

Codificação de Canal (Setembro 2017)

Dylan Nakandakari Sugimoto, Estudante, ITA, Ivan Monteiro Padalko *Estudante, ITA.*

Abstract—

Index Terms—Codificação, Canal, Hamming, code

I. INTRODUÇÃO

Este documento é um estudo do algoritmo de...

II. MEDIDAS, ANÁLISE E INDICADORES

A. Descrição do algoritmo de codificação de Hamming

No algoritmo de codificação de Hamming ...

B. Descrição do algoritmo de codificação da adaptação do código de Hamming

Adaptou-se o código de Hamming para que o vetor transmitido tivesse 11 bits de informação e 4 de verificação de paridade, de forma que a taxa fosse 73%, que é maior que a taxa do código de Hamming (57,14%)...

C. Descrição do algoritmo de codificação que corrige erro duplo.

Percebendo-se a necessidade de detectar e corrigir os casos em que ocorre dois erros no vetor transmitido, erro duplo, codificou-se um algoritmo de codificação que possui a mesma estrutura lógica do código de Hamming, mas adaptado para operar com 11 bits de informação e 8 bits de paridade, sendo, assim chamado de código 1911. Portanto, a taxa é de 57,89% que é maior que a taxa do código de Hamming. Dessa forma, as únicas diferenças desse código com o código anterior é a forma da matriz de codificação, e, consequentemente, a forma da matriz geradora; e o algoritmo que encontra o vetor erro a partir da síndrome.

Assim, utilizou-se da estratégia de encontrar primeiro a matriz de codificação para a partir dela encontrar a matriz geradora. Para encontrar, a matriz de codificação escolheu-se para efeito de simplicidade manter a distribuição de paridade dos quatro bits que já existiam (bit 12, 13, 14 e 15), ou seja, o bit 12 continuou sendo o bit de paridade do seguinte conjunto de bits 1,2,3,4,5,6 e 7; o bit 13, do seguinte conjunto de bits

1,2,3,4,8,9 e 10; o bit 14, do seguinte conjunto de bits 1,2,5,6,8,9 e 11; o bit 15, do seguinte conjunto de bits 1,3,5,7,8,10 e 11, que resulta na Tabela A. E escolheu-se a seguinte lógica de atribuição de conjunto de bits para os outros quatro bits de paridade (bit 16,17,18 e 19). Na Tabela A, aonde tem 1 trocou-se por 11, aonde tem 2 trocou-se por 10, aonde tem 3 trocou-se por 9, e assim por diante, obtendo-se a Tabela B, e a matriz de codificação (H^T , equação 2) e a matriz geradora abaixo (G , equação 1).

$$G = \begin{bmatrix} 1,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,1,1 \\ 0,1,0,0,0,0,0,0,0,0,1,1,1,0,0,1,0,1 \\ 0,0,1,0,0,0,0,0,0,0,1,1,0,1,0,1,1,0 \\ 0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,1,1,1 \\ 0,0,0,0,1,0,0,0,0,0,1,0,1,1,1,0,0,1 \\ 0,0,0,0,0,1,0,0,0,0,1,0,1,0,1,0,1,0 \\ 0,0,0,0,0,0,1,0,0,0,1,0,0,1,1,0,1,1 \\ 0,0,0,0,0,0,0,1,0,0,0,1,1,1,1,1,0,0 \\ 0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,1,0,1 \\ 0,0,0,0,0,0,0,1,0,0,0,1,1,0,1,1,1,0 \\ 0,0,0,0,0,0,0,0,1,0,0,1,0,1,1,1,1,0 \\ 0,0,0,0,0,0,0,0,0,1,0,0,1,1,1,1,1,1 \end{bmatrix} \quad (1)$$

A ideia é que se ocorrer um erro, os conjuntos de bits de paridade da Tabela A e da Tabela B apontam para o mesmo erro de um bit. Por exemplo, se ocorrer erro no bit 1, então, ambos os conjuntos apontam para erro no bit 1. Porém, quando ocorrer erro duplo, os quatro primeiros bits de paridade indicam erro em um determinado bit, e os outros quatro bits de paridade indicam erro em outro bit o que sinaliza que ocorreu um erro duplo na transmissão. Para encontrar, qual é o vetor erro, que tenta consertar o vetor recebido, realiza-se a soma módulo 2 da síndrome com uma linha da matriz H^T , verifica-se se o resultado é uma linha da matriz de codificação, se pertencer, então o índice das duas linhas que foram somadas indicam os bits que precisam ser consertados. Se não pertencer, realiza-se a soma anteriormente citada para a próxima linha da matriz de codificação. Se após realizado a soma para todas as linhas da matriz de codificação, o resultado da soma não ter sido uma linha da matriz de codificação, então atribui-se o valor zero para o vetor de correção, ou seja, não se tenta corrigir o vetor de transmissão.

É fato que essa codificação não conserta todas as combinações possíveis de erros duplos, porém os dados coletados mostraram que a quantidade de erros duplos que essa codificação conserta foi suficiente para obter o resultado desejado, como se mostra na secção D (apresentação e análise dos dados obtidos).

S. N. Dylan é estudante no Instituto Tecnológico de Aeronáutica, São José dos Campos, SP 12228-900 BRL (e-mail: dylan-ns@hotmail.com).

P. M. Ivan, é estudante no Instituto Tecnológico de Aeronáutica, São José dos Campos, SP 12228-900 BRL.

$$H^T = \begin{bmatrix} 1,1,1,1,0,0,1,1 \\ 1,1,1,0,0,1,0,1 \\ 1,1,0,1,0,1,1,0 \\ 1,1,0,0,0,1,1,1 \\ 1,0,1,1,1,0,0,1 \\ 1,0,1,0,1,0,1,0 \\ 1,0,0,1,1,0,1,1 \\ 0,1,1,1,1,1,0,0 \\ 0,1,1,0,1,1,0,1 \\ 0,1,0,1,1,1,1,0 \\ 0,0,1,1,1,1,1,1 \\ 1,0,0,0,0,0,0,0 \\ 0,1,0,0,0,0,0,0 \\ 0,0,1,0,0,0,0,0 \\ 0,0,0,1,0,0,0,0 \\ 0,0,0,0,1,0,0,0 \\ 0,0,0,0,0,1,0,0 \\ 0,0,0,0,0,0,1,0 \\ 0,0,0,0,0,0,0,1 \end{bmatrix} \quad (2)$$

Tabela A :a primeira linha contém dos quatro primeiros bits de paridade do código adaptado (11/15), e do código 1911, em baixo de cada um desses bits de paridade estão os bits que são considerados na soma da paridade daquele bit de paridade. Por exemplo, o bit 12 é a soma módulo 2 dos bits de 1 até 7 do vetor transmitido.

Bit 12	Bit 13	Bit14	Bit15
1	1	1	1
2	2	2	3
3	3	5	5
4	4	6	7
5	8	8	8
6	9	9	10
7	10	11	11

Tabela B: a primeira linha contém dos quatro últimos bits de paridade do código 1911, em baixo de cada um desses bits de paridade estão os bits que são considerados na soma da paridade daquele bit de paridade. Por exemplo, o bit 12 é a soma módulo 2 dos bits de 1 até 7 do vetor transmitido.

Bit 12	Bit 13	Bit14	Bit15
11	11	11	11
10	10	10	9
9	9	7	7
8	8	6	5
7	4	4	4
6	3	3	2
5	2	1	1

D. Apresentação e análise dos dados obtidos.

Gerou-se um milhão de bits utilizando uma função pseudo-aleatória já implementada em Python da biblioteca numpy, que seleciona de forma uniforme os números inteiros de um dado intervalo, nesse caso, o intervalo contém apenas os números 0 e 1, e organizou-se esses bits em vetores de 4 bits, que foram passados na função que simula o canal BSC. Assim, a quantidade de bits alterados após a passagem pelo canal é a quantidade de erro causados pelo canal e a probabilidade de erro sem codificação é essa quantidade de erro dividido por um milhão, que se apresenta na cor amarela na Figura A. O mesmo foi feito com a codificação de Hamming, a menos da diferença que na codificação de Hamming, codificou-se o vetor de quatro bits com a codificação Hamming, ou seja, os vetores transmitidos são de 7 bits em que 4 bits são de informação e 3 bits são de verificação de paridade. E após a passagem pelo canal BSC, fez-se a decodificação e apenas após a decodificação é que se contabilizou os erros presentes nos bits de informação. Igualmente para o caso sem codificação fez esse processo para vários valores de probabilidade de erro de um bit, começando com probabilidade de 50% de erro de um bit após passagem pelo canal BSC até a perda da capacidade de medição, que é quando a probabilidade de erro vai a zero, ou seja, a decodificação consegue consertar todos os erros que aparecem após passagem pelo canal BSC, e assim, não se contabiliza nenhum erro. Esse momento da perda da capacidade de medida está marcado na Figura A com uma reta quase vertical.

A mesma situação de repete para a codificação adaptada do código de Hamming para transmissão de 11 bits de informação e 4 bits de paridade, ou seja, gerou-se 999.999 bits aleatoriamente, e organizou-se em vetores de 11 bits. Porém, observa-se que a probabilidade de erro da codificação adaptada (curva vermelha da Figura A), é maior para quase todos os valores de probabilidade de erro de um bit, a exceção de um ponto que ocorre para probabilidade de erro de um bit igual à $2 \cdot 10^{-4}$, que pode ser desconsiderado devido à alta variância da medida para esse caso, o que significa que depende da medida, esse ponto pode ficar tanto acima quanto a baixo da curva azul, pois quantidade de erros que ocorre é muito pequena. O principal é compreender que a curva da codificação adaptada de Hamming fica acima da curva do código de Hamming porque o vetor transmitido no caso da

codificação adaptada é de tamanho 15 enquanto que no código de Hamming, o tamanho do vetor transmitido é 7 o que implica pelo fato da quantidade de erros que ocorre após passagem pelo canal BSC ser uma distribuição binomial que o vetor transmitido de tamanho maior tem probabilidade de erro múltiplo, que é quando ocorre mais de um erro no vetor transmitido, maior do que o vetor menor, como se observa na Figura C. Assim, como o código adaptado conserta apenas um erro, da mesma forma que o código de Hamming só conserta um erro, tem-se que a probabilidade de erro da codificação adaptada (curva vermelha) é maior que a probabilidade de erro da codificação de Hamming, apesar de ter se observado mais erros ocorrerem após passagem pelo canal e antes da decodificação para o código de Hamming do que para o código adaptado, como se observa na Figura B. Assim, em termos de probabilidade de erro, a codificação de Hamming é melhor do que a adaptada.

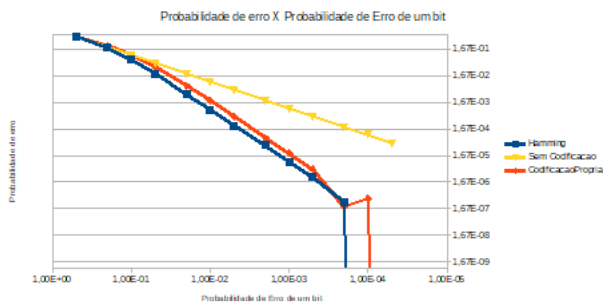


Figura A: O eixo y é a probabilidade de erro e o eixo x é a probabilidade de erro de um bit após passagem pelo canal BSC. Ambos os eixos estão em escala logarítmica para melhor visualização. As retas quase verticais indicam o momento da perda da capacidade de medida.

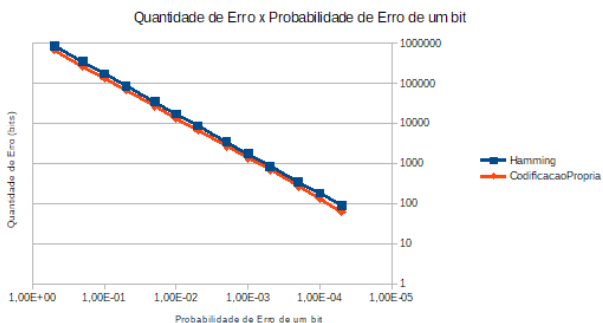


Figura B: Quantidade de erro total após passagem pelo canal BSC e antes da decodificação por probabilidade de erro de um bit. Em vermelho, a codificação adaptada para transmitir 11 bits de informação e 4 bits de paridade; e, em azul, a codificação de Hamming.

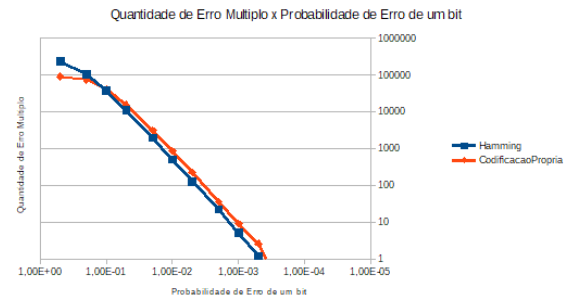


Figura C: Quantidade de erro múltiplo após passagem pelo canal BSC e antes da decodificação por probabilidade de erro de um bit. Erro múltiplo acontece quando ocorre mais de um erro no vetor transmitido. Em vermelho, a codificação adaptada para transmitir 11 bits de informação e 4 bits de paridade; e, em azul, a codificação de Hamming.

Dessa forma, percebeu-se que para se ter uma melhor performance do que o código de Hamming em termos de probabilidade de erro, é necessário que a codificação utilizada consiga detectar e corrigir dois erros na etapa de decodificação, se o vetor transmitido tiver tamanho maior do que o código de Hamming, que é o desejado neste estudo. Assim, adicionou-se mais quatro bits de paridade para tentar realizar a detecção e correção de erros duplos, que ocorre quando há dois erros no vetor transmitido, chegando-se assim ao código 1911 (em verde na Figura D), que não corrige todos os erros duplos possíveis, mas o subconjunto dos erros duplos que esse código corrige é suficiente para ter uma performance melhor do que o código adaptado, praticamente, para todo valor de probabilidade de erro de um bit, e do que o código de Hamming, a partir de $5 \cdot 10^{-4}$, como se observa na Figura D.

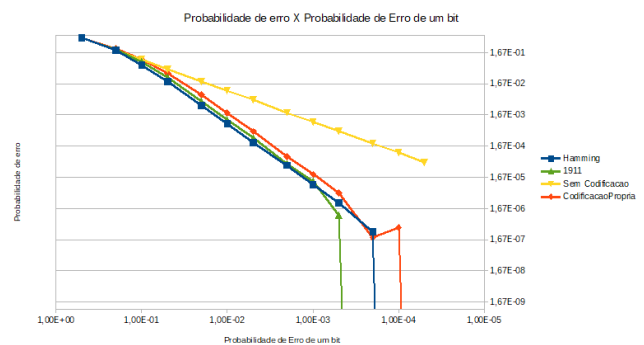


Figura D: O eixo y é a probabilidade de erro e o eixo x é a probabilidade de erro de um bit após passagem pelo canal BSC. Ambos os eixos estão em escala logarítmica para melhor visualização. As retas quase verticais indicam o momento da perda da capacidade de medida.

Na Figura D, tem-se a relação entre o tamanho do bloco e o desempenho, ou seja, dividiu-se os valores de probabilidade de erro encontrados e dividiu-se pelo tamanho do vetor transmitido. Nesse caso, a curva da codificação 1911 ficou menor que as demais, pois o seu vetor transmitido é maior (19 bits), e a sua probabilidade de erro não é tão maior do que as

demais codificações proporcionalmente ao tamanho do vetor. No caso da codificação adaptada de Hamming para transmissão de 15 bits, inicialmente o valor é menor do que o Hamming 4/7 porque o seu vetor é maior e as probabilidades de erros são bem próximas (vide Figura A). Depois, essas duas curvas se aproximam, pois a probabilidade de erro da codificação Hamming diminui mais do que a codificação adaptada. Assim, observou-se que quanto maior o tamanho do vetor maior é a probabilidade de erro, ou seja, menor é o desempenho para códigos que corrigem apenas um bit de erro, e supondo o canal seja BSC. Isso ocorre porque a probabilidade de erro múltiplo aumenta com o aumento do tamanho do vetor, como se observa na Figura F. Assim, é necessário aumentar a quantidade de erros corrigíveis ao aumentar o tamanho do vetor para se ter um desempenho melhor, como ocorreu no código 1911 em que se corrigiu um subconjunto dos erros duplos além dos erros simples, quando ocorre apenas um erro.

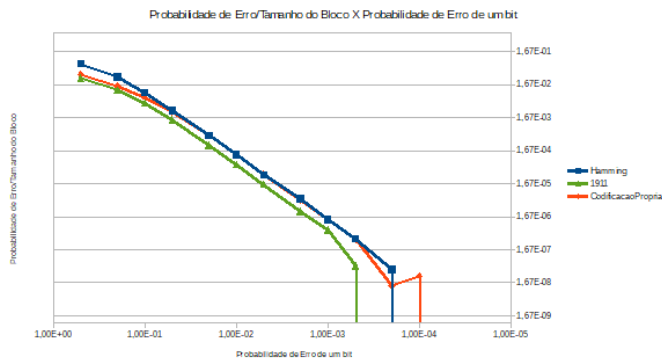


Figura E: As curvas desse gráfico se referem aos valores da Probabilidade de erro de cada codificação dividido pelo tamanho do vetor codificado transmitido por valor de probabilidade de erro de um bit para o canal BSC. Em azul, a codificação Hamming (4/7), em vermelho, a codificação adaptada (11/15), e em verde a codificação 1911.

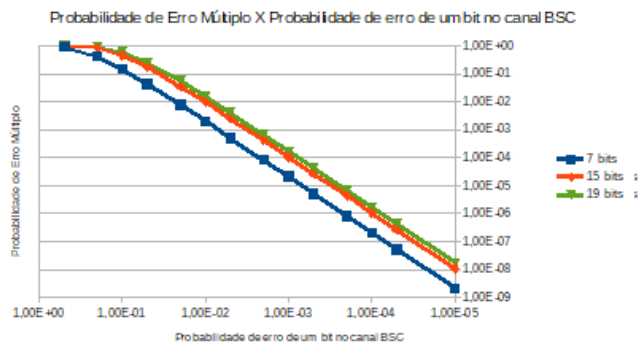


Figura F: Probabilidade de erro múltiplo para diferentes tamanhos de vetores transmitidos por probabilidade de erro de um bit ao passar pelo canal BSC.

Na Figura G, percebe-se que as três primeiras medidas de probabilidade de erro realizadas foram feitas com taxa maior do que a capacidade do canal o que significa que não existe uma codificação que consiga diminuir a probabilidade de erro para um valor tão baixo quanto se queira, mas depois da

terceira medida, a taxa fica menor do que a capacidade do canal o que significa que existe uma codificação que diminua a probabilidade para tão baixo quanto se queira.

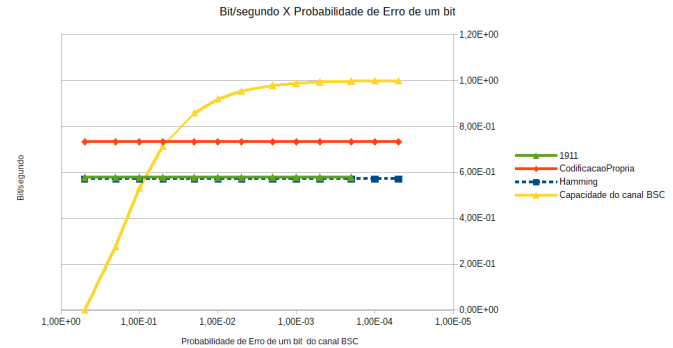


Figura G: As curvas em vermelho, azul e verde são as taxas, quantidade de bits de informação por tamanho do vetor transmitido, e a curva em amarelo é a capacidade do canal BSC. O eixo das abscissas está em escala logarítmica.

III. CONCLUSÃO

Este documento apresenta resultados sobre ...

APÊNDICE

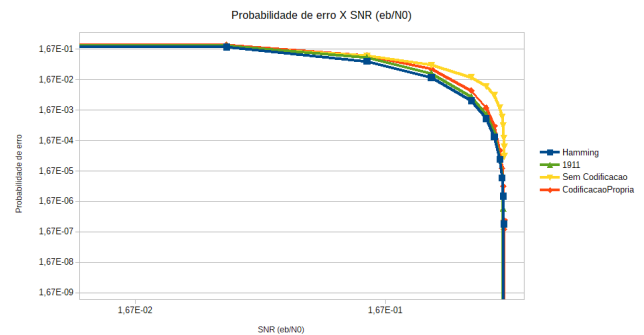


FIGURA A1: PROBABILIDADE DE ERRO POR RELAÇÃO SINAL RUÍDO. OS EIXOS ESTÃO EM ESCALA LOGARÍTMICA.

REFERÊNCIAS BIBLIOGRÁFICAS

[1]

Dylan Nakandakari Sugimoto estudante do 3rd ano do curso de engenharia da computação no Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brasil.

Ivan Monteiro Padalko estudante do curso de engenharia da computação no Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brasil.

