ELE32
Introdução a Comunicações
LAB 2 – Códigos Ciclicos

ITA
2º. Semestre de 2018
manish@ita.br

Representação de um vetor como um polinômio

- $\mathbf{v} = [v_0\ v_1\ v_2\ ...\ v_{n-2}\ v_{n-1}]$ equivale ao polinômio
  $v(D) = v_0+v_1D+v_2D^2+...+v_{n-2}D^{n-2}+v_{n-1}D^{n-1}$
- O valor de n é implícito pelo sistema e é a dimensão do vetor
- D é uma variável dummy
- Assim, o polinômio 1+D pode representar
  - [1 1 0 0 0] se n = 4
  - [1 1 0 0 0 0] se n = 5
- O grau do polinômio é o maior valor de i tal que $v_i \neq 0$

Códigos cícliclos

- Para códigos cíclicos, se **v** = [v_0 v_1 v_2 ... v_{n-2} v_{n-1}] é uma palavra código, então **v** = [v_{n-1} v_0 v_1 v_2 ... v_{n-2}] também é
- Não basta multiplicar o polinômio correspondente por D pois poderíamos ter um grau>n
- Seja $\mathbf{v}^{(i)}$ a rotação cíclica de **v** e $v^{(i)}(D)$ o polinômio correspondente Temos a seguinte igualdade:
  - $v(D)D^i = q(D)(1+D^n) + v^{(i)}(D)$
- Isto é: o resto da divisão de $v(D)D^i$ por $(1+D^n)$ é $v^{(i)}(D)$

Consequências: polinômio gerador

- Entre todas os vetores (palavras) código, existe um com grau mínimo: este é $g(D)$
- Propriedades:
  - $g_0 = 1$
  - $g(D)$ é único
  - Seu grau é por projeto $n-k$
  - Qualquer e toda combinação linear de $g(D), g(D)D, g(D)D^2, \ldots, g(D)D^{k-1}$ é uma palavra código
- O polinômio $g(D)$ é o <u>polinômio gerador</u> do código

Gerando palavras código

- Via matriz geradora:

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & g_2 & g_3 & \cdots & g_{n-k-2} & g_{n-k-1} & g_{n-k} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k-3} & g_{n-k-2} & g_{n-k-1} & g_{n-k} & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & \cdots & g_{n-k-4} & g_{n-k-3} & g_{n-k-2} & g_{n-k-1} & g_{n-k} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & g_{n-k} & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & g_{n-k-1} & g_{n-k} & 0 \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & g_{n-k-2} & g_{n-k-1} & g_{n-k} \end{bmatrix}$$

- Via produto de polinômios:
  - v(D) = g(D)u(D)
  - u(D) é a <u>informação</u> e tem grau máximo k-1 (comprimento k)

Como obter g(D)

- g(D) é sempre um fator de $(1+D^n)$, isto é, ele sempre divide $(1+D^n)$
- Logo, ele pode ser obtido via combinação dos fatores irredutíveis de $(1+D^n)$
- Por exemplo, para n = 7 temos:
  - $(1+D^7) = (1+D)(1+D+D^3)(1+D^2+D^3)$
  - Há 6 polinômios g(D) possíveis (quais?)
  - (o código de Hamming é um código cíclico)

Decodificação

- A divisão polinomial de uma palavra código pelo polinômio gerador gera a palavra de informação
- Em casos de erros de transmissão, receberíamos r(D) = v(D) + e(D), onde e(D) é o polinômio correspondente do padrão de erro
- A divisão de r(D) por g(D) pode ter resto diferente de zero que é a <u>síndrome </u>s(D)
- O grau da síndrome é sempre menor que o grau de g(D)

Correção via síndrome

- Se $s(D)$ é a síndrome de um vetor recebido $v'(D)$, rotacionar ciclicamente $v'(D)$ resultará na síndrome $s^{(1)}(D)$, onde $s^{(i)}(D)$ é o resultado da rotação cíclica da síndrome em relação a $g(D)$:
  - Quando rotacionamos $v(D)$ estamos fazendo-o em relação a $(1+D^n)$, isto é, quando o produto por D resulta em um termo D^n, substituímos este por 1
  - Quando rotacionamos $s(D)$ em relação a $g(D)$, substituiremos o termo D^{n-k} pelos outros termos de $g(D)$

Exemplo de rotação em relação a g(D)

- Neste exemplo $s(D) = 1+D^2 = [1\ 0\ 1]$ e $g(D) = 1+D+D^3=[1\ 1\ 0\ 1]$

$$
\begin{array}{llll}
101 \Rightarrow & 010'1 \rightarrow & 010+110 & =100 \\
100 \Rightarrow & 010'0 \rightarrow & & =010 \\
010 \Rightarrow & 001'0 \rightarrow & & =001 \\
001 \Rightarrow & 000'1 \rightarrow & 000+110 & =110 \\
110 \Rightarrow & 011'0 \rightarrow & & =011 \\
011 \Rightarrow & 001'1 \rightarrow & 001+110 & =111 \\
111 \Rightarrow & 011'1 \rightarrow & 011+110 & =101
\end{array}
$$

Correção via síndrome

- Não é necessário armazenar todas as síndromes: basta armazenar aquelas em que há erro na primeira posição, por exemplo.
- Também não é necessário armazenar o padrão de erro associado a cada síndrome: basta corrigir o erro na primeira posicão e continuar o algoritmo

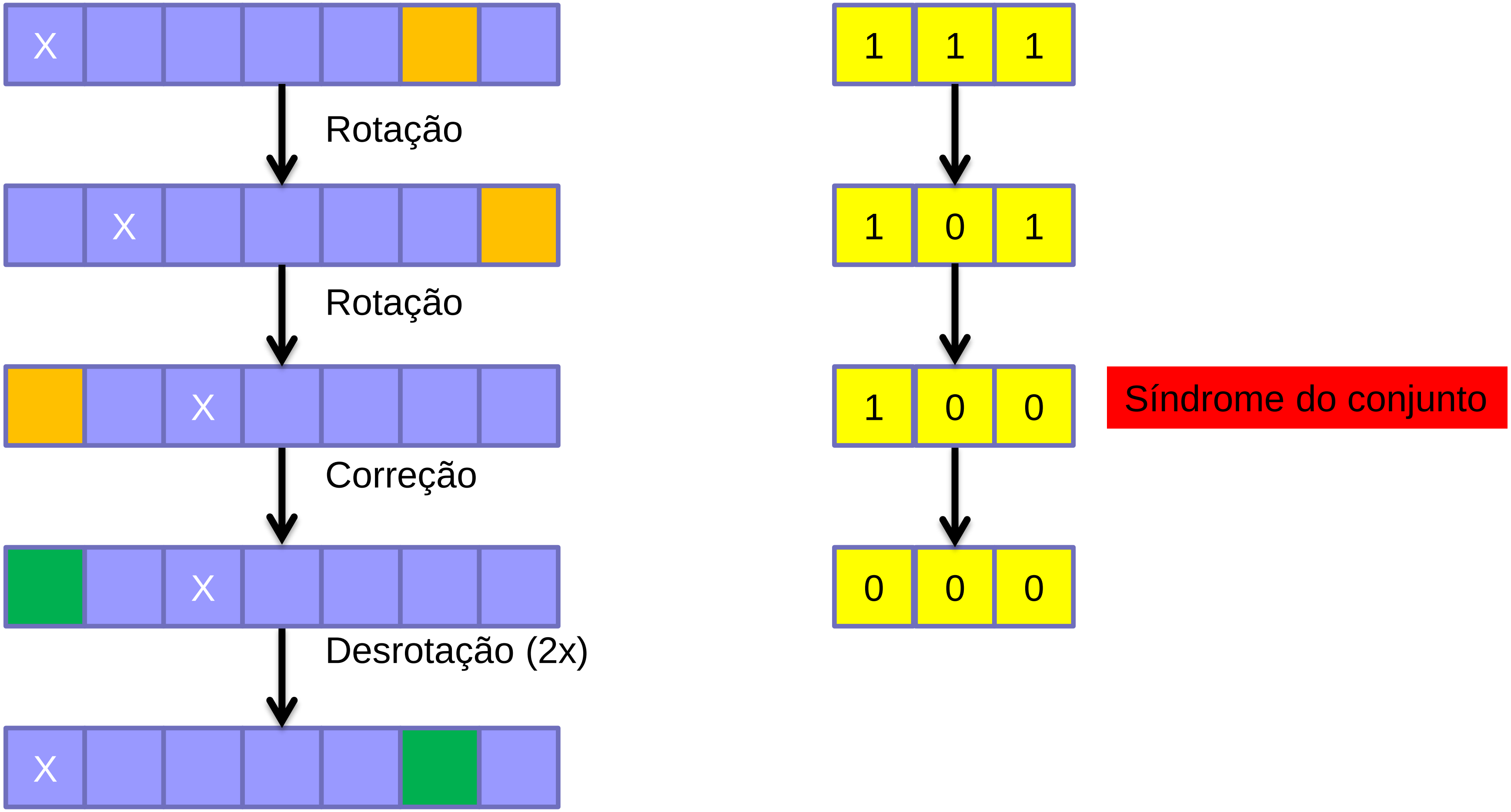
Algoritmo de decodificação

- Esta propriedade anterior permite armazenar somente as síndromes onde há erro no primeiro bit da palavra código
- Algoritmo:

1. Identifique a síndrome. Se síndrome igual a zero, pule para o penúltimo passo.
2. Síndrome dentro do conjunto de síndromes associados a erros na última posição?
   1. Não - pule para o passo 3
   2. Sim - Troque o valor do último bit e retorne ao passo 1
3. Gire ciclicamente a síndrome e a palavra, apropriadamente. Retorne ao passo 1.
4. Gire ciclicamente a palavra código até que a posição seja igual à original
5. Obtenha a palavra de informação dividindo a palavra código por g(D)

Exemplo

Conjunto de síndromes
onde há erro na primeira posição:
s(D) = 1



Atividades - projeto

1. Gere códigos com taxa semelhante (+-5%) à taxa do código de Hamming do laboratório anterior para pelo menos 5 valores distintos de n entre 8 e 16. Para isso utilize o comando cyclpoly(n,k,'all') do MATLAB, que gera a representação vetorial de todos os polinômios geradores para códigos cíclicos com tamanho n e taxa k/n, se existirem. Esta é a única função específica do MATLAB que você pode usar neste laboratório.
2. Entre todas as alternativas possíveis para o mesmo valor de k e n, pode haver uma melhor do que as outras. Calcule a distância mínima dos códigos gerados no item anterior e escolha, para o mesmo n e k, aquele com a maior distância mínima

Atividades - implementação

1. Implemente o codificador para os códigos escolhidos no item anterior.
2. Sabendo a distância mínima, implemente o decodificador para os códigos escolhidos. O seu decodificador só pode armazenar as síndromes associadas a erros na primeira posição da palavra código. Além disso, o seu codificador não pode armazenar o padrão de erro associado a cada síndrome.
3. Calcule a probabilidade de erro para todos os seus códigos de forma semelhante a como foi feito para o código de Hamming do laboratório anterior. Compare os resultados.
4. [Opcional e difícil] Implemente um codificador e decodificador eficientes para códigos BCH, que são um tipo de códigos cíclicos.