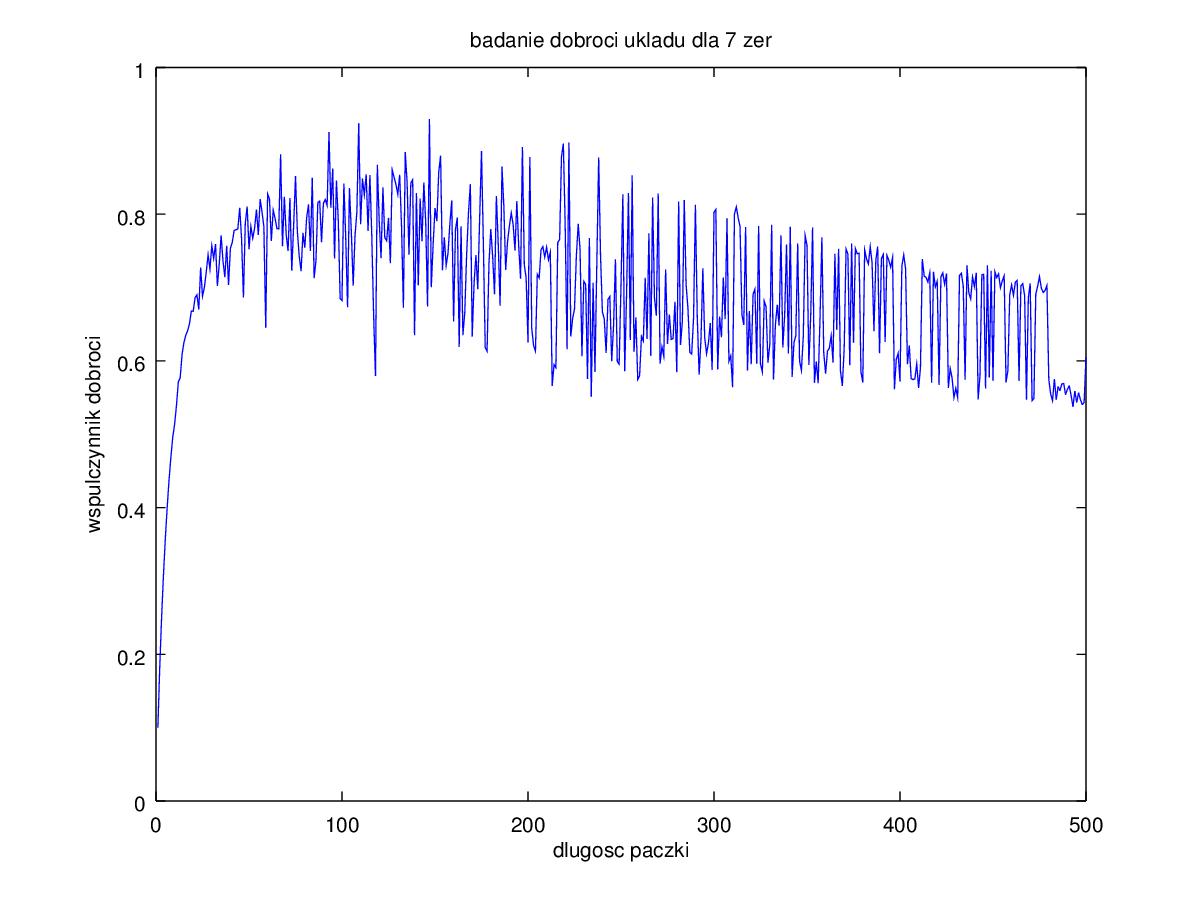
*Wykres 4.3.* *Badanie nadajnika DVB sygnału przy przerwaniu dla 5 zer*

Na *Wykres 4.3.* zostały przedstawione wyniki zautomatyzowanego badania współczynnika dobroci układu dla sygnału w sytuacji gdy utrata synchronizacji odbywa się po przesłaniu pięciu zer. Na poziomej osi przedstawiony jest rozmiar pojedynczej paczki, natomiast na osi pionowej dobroć układu. Jak wskazuje *Wykres 4.3* funkcja w początkowej fazie ma charakter szybko rosnący a następnie osiąga swoje maksimum na przedziale: [40,80] i nie przekracza on wartości około 7,8. w dalszej fazie stosunek zaczyna wolno maleć wraz ze wzrostem rozmiaru paczki.

To samo badanie przeprowadziliśmy dla nadajnika v34.

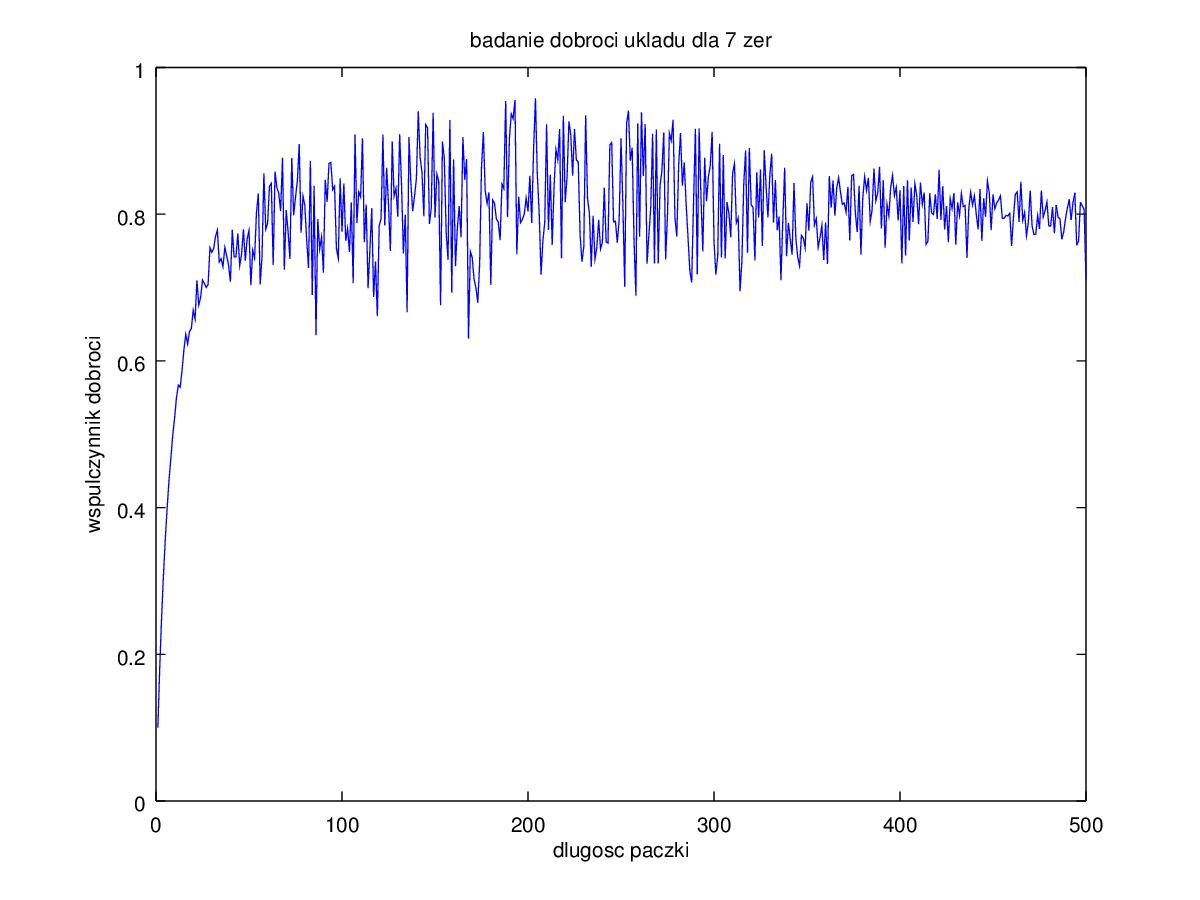
Charakter tej funkcji ukazany na *Wykres 4.3.* jest stosunkowo inny niż ten na *Wykres 4.2.*Tu również możemy zaobserwować szybki przyrost w początkowej fazie funkcji lecz przyrost ten wyhamowuje stosunkowo łagodniej osiągając maksimum w okolicach [80,120] osiągając wartości 0,9 w dalszych etapach funkcja ma tendencje malejącą.



*Wykres 4.3.* *Badanie nadajnika V34 sygnału przy przerwaniu dla 5 zer*

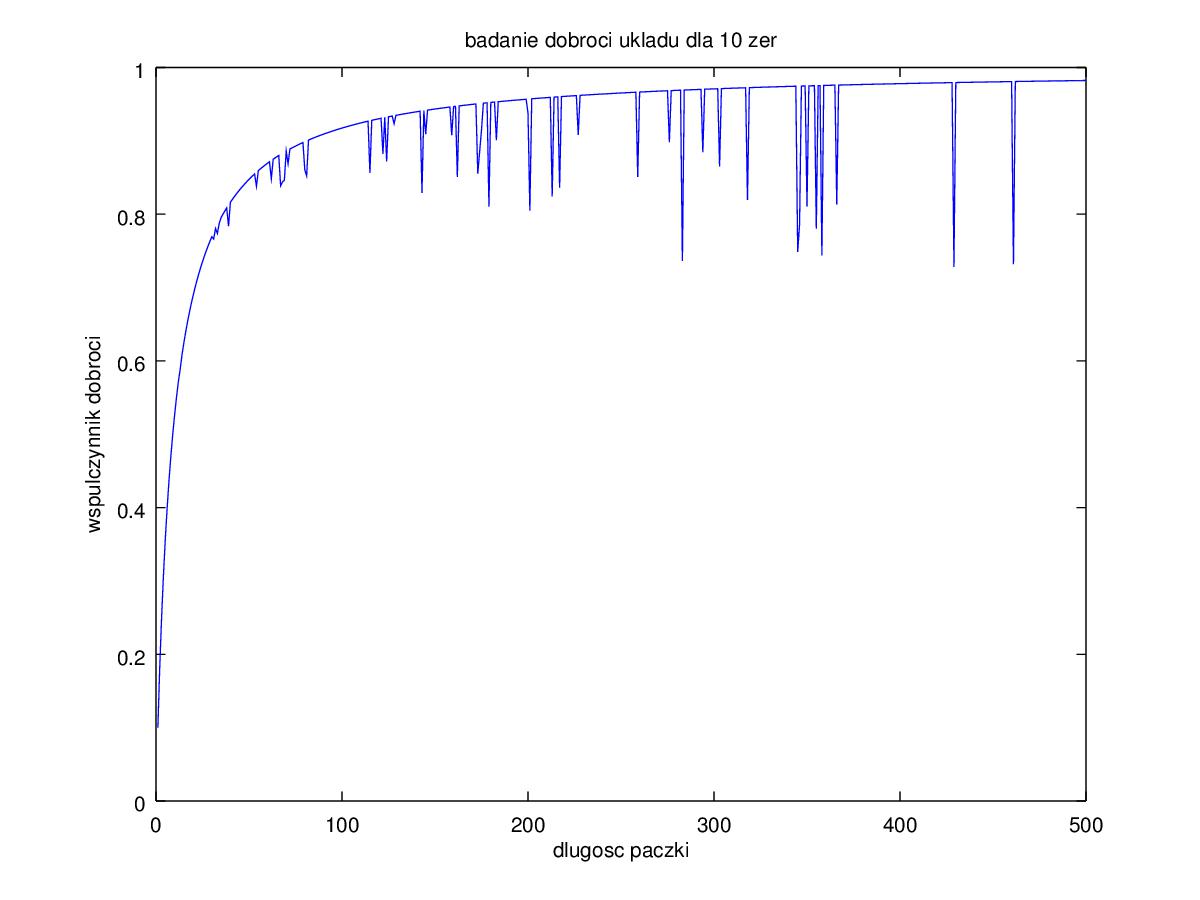
*Wykres 4.4.* *Badanie nadajnika DVB sygnału przy przerwaniu dla 7 zer*

Na *Wykres 4.4.* zostały przedstawione wyniki zautomatyzowanego badania współczynnika dobroci układu dla sygnału w sytuacji gdy utrata synchronizacji odbywa się po przesłaniu siedmiu zer. Jak wskazuje *Wykres 4.3* funkcja w początkowej fazie ma charakter szybko rosnący, a następnie osiąga swoje maksimum na przedziale: [110,140] i nie przekracza on wartości około 8,5. w dalszej fazie stosunek zaczyna wolno maleć wraz ze wzrostem rozmiaru paczki.

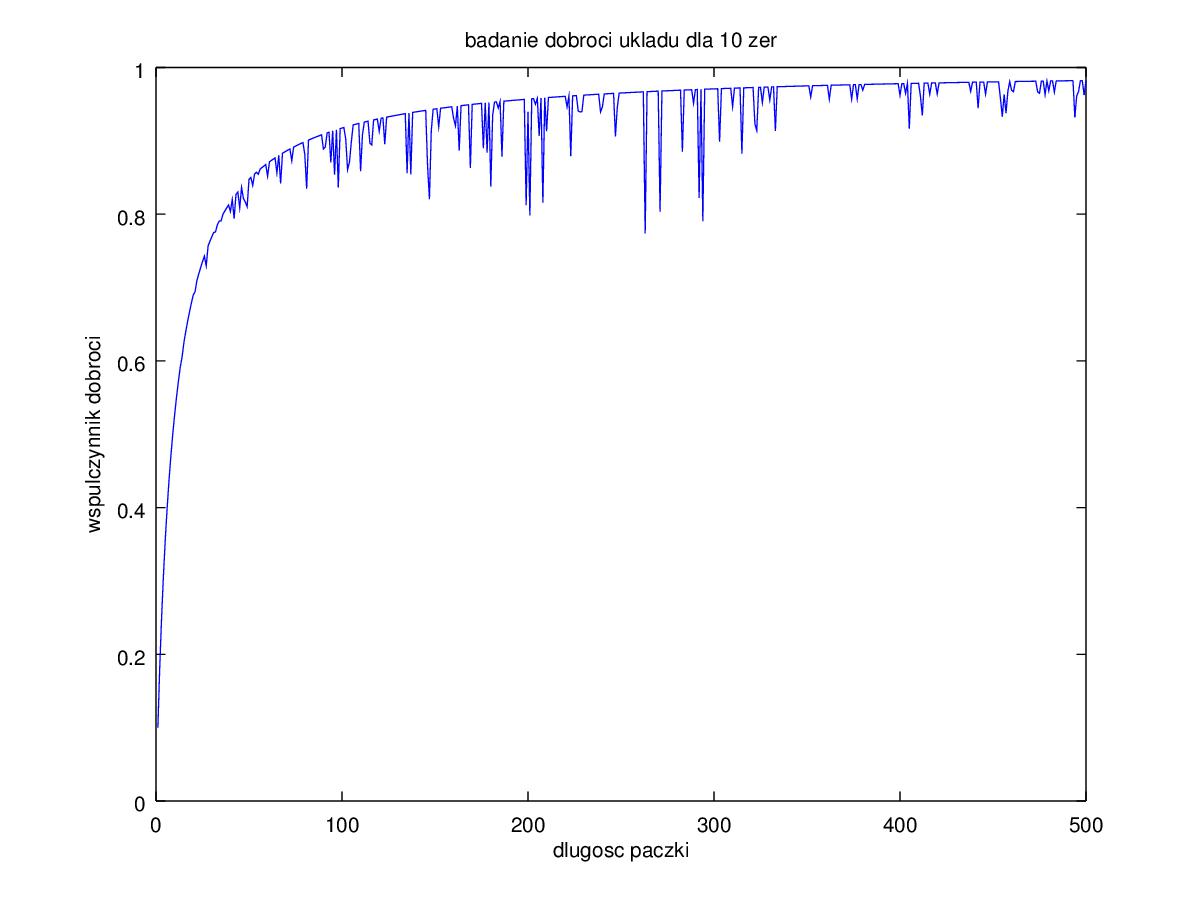


*Wykres 4.5.* *Badanie nadajnika v34 sygnału przy przerwaniu dla 7 zer*

Charakter tego funkcji ukazany na *Wykres 4.5.* jest stosunkowo inny niż ten na *Wykres 4.4.*Tu również możemy zaobserwować szybki przyrost w początkowej fazie funkcji lecz przyrost ten wyhamowuje stosunkowo łagodniej osiągając maksimum w okolicach nawet [180,220] osiągając wartości ponad 0,94 w dalszych etapach funkcja ma tendencje wolno malejącą jest to charakterystyka jak do tej pory o najwolniejszym spadku.



*Wykres 4.6.* *Badanie nadajnika DVB sygnału przy przerwaniu dla 10 zer*

**

*Wykres 4.7.* *Badanie nadajnika v34 sygnału przy przerwaniu dla 10 zer*

Na *Wykres 4.7* oraz *Wykres 4.8* ukazano wyniki ostatniego badania dla przerwania synchronizacji po 10 zerach. Funkcje te mają tendencje rosnącą przez całą swoją dziedzinę. W początkowym etapie rosą one bardzo szybka lecz ten wzrost wyhamowuje wraz z wzrostem rozmiaru paczki. Funkcje te dążą w nieskończoności do wartości jeden.

* 1. Wnioski

Nie jest możliwe zaprojektowanie układu pozbawionego problemu desynchronizacji. Wyniki badań z *Wykres 4.2. oraz Wykres 4.3.* ukazują nam, iż w sytuacji, gdy dochodzi do stosunkowo częstej utraty synchronizacji spowodowanej np. zakłóceniami w przenośniku sygnału, powinniśmy stosować dość częste działania odzyskania synchronizacji, gdyż prawdopodobieństwo przesłania dużej liczby uszkodzonych paczek w dłuższych sygnałach jest niezwykle wysokie.  
Na podstawie badania z *Wykres 4.4. oraz Wykres 4.5.* łatwo wywnioskować, że jednym z głównych ograniczeń co do przesyłania dłuższych sygnałów o mniejszej ilości kluczy synchronizacyjnych jest łatwość z jaką dochodzi do utraty synchronizacji.   
W przeciwieństwie do wyników z *Wykres 4.2. oraz Wykres 4.2.* badania te pokazały iż optymalna długość paczki zamyka się w granicach od 110 do 140 bitów dla nadajnika DVB i od 180 do 220 bitów dla nadajnika v34.

Końcowe badania potwierdzają tą tezę, gdyż jak wynika z *Wykres 4.6. oraz Wykres 4.7* w sytuacji w której nie dochodzi prawie wcale do utraty synchronizacji, stosunek dobroci rośnie wraz z długością paczki. To zaś wskazuje na to, że wstawiania kluczy synchronizacyjnych oraz podział na paczki są zbędne.

1. **Wnioski**
2. **Literatura**

[1] V.34 Transmitter and Receiver Implementation on the TMS320C50 DSP; Texas Intruments

[2] Transmisja Danych, prezentacja Zakładu Sieci Telekomunikacyjnych Instytutu Telekomunikacji i Akustyki na Politechnice Wrocławskiej

[3] Analysis Of The DVB Common Scrambling Algorithm; Ralf-Philipp Weinmann, Kai Wirt

[4]