



INSTITUTO FEDERAL
SANTA CATARINA
Campus São José

LISTA 2: TEORIA DE CÓDIGOS **TEORIA DA INFORMAÇÃO**

Disciplina: Sistemas de Comunicação II

Aluno: Guilherme Medeiros

São José, 31 de Maio de 2020

1. Considere um código de bloco linear (7,3) binário no qual os últimos bits da palavra-código correspondem aos bits de informação ou mensagem, m_0 , m_1 , m_2 . Os demais bits correspondem aos bits de paridade e são definidos da seguinte forma:

$$b_0 = m_0 + m_1 + m_2$$

$$b_1 = m_0 + m_1$$

$$b_2 = m_0 + m_2$$

$$b_3 = m_1 + m_2$$

- Construa a matriz geradora associada ao codificador em questão.
- O codificador é sistemático? Justifique.
- Encontre as palavras-código associada a todos os possíveis bits de entrada.
- Determine a capacidade de correção de erros do código, t .
- Prepare uma tabela de decodificação síndrome, padrão de erro.
- Decodifique as seguintes palavras recebidas: 1011001, 1101101, 1010101

Primeiramente, identificam-se $n = 7$, $K = 3$, $n-K = 4$, dados importantes para a realização do exercício.

Os bits m_0 , m_1 e m_2 são bits de informação. b_0 , b_1 , b_2 e b_3 são construídos com operações XOR entre os bits de entrada. Pode-se construir a tabela 1 para examinar os resultados dessas operações e visualizar as palavras código possíveis.

m_0	m_1	m_2	b_0	b_1	b_2	b_3	palavra código
0	0	0	0	0	0	0	0000000.
0	0	1	1	0	1	1	1011001.
0	1	0	1	1	0	1	1101010.
0	1	1	0	1	1	0	0110011.
1	0	0	1	1	1	0	1110100.
1	0	1	0	1	0	1	0101101.
1	1	0	0	0	1	1	0011110.
1	1	1	1	0	0	0	1000111.

Tabela 1: Construção das palavras código do sistema

Além disso, é necessário compreender o código P_{ij} de coeficientes dos bits de paridade b_i , que podem ser descritos pela equação 1.

$$p_{ij} = \{1, \text{ se } b_i \text{ depende de } m_j, 0 \text{ caso contrário}\} \quad (\text{equação 1: coeficientes}).$$

Desta forma, é possível criar os $n-k$ coeficientes p_i com base nas equações que definem b_0 , b_1 , b_2 e b_3 . Tais coeficientes podem ser vistos na tabela 2.

p_0	1110.
p_1	1101
p_2	1011

Tabela 2: coeficientes p_i

Este é o primeiro dado importante para a criação da matriz geradora (questão “a” da lista) G , definida pela equação 2,

$$G = [P \mid I_k] \quad (\text{equação 2: matriz geradora})$$

onde I_k é a matriz identidade $k \times k$

Com todos os dados obtidos e desenvolvidos até aqui, é possível criar a matriz geradora facilmente, simplesmente concatenando a matriz P com a matriz identidade 4×4 :

a.

1	1	1	0	⋮	1	0	0
1	1	0	1	⋮	0	1	0
1	0	1	1	⋮	0	0	1

Tabela 3: Matriz geradora

b. Nota-se que este código de bloco é um código sistemático, já que possui os bits m_0 , m_1 e m_2 inalterados na sua composição.

c. Para se obter-se todas as palavras códigos associadas a todas as diferentes informações de entrada deve-se aplicar a definição de b0, b1, b2 e b3 definidas no enunciado do exercício. Este desenvolvimento já foi realizado para a determinação da tabela 1. As palavras códigos encontradas lá também estão apresentadas abaixo, na tabela 4.

Informação	Palavra Código
000.	0000000.
001.	1011001.
010.	1101010.
011.	0110011.
100.	1110100.
101.	0101101.
110.	0011110.
111.	1000111.

Tabela 4: Palavras código possíveis

Para o entendimento total do problema, pode ser interessante se obter a matriz verificação de paridade H, definida por:

$$H = [I_{n-k} \mid P^T] \quad (\text{equação 3: matriz verificação de paridade})$$

A matriz verificação de paridade pode ser observada na tabela 5.

1	0	0	0	⋮	1	1	1
0	1	0	0	⋮	1	1	0
0	0	1	0	⋮	1	0	1
0	0	0	1	⋮	0	1	1

Tabela 5: Matriz verificação de paridade

d. Para calcular a capacidade de correção de erros t, primeiro é necessário calcular a distância de Hamming de cada palavra código. A distância de Hamming define-se como o número de localizações nas quais seus respectivos elementos diferem entre si, e pode ser simplificado para simplesmente o número de elementos não nulos de um conjunto de bits. Para o exemplo, a distância de Hamming de cada palavra código pode ser vista na tabela 6.

Palavra Código	Distância de. Hamming
0000000.	0
1011001.	4
1101010.	4
0110011.	4
1110100.	4
0101101.	4
0011110.	4
1000111.	4

Tabela 6: Distância de Hamming das palavras código

Como pode ser visto na tabela 6, a única e menor distância de Hamming diferente de zero é 4. Define-se a capacidade de correção de erros pela equação 4, que determina o número máximo de erros que um código consegue corrigir, com o valor resultante da equação sendo arredondado para baixo.

$$t = \left\lfloor \frac{1}{2}(d_{min} - 1) \right\rfloor \quad (\text{equação 4: capacidade de correção de erros})$$

Pela equação, fica simples calcular t para este caso, que resulta em 3/2, arredondando para baixo, $t = 1$.

e.

Para definir uma equação que caracteriza a síndrome, deve-se tomar a definição, que diz:

$$s = rH^T, \text{ sendo } r = c + e,$$

Onde c é a palavra código recebida e “e” um vetor de erro com mesmo tamanho de “c” e valor 1 no local onde o erro está presente. r é a sequência de bits recebida por um decodificador.

Entretanto:

$$s = rH^T = cH^T + eH^T = eH^T, \text{ sabendo que } cH^T \text{ é igual a zero.}$$

Logo, podemos definir a síndrome pela equação 5.

$$s = eH^T \quad (\text{equação 5: síndrome})$$

Sabemos que a capacidade de erro $t = 1$, por isso usaremos um vetor padrão de erro com apenas 1 ou nenhum valor não nulo.

Padrão de erro
0000000.
1000000.
0100000.
0010000.
0001000.
0000100.
0000010.
0000001.

Tabela 7: Padrão de erro

Possuindo os vetores padrão de erro e a matriz H^T , encontrar a síndrome é meramente fazer a multiplicação das duas matrizes. O processo foi realizado utilizando o software MATLAB (Mathworks), o resultado pode ser visto na tabela 8.

Síndrome	Padrão de erro
0000.	0000000.
1000.	1000000.
0100.	0100000.
0010.	0010000.
0001.	0001000.
1110.	0000100.
1101.	0000010.
1011.	0000001.

Tabela 8: Síndrome

f.

Possuímos tudo para realizar a recepção, decodificação, detecção e correção de erros. Para tal, possuindo os códigos recebidos $r_0 = \mathbf{1011001}$, $r_1 = \mathbf{1101101}$, $r_2 = \mathbf{1010101}$, e sabendo que defini-se a síndrome como $s_0 = r_0 H^T$, o código terá sido recebido sem erro se a síndrome extraída da equação anterior for 0000.

Isso só acontecerá obviamente quando um código recebido for igual a um dos códigos possíveis apresentados na tabela 1.

r_0 : 1011001

$s_0 = r_0 H^T = [1011001] * H^T = 0000 \rightarrow$ Código recebido sem erros.

r_1 : 1101101

Logo percebemos que esta palavra código não está presente na tabela 1, e detectamos um possível erro. Fazendo o cálculo:

$s_1 = r_1 H^T = [1101101] * H^T = 1000$. A síndrome é não nula, logo houve erro. Observando na tabela 8, detecta-se que o padrão de erro é 1000000.

Para corrigir o erro basta fazer:

$c_{1r} = r_1 + e = 0101101$, palavra código correta presente na tabela 1.

r_2 : 1010101

É percebido aqui o que ocorreu em r_1 , fazendo o cálculo:

$s_2 = r_2 H^T = [1010101] * H^T = 1111$.

Aqui temos como resultado uma síndrome que não está presente na tabela 8. É impossível corrigir o erro e provavelmente isso acontece porque mais do que um bit foi trocado na transmissão, sendo que a capacidade de correção de erro é de apenas 1 bit. Entretanto o sistema ainda é capaz de verificar que um erro aconteceu, já que a síndrome s_2 é diferente de 0000.