

Controlo de Erros em Comunicações Digitais

22/12/2022

Aluno: Bruno Teixeira (98244) bmteixeira@ua.pt

Aluno: João Moreira (95292) joaomoreirarafael@ua.pt

**Indice**

* Introdução (pág 3)
* Canal discreto com independência estatística (pág 3)
* Avaliação da Simulação (pág 4)
  + - Comparação entre a probabilidade de erros não codificados e probabilidade de erros introduzidos no canal para Hamming (15,11), (31,26), (63,57) (pág 4)
    - Probabilidade de erro teórico em função do SNR para Hamming (15,11), (31,26), (63,57) e para r=1 (pág 6)
    - Comparação entre probabilidade de erros sem codificação e probabilidade de erros introduzidos no canal para Hamming(31,26), BCH(31,21) e BCH(31,16) (pág 7)
    - Probabilidade de erro teórico em função SNR para Hamming(31,26), BCH(31,21),BCH(31,16) e para r=1(pág 8)
    - Ganho para uma probabilidade de erro de 10-2 (pág 9)
    - Ganho para uma probabilidade de erro de 10-4 (pág 9)
* Canal discreto com dependência estatística (pág 10)
  + - Probabilidade de erro introduzido no canal em função P(1/1) com valores (0,0.1,0.3,0.5) para os códigos Hamming(31,26), BCH(31,21) e BCH(31,16) (pág 11)
    - Probabilidade de erro introduzido no canal em função de P(1/1) com valores (0,0.1,0.3,0.5) para os códigos Hamming(31,26), BCH(31,21) e BCH(31,16) usando o interleaving e o deinterleaving em comparação entre o obtido anteriormente(pág 13)

**Introdução**

Este trabalho tem como objetivo estudar a capacidade de correção de erros de alguns códigos nomeadamente de Hamming e código BCH. Para estes códigos vamos observar os efeitos dos códigos com independência estatística e dependência estatística, respetivamente. No último caso vamos observar os efeitos de interleaving/deinterleaving, isto é, os resultados devido a erros sucessivos nos canais de comunicação.

**Canal discreto com independência estatística**

**Uma imagem com texto, captura de ecrã, dispositivo

Descrição gerada automaticamente**

Figura 1 - Simulação com independência estatística

Para as seguintes alíneas vamos prosseguir o desenvolvimento das simulações face ao esquema da *figura 1* do qual representa um sinal inicial “Data Source” do qual será codificado pela codificação de Hamming e BCH, respetivamente, sendo introduzido um ruido ao sinal que estaremos a trabalhar.

A seguinte tabela contém os códigos simulados bem como a sua taxa de código, tendo sido usado a equação .

|  |  |
| --- | --- |
| Código | r |
| Uncoded | 1 |
| Hamming(15,11) | 0.73 |
| Hamming(31,26) | 0.84 |
| Hamming(63,57) | 0.90 |
| BCH(31,21) | 0.68 |
| BCH(31,16) | 0.52 |

Foi considerado que o código é não codificado quando o número de bits antes da codificação equivale o número de bits após a codificação, ou seja, não existe bits de paridade que possam possivelmente corrigir no evento da ocorrência de erros.

Para a simulação dos vários tipos de códigos binários, fizemos uma função no Matlab que implementa um canal discreto AWGN BPSK com codificação e descodificação tanto do BER como da probabilidade de erro, como mostra a seguinte *figura 1*.

**Avaliação da Simulação**

Inicialmente foram gerados bits aleatórios para diferentes códigos de Hamming(15,11),(31,26) e (63,57) com valores de SNR entre 3 e 8 dB com um step de 0.5 dB.

Para a probabilidade de erro associada aos canais de comunicação utilizamos a expressão: (1)

**Comparação entre a probabilidade de erros não codificados e probabilidade de erros introduzidos no canal para Hamming (15,11),(31,26),(63,57)**

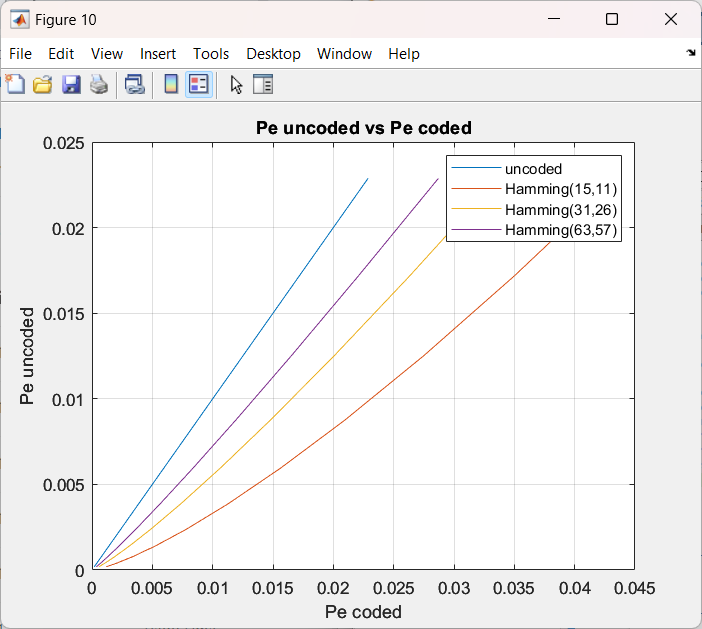


Figura 2 – Comparação das várias probabilidades de erros quando ocorre e não a codificação do sinal enviado no canal.

Para a *figura 2* podemos observar uma diferença do valor das probabilidades de erro para cada codificação, a razão pela qual existe diferenças de probabilidade entre códigos dá-se que para um maior código de Hamming a probabilidade do erro está associada ao comprimento da mensagem codificada enviada, isto deve-se a que quanto menor for a taxa de bit do código, menor será a energia do bit no canal uma vez que para o mesmo canal estaremos a transmitir uma maior quantidade de bits, pela seguinte expressão para a probabilidade da expressão (1) é possível verificar que, com os dados da tabela das taxas de código calculadas, para Hamming com maior “n” a taxa do bit é menor que, por sua vez, diminui a energia do bit que resulta num aumento da probabilidade de erro no canal para a codificação em especifico.

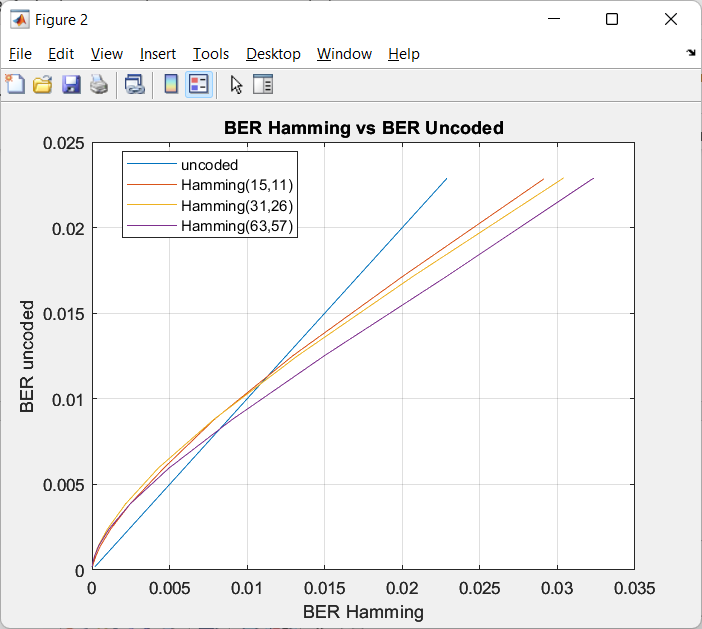


Figura 3- Comparação entre o BER das várias codificações de hamming e um sinal não codificado.

Para a *figura 3* podemos observar que à medida que o SNR aumenta o bit error rate (BER) de cada codificação diminui uma vez que não só um maior SNR significa mais energia do sinal, mas também como estas codificações tem a possibilidade de se corrigirem na descodificação isto também influencia positivamente para uma diminuição do BER comparado ao sinal não codificado. O contrário pode ser dito, para SNR baixo existe um menos energia de bit e ,sabendo que para a codificação dos bits é necessário introduzir os bits de paridade, estes também podem sofrer alterações devido ao ruido introduzido, ou seja, a partir de um certo SNR e para menores valores é mais beneficial, em termos de obter menos erros, não codificar o sinal para evitar um acréscimo de bits que não só devido ao erro iram sofrer alterações mas também este tipo de código não ser suficientemente robusto para conseguir corrigir na descodificação.

Após a análise das figuras anteriores podemos concluir que para códigos com menor code rate, m (k-n), estes apresentam menor probabilidade de erro após a descodificação, para SNR consideráveis.

Por outro lado, aumentando demasiado a probabilidade de erro de canal não é conveniente utilizar uma codificação Hamming porque a uma certa probabilidade a codificação aumenta a taxa de erro.

* **Probabilidade de erro teórico em função do SNR para Hamming(15,11),(31,26),(63,57) e para r=1**

Graphical user interface, chart

Description automatically generated

Figura 4 – Probabilidade do erro em função do SNR.

Para o cálculo da probabilidade de erro teórico para os códigos de Hamming utilizou-se a seguinte equação:

Em seguida traçou-se as curvas de probabilidade em função do SNR, tendo sido obtido as curvas da *figura 4*.

Analisando essa mesma figura, verificamos que para um SNR baixo a probabilidade de erro dos códigos de Hamming é superior à probabilidade de erro do sinal não codificado(uncoded). Isto deve-se à capacidade de correção de um erro do Hamming, que para relações de sinal-ruído baixas a probabilidade de existir mais do que um erro é elevado, portanto a codificação de Hamming não são capazes de corrigir corretamente um maior número de palavras.

Por outro lado, para uma relação sinal-ruido alto a probabilidade de erro é inferior ao sinal não codificado, levando assim à uma introdução de menos erros no canal, podendo assim a codificação de Hamming descodificar corretamente a palavra.

A *figura 4* ainda demonstra, para os casos analisados, que quanto maior for a distância de Hamming, m, em 2m-1 (Hamming (63,57), m=6), para SNR altos, a probabilidade de erro é inferior comparando com m’s mais baixos como o do Hamming (15,11) que possui um m=3.

* **Comparação entre probabilidade de erros sem codificação e probabilidade de erros introduzidos no canal para Hamming(31,26), BCH(31,21) e BCH(31,16)**

Nesta parte do trabalho realizamos as mesmas simulações que foram feitas anteriormente. Em seguida foi realizado para códigos BCH em comparação com os de Hamming, sendo os BCH códigos cíclicos.

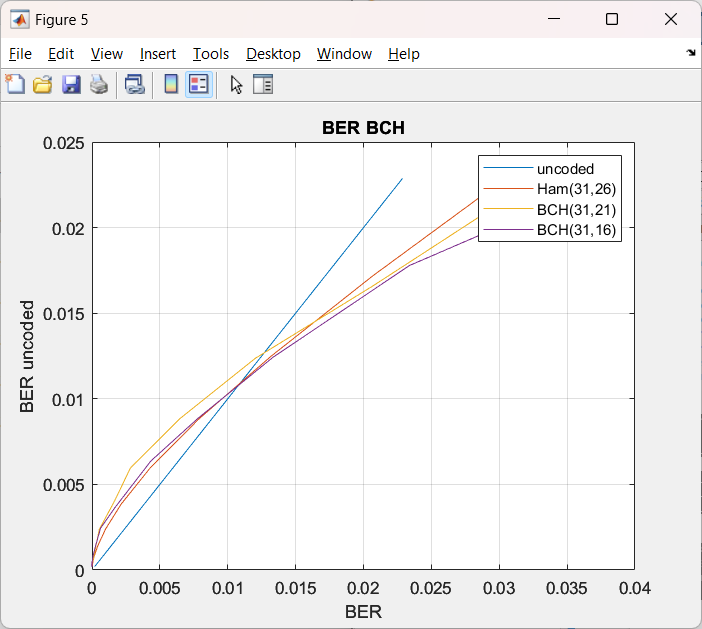
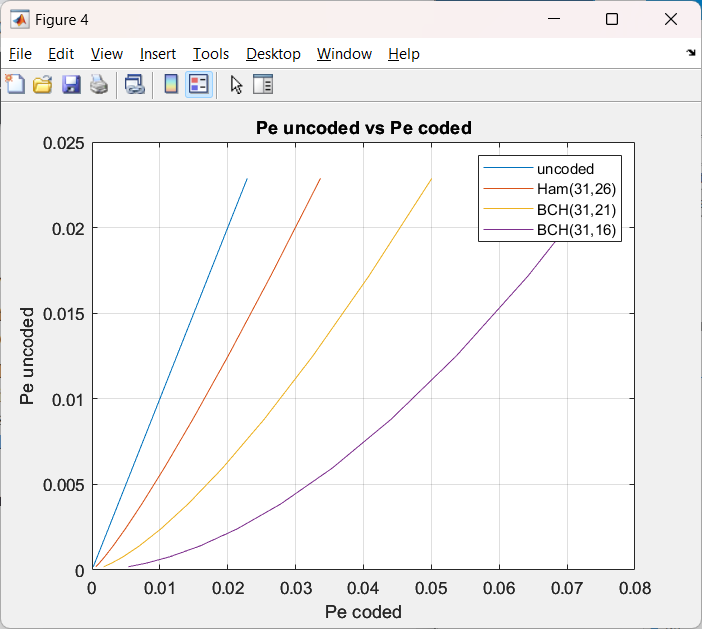


Figura 5- Pe sem codificação em função Pe codificada Figura 6 – BER sem codificado em função BER

Pela análise das *figuras 5 e 6* é possível observar que quando a probabilidade de erro do canal é maior no sinal sem codificação, os códigos que possuem uma menor taxa de bit têm um desempenho superior em relação aos restantes.

Ao compararmos a probabilidade de erro do código BCH com o de Hamming e o não codificado, observamos que até uma certa probabilidade de erro tem comportamento semelhantes, ou seja, para as probabilidades de erro próximas das anteriores, as suas probabilidades após descodificação apresentam valores semelhantes no caso de Hamming e ambos os códigos BCH apresentam a probabilidade de erro mais baixa. Isto deve se ao tipo de codificação do BCH (31,16) pois apresenta uma maior redundância, isto é, m=7 (n-k), o que leva a uma capacidade de correção até três erros nesta codificação em específico.

* **Probabilidade de erro teórico em função SNR para Hamming(31,26), BCH(31,21),BCH(31,16) e para r=1**

Para os códigos BCH a probabilidade de erro teórico muda dependendo do N e do K o que permite a correção de 2 erros para o BCH (31,21) e 3 erros para o BCH (31,16).

Partindo da expressão base:

Obtemos assim para o BCH (31,21):

E para o BCH (31,16):

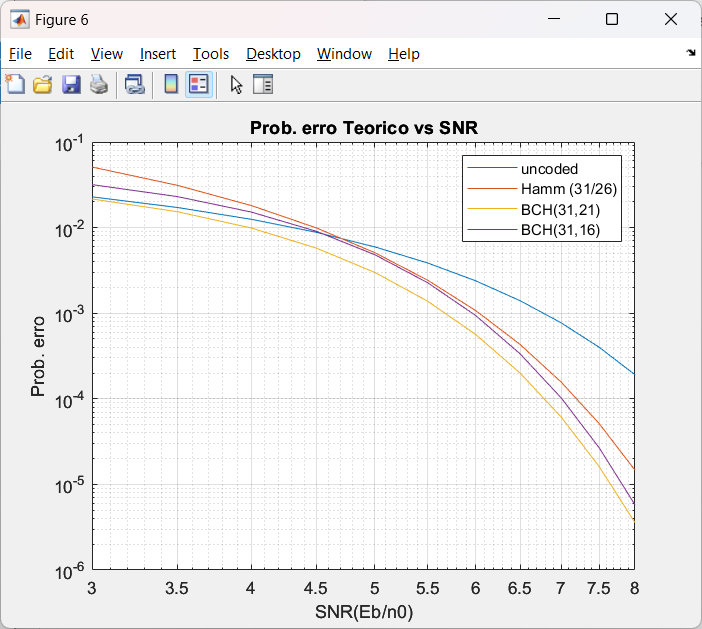


Figura 7 -Probabilidade de erro em função do SNR

Observando a *figura 7*, é possível concluir que á medida que o SNR aumenta, menor é a probabilidade de erro do canal e por consequência após a sua descodificação menor será essa probabilidade de erro, como observado anteriormente.

De maneira geral este tipo de codificação quando apresenta um maior comprimento, mas menor será a energia necessária para atingir probabilidades de erro mais baixa, sendo percetível para os códigos BCH.

Podemos ainda concluir que os códigos com maior comprimento são aqueles que tem uma menor probabilidade de erro para SNR mais alto.

* **Ganho para uma probabilidade de erro de 10-2**

Para o cálculo dos ganhos, observámos visualmente, a partir dos gráficos da probabilidade de erro em função do SNR, onde se encontrava o ponto em que a reta y=10-2 interceta as várias codificações.

Comparando esses valores com o valor da reta do não codificado obtemos os seguintes ganhos.

Uma imagem com mesa

Descrição gerada automaticamente

Figura 8 - ganho teórico e pratico para uma probabilidade de erro 10-2

* **Ganho para uma probabilidade de erro de 10-4**

Realizando o mesmo processo que anteriormente, mas agora para a probabilidade de erro de 10-4.

Uma imagem com mesa

Descrição gerada automaticamente

Figura 9 - ganho teórico e pratico para 10-2

Pela análise dos resultados nas *figuras 8 e 9*, concluímos que os valores têm algumas variações significantes em relação ao seu valor teórico, sendo possível que poderá ser pelo número de bits codificados ser inferior ao desejado ou mesmo erro humano, já que a tiragem dos valores foi realizada por observação a olho.

Observando ainda as mesmas figuras, podemos inferir que quanto menor for a probabilidade de erro maior é o ganho. É de salientar que quando obtivemos ganhos negativos significa que a taxa de erros aumentou em relação a um canal não codificado, logo a correção para o determinado canal de comunicação degradou.

* **Canal discreto com dependência estatística**

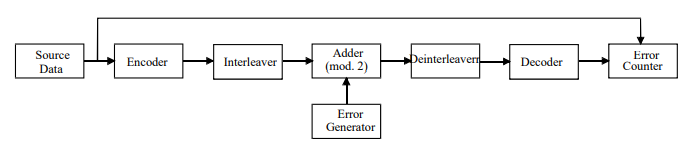


Figura 10 - simulação com dependência estática

Uma imagem com texto, relógio, antena

Descrição gerada automaticamente

Figura 11 - Dependência estatística

Nesta parte do trabalho vamos observar o desempenho de cada codificação para um canal de comunicação com dependência estatística como mostra a *figura 11*.

* **Probabilidade de erro introduzido no canal em função P (1/1) com valores (0,0.1,0.3,0.5) para os códigos Hamming (31,26), BCH (31,21) e BCH (31,16)**

Com P (1/1) = [0 , 0.1, 0.3, 0.5] , observamos o seu efeito para as diversas codificações.

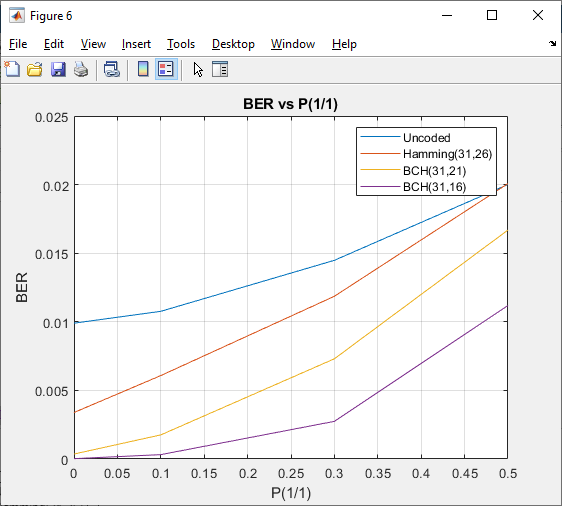


Figura 12 - BER em função de P (1/1)

Pela análise *da figura 12*, ao modificarmos o valor de P (1/1) para um valor superior verificamos que a eficácia dos códigos de hamming vai diminuindo, ou seja, a probabilidade de erro após a codificação é superior devido a que os erros consecutivos não são corrigidos.

Para a codificação BCH observamos o mesmo, à medida que aumentamos o P (1/1) a eficácia dos códigos diminui.

A diferença mais significativa entre ambos os códigos BCH independentemente do quanto aumentamos P (1/1), até ao máximo imposto (0.5), são sempre melhores que não termos qualquer codificação no canal. Concluímos assim que o uso dos códigos BCH acaba sempre por ser benéfico nesta situação.

* **Utilização das funções “matintrlv” e “matdeintrlv” (Matlab) para realizar o “interleaving” e o “deiterliving”**

Para otimizar a secção anterior vamos agora utilizar blocos que misturam os bits todos das palavras antes de serem adicionados os erros ao canal e, em seguida, quando “recebidos” voltam a ser colocados na posição certa, sendo estes blocos “interleaving” e o “deinterleaving”, respetivamente.

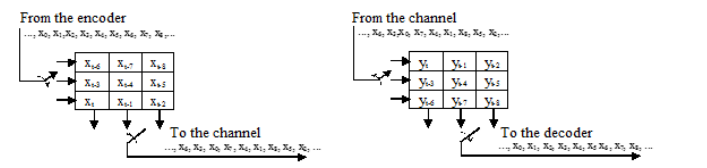
****

Figura 13 -Blocos interleaving e deinterleaving

O primeiro bloco da *figura 13* é o interleaving. Este bloco serve para misturar os bits recebidos do encoder e adicioná-los fazendo com que os erros sejam de novo independentes e não havendo uma grande degradação quando existe uma serie consecutiva de erros.

O seu tamanho é igual a n\*2 linhas e n\*2 colunas onde ‘n’ é o número de bits da palavra codificada.

* **Probabilidade de erro introduzido no canal em função de P (1/1) com valores (0,0.1,0.3,0.5) para os códigos Hamming(31,26), BCH(31,21) e BCH(31,16) usando os blocos interleaving e o deinterleaving e comparação entre o obtido anteriormente**

Utilizando agora os blocos interleaving e o deinterlveaving obtivemos a os seguintes resultados:

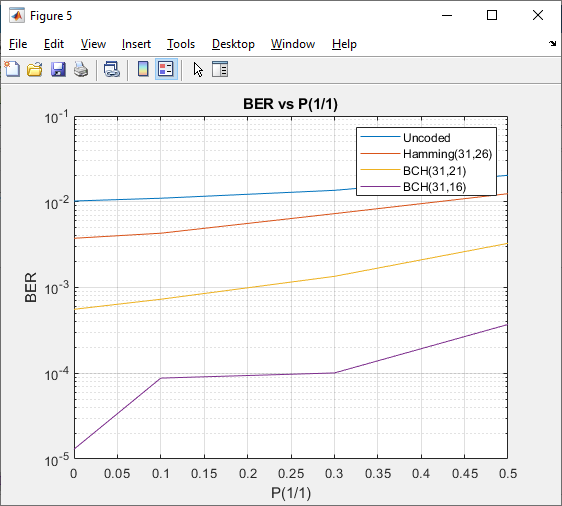


Figura 14 - BER em função de P (1/1) usando os blocos interleaving e deinterleaving

Pela análise da *figura 14*, podemos observar que o processo de interleaving/deinterleaving tem efeito de amortizar a subida da probabilidade do erro para P (1/1) alto relativamente ao obtido anteriormente, sendo que a eficácia voltou a ser superior do que no caso de não termos qualquer codificação uma vez que através do uso destes blocos o erro tornou a ser independente, ou seja, robusto a vários erros consecutivos.

Podemos observar ainda que os códigos BCH continuam a ser melhores que o código de Hamming, uma vez que a sua BER depois da descodificação se aproxima do não codificado e também pelo facto da BER dos BCH estar mais afastada.