Niezawodność i diagnostyka układów cyfrowych 2 - projekt

Jan Wawruszczak, 241344
Filip Gajewski, 236597
Andrzej Gierlak, 236411

Dokument projektowy służący opisowi kolejnych etapów realizacji projektu

Część I

Etap 27.03.2019

1 Wybrana tematyka

FEC (forward error correction) - założeniem projektu jest zbadanie wpływów nadmiarowości na poprawność i szybkość przesyłania informacji.
Z założenia wynika, że będziemy chcieli zasymulować działanie kodowania korekcyjnego na możliwość detekcji i korekcji (częściowej lub całkowitej) błędów, które powstają w wyniku zakłóceń związanych z przesyłem danych.
W celu analizy takiego modelu należy go skonstruować i przeprowadzić testy, na których to podstawie można będzie wyciągnąć odpowiednie wnioski.
Wartym zauważenia jest fakt, że z twierdzenia Shanonna-Hartleya wynika, że można z dowolnie niskim prawdopodobieństwem przesyłać informację przy ustalonej prędkości przesyłu. Istotym jest jednak to, że nie wystarczy zmniejszyć prędkości transmisji, aby pozbyć się błędów. Aby osiągnąć niskie prawdopodobieństwo błędu konieczne jest przesyłanie informacji w formie zakodowanej, to znaczy zawierającej pewną nadmiarowość. Niefortunnie, z tej teorii nie wynika w żaden sposób jak takie kodowania konstruować, aby otrzymać zadowalające wyniki zarówno w tempie przesyłu, jak i dokładności przesyłanych danych.

2 Narzędzia pracy

Do stworzenia odpowiedniego modelu wykorzystamy język programowania Python ze względu na rozbudowane biblioteki pozwalające bez przeszkód zwią-

zanych z zawiłą implementacją tworzyć interesujący nas model transmisji. W szczególności, spodziewamy się wykorzystania elementów z bibliotek:

1. random
2. numpy
3. matplotlib
4. codecs

Te i inne narzędzia umożliwią symulację generowania, kodowania, zaszumiania i dekodowania informacji W wytwarzanym przez nas modelu wygląd trasy prezentuje się następująco:

Generator → koder → kanał zaszumiający → dekoder

Jest to w miarę prosta koncepcja na odpowiednim poziomie abstrakcji, ale na tyle praktyczna, że uzyskane wyniki mogą być miarodajne w praktyce.

Część II
Etap 10.04.2019

3 Kod BCH

Kod BCH (Bose–Chaudhuri–Hocquenghem) to kodowanie FEC wykorzystujące wielomiany z ciała skończonego.
Kody BCH należą do kodów cyklicznych, czyli kodów wielomianowych o długości słowa kodowego n, których wielomian generujący g(x) (niezbędny do konstrukcji słowa kodowego) jest dzielnikiem wielomianu $x^n - 1$. Można pokazać, że istnieje taki wielomian $k(x)$ stopia k, że:
$g(x)k(x) = x^n + 1$
lub
$(x^n + 1) \bmod g(x) = 0$
Dla każdej liczby całkowitej m i istnieje kod BCH o długości Może on korygować do t błędów i ma nie więcej niż elementów kontrolnych. Parametry tych kodów:
$n = 2^m - 1$
$k \geqslant n - m * t$
$d_min \geqslant 2 * t + 1$
gdzie:
n - długość wektora kodowanego
k - długość ciągu informacyjnego
d - odległość minimalna
t - zdolność korekcji błędów

3.1 Aglorytm generowania

Jedną z metod generowania jest wykorzystanie wielomianu generującego. Można to sprowadzić do mnożenia wielomianu $m(x)$ odpowiadającemu ciągowi informacyjnemu kodu przez wielomian generujący $g(x)$.
Wtedy wektor kodu źródłowego wygląda następująco:
$C = [m_{n-1}, ..., m_{n-k}, r_{n-k-1}, r_0]$,
gdzie:
m_i - element informacji
r_i - element kontrolny

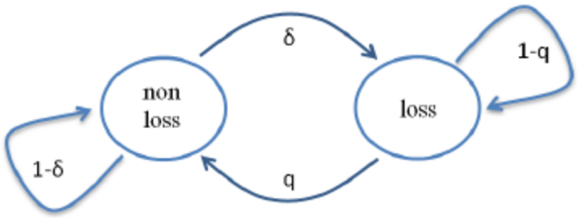
Mając wielomian generujący g(x) stopnia n-k i chcąc wyliczyć wektor kodowy kodu (n,k) postępujemy w poniżej przedstawiony sposób:
$x^n - k * m(x)/g(x) = q(x) + r(x)/g(x)$
i mnożąc obustronnie razy $g(x)$ otrzymamy:
$x^n - k * m(x) = q(x) * g(x) + r(x)$
gdzie:
$m(x)$ - ciąg informacyjny; $m(x) = m_{k-1} * x^{k-1} + ... + m_1 * x + m_0$
$x^{n-k} * m(x)$ - ciąg informacyjny przesunięty w lewo o n-k pozycji w postaci; $m(x) * x^{n-k}$ $m_{k-1} * x^{n-1} + ... + m_1 * x^{n-k+1} + m_0 * x^{n-k}$
g(x) - wielomian generujący;
q(x) - wynik z dzielenia (część całkowita);
r(x) - reszta z dzielenia;
c(x) - wielomian odpowiadający wektorowi kodowemu, powstaje przez dodanie reszty z dzielenia do $k^{n-k} * m(x)$
Za pomocą poniższej zależności można przedstawić część informacyjną wektora kodowego (lewa strona) oraz część kontrolną wektora kodowego (strona prawa):
$\sum_{i=1}^{k} m_{n-i} * x^{n-i} = \sum_{i=1}^{n-k} r_{n-k-i} * x^{n-k-i} \mod g(x)$

3.2 Zalety kodów BCH

Jedną z kluczowych cech kodów BCH jest to, że podczas projektowania kodu istnieje precyzyjna kontrola liczby błędów symboli możliwych do skorygowania przez kod. W szczególności możliwe jest zaprojektowanie binarnych kodów BCH, które mogą poprawić błędy wielu bitów. Kolejną zaletą kodów BCH jest łatwość ich dekodowania, a mianowicie metoda algebraiczna znana jako *dekodowanie syndromu*.
Dekodowanie syndromu jest skuteczną metodą dekodowania kodu liniowego na kanale zaszumiającym, tj. takim, na którym popełniane są błędy. W istocie, dekodowanie syndromu to dekodowanie minimalnej odległości przy użyciu zmniejszonej tabeli odnośników. Jest to poprawne ze względu na liniowość kodu.
Ze względu na wydajnosć kodu możliwe jest dekodowanie sprzętem o niewielkiej mocy.

4 Model Gilberta

Do analizy działania transmisji danych określimy model kanału z zakłóceniami. Ważnym jest to, aby zastosowany model uwzględniał fizyczną naturę błędów, tj. nie tylko psuł niektóre bity, ale też psuł je w sposób porównywalny do zaszumień w świecie rzeczywistym - często występują serie błędów, a nie tylko błędy pojedyńcze. Modele takie bardzo często opierają się na łańcuchach Markowa.
My wykorzystamy tzw. model Gilberta w wersji dwustanowej - możliwe stany to dobry i zły. W stanie dobrym prawdopodobieństwo błędu (popsucia się bitu) jest bardzo niskie, w okolicach 0 (choć niezerowe). W stanie złym wystąpnienie błedu jest znacząco wyższe niż w stanie dobrym. Co więcej, przejście z jednego stanu w drugi obarczone jest pewnym prawdopodobieństwem - dzięki temu możliwe jest symulowanie serii błędów.



Rysunek 1: Model Gilbera