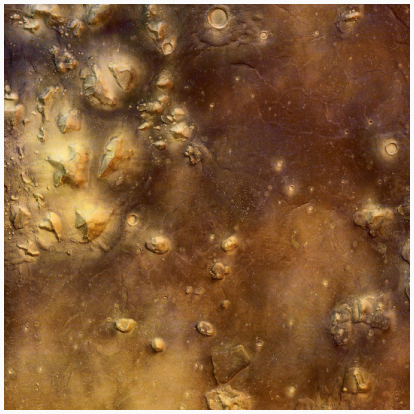
Teoria da Informação e Codificação

Códigos de bloco



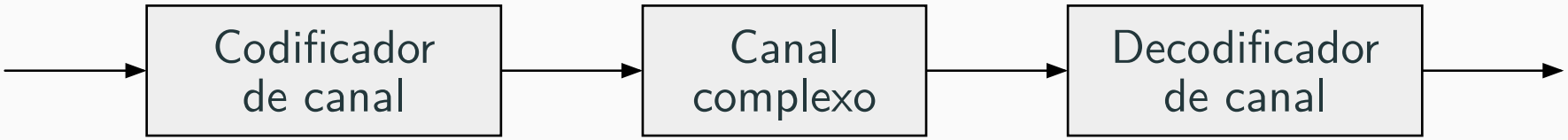
Prof. Roberto Wanderley da Nóbrega

Instituto Federal de Santa Catarina

Introdução

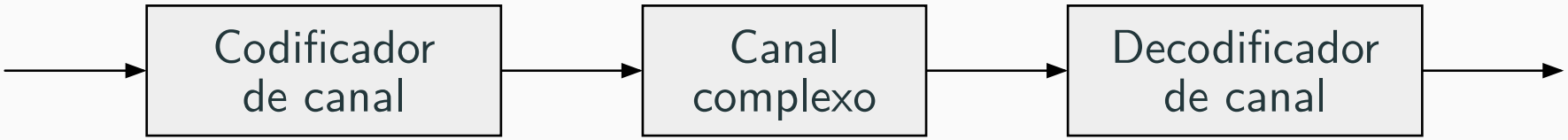
O problema de codificação de canal

Modelo típico de telecomunicações:



O problema de codificação de canal

Modelo típico de telecomunicações:



- **Informação:** sequência de bits independentes e equiprováveis.
- **Canal:** entrada e saída são sequências de números complexos.
- **Codificador de canal:** bits $\mapsto$ sequência complexa (entrada do canal).
- **Decodificador de canal:** sequência complexa (saída do canal) $\mapsto$ estimativa dos bits.

Codificação de canal = controle de erros + modulação

O estudo da codificação de canal costuma ser separado em duas partes:

- **Controle de erros:** bits $\mapsto$ bits.
  Insere redundância na informação.
- **Modulação:** bits $\mapsto$ símbolos complexos.
  Como visto em comunicações digitais.

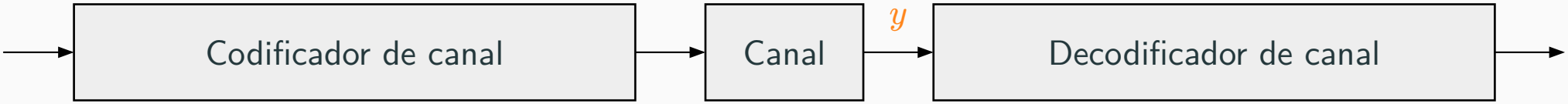
Codificação de canal = controle de erros + modulação

O estudo da codificação de canal costuma ser separado em duas partes:

- **Controle de erros:** bits $\mapsto$ bits.
  Insere redundância na informação.
- **Modulação:** bits $\mapsto$ símbolos complexos.
  Como visto em comunicações digitais.

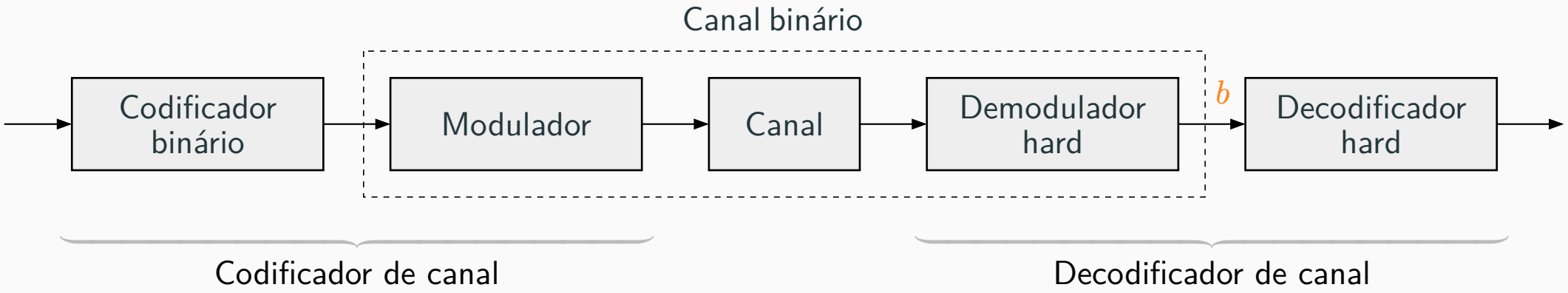
Observação: Essa nomenclatura, em que o termo "codificação de canal" engloba tanto a modulação quanto o controle de erros, é baseada em *Gallager: Principles of Digital Communication (2008)*. Outros autores utilizam o termo "codificação de canal" para se referir exclusivamente ao controle de erros.

Abordagem 1: Codificação e modulação conjuntas



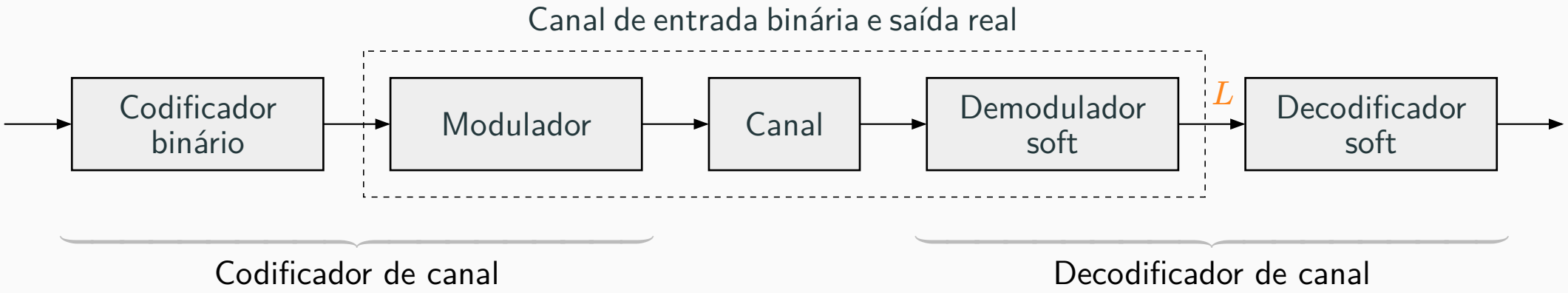
- O decodificador recebe os *números complexos* y que saem do canal.
  Toda a informação disponível.
- Compara y diretamente com as sequências de símbolos do código.
  No canal AWGN: por *distância euclidiana*.
- Código e modulação são projetados e decodificados *juntos*.
  Exemplos: Multilevel coding (Imai & Hirakawa, 1977) e trellis-coded modulation (Ungerboeck, 1982).

Abordagem 2: Separação com decisão hard



- O demodulador *decide* cada bit: o decodificador recebe só *bits* b, alguns *invertidos*.
  Ao decidir, o demodulador descarta a "confiança" de cada decisão. Costuma custar de 2 a 3 dB.
- Compara b com as sequências de bits do código.
  Por *distância de Hamming*: toda discordância pesa o mesmo.
- Modulador–canal–demodulador $\equiv$ *canal binário* (entrada e saída são bits).
  Código e modulação ficam *independentes*. É a abordagem clássica.

Abordagem 3: Separação com decisão soft

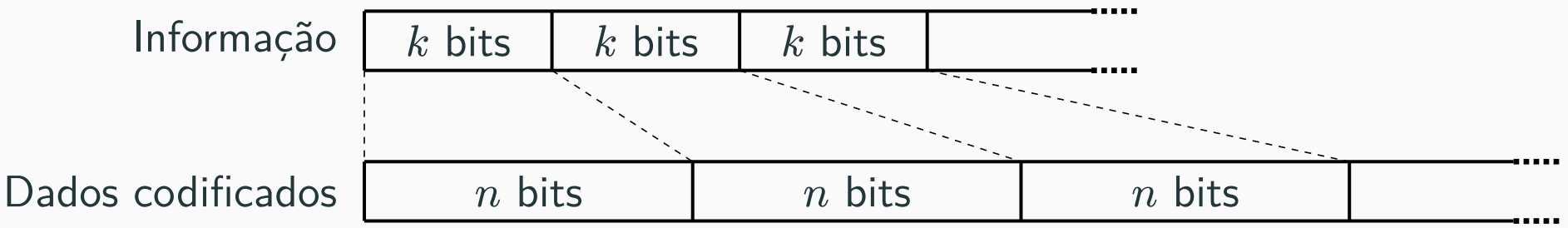


- O demodulador *não decide*: entrega, para cada bit, um *número real* L.
  Mais tarde definiremos o "valor L": o sinal é a decisão e a magnitude é a "confiança".
- Compara L com as sequências de bits do código.
  Por distância de Hamming *ponderada*: cada discordância pesa conforme a confiança do bit.
- Modulador–canal–demodulador $\equiv$ *canal de entrada binária e saída real*.
  Código e modulação *continuam independentes*. É a arquitetura usada no 5G e no Wi-Fi.

Códigos de bloco

Ideia básica

Na **codificação de bloco**, blocos de k bits são mapeados em blocos de n bits.



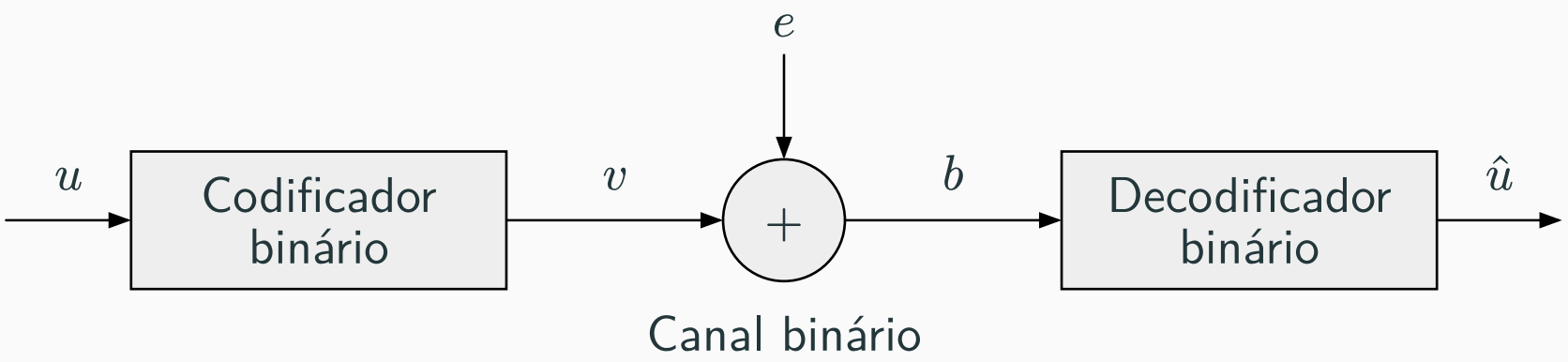
- Cada bloco de k bits na entrada do codificador é uma **mensagem**, denotada por u.
- Cada bloco de n bits na saída do codificador é uma **palavra-código**, denotada por v.

Utiliza-se a nomenclatura "código (n, k)", em que k é a **dimensão** e n é o **comprimento**.
Define-se também a **redundância** como $m = n - k$ e a **taxa** como $R = k/n$.

Observação: A rigor, "dimensão" só faz sentido para códigos lineares, mas cometeremos esse abuso de nomenclatura.

Canal binário

Seguindo a abordagem 2 (separação com decisão hard):



- O canal binário pode ser modelado com uma *soma módulo 2* (XOR) elemento a elemento entre a palavra-código v e um **padrão de erro** e, resultando na **palavra recebida** b.
- O padrão de erro é uma sequência de n bits contendo `1` nas posições em que ocorreu inversão de bit e `0` nas demais posições.

Resumo da notação

Símbolo	Descrição
n	comprimento
k	dimensão
m	redundância
R	taxa

Símbolo	Descrição	Tamanho
u	mensagem	k bits
v	palavra-código	n bits
e	padrão de erro	n bits
b	palavra recebida	n bits
$\hat{u}$	estimativa da mensagem	k bits

Observação: Por convenção, os bits de cada sequência serão indexados a partir de zero. Assim, por exemplo, a mensagem é $u = (u_0, u_1, \cdots, u_{k-1})$ e a palavra-código é $v = (v_0, v_1, \cdots, v_{n-1})$.

Exemplo: Código de repetição $(3, 1)$

- **Codificador:** repete o bit da mensagem três vezes.
- **Decodificador:** maioria de votos.

Codificador

u	v
0	000
1	111

Decodificador

b	$\hat{u}$
000	0
100	0
010	0
110	1
001	0
101	1
011	1
111	1

Exemplo: Código de repetição $(3, 1)$

- **Codificador:** repete o bit da mensagem três vezes.
- **Decodificador:** maioria de votos.
- *Corrige* qualquer inversão de bit única.

Codificador

u	v
0	000
1	111

Decodificador

b	$\hat{u}$
000	0
100	0
010	0
110	1
001	0
101	1
011	1
111	1

Exemplo: Código de repetição $(3, 1)$

- **Codificador:** repete o bit da mensagem três vezes.
- **Decodificador:** maioria de votos.
- *Corrige* qualquer inversão de bit única.
- Taxa: $R = 1/3$.

Codificador

u	v
0	000
1	111

Decodificador

b	$\hat{u}$
000	0
100	0
010	0
110	1
001	0
101	1
011	1
111	1

Exemplo: Código de paridade simples $(3, 2)$

- **Codificador:** acrescenta um bit de paridade no final. (número par de bits 1.)
- **Decodificador:** verifica a paridade.
  - Par: $\hat{u} =$ dois primeiros bits.
  - Ímpar: declara falha ($\perp$) (alternativa: chutar).

Codificador

u	v
00	000
10	101
01	011
11	110

Decodificador

b	$\hat{u}$
000	00
100	$\perp$
010	$\perp$
110	11
001	$\perp$
101	10
011	01
111	$\perp$

Exemplo: Código de paridade simples $(3, 2)$

- **Codificador:** acrescenta um bit de paridade no final. (número par de bits 1.)
- **Decodificador:** verifica a paridade.
  - Par: $\hat{u} =$ dois primeiros bits.
  - Ímpar: declara falha ($\perp$) (alternativa: chutar).
- *Detecta* (mas não corrige) uma única inversão de bit.

Codificador

u	v
00	000
10	101
01	011
11	110

Decodificador

b	$\hat{u}$
000	00
100	$\perp$
010	$\perp$
110	11
001	$\perp$
101	10
011	01
111	$\perp$

Exemplo: Código de paridade simples $(3, 2)$

- **Codificador:** acrescenta um bit de paridade no final. (número par de bits 1.)
- **Decodificador:** verifica a paridade.
  - Par: $\hat{u} =$ dois primeiros bits.
  - Ímpar: declara falha ($\perp$) (alternativa: chutar).
- *Detecta* (mas não corrige) uma única inversão de bit.
- Taxa: $R = 2/3$.

Codificador

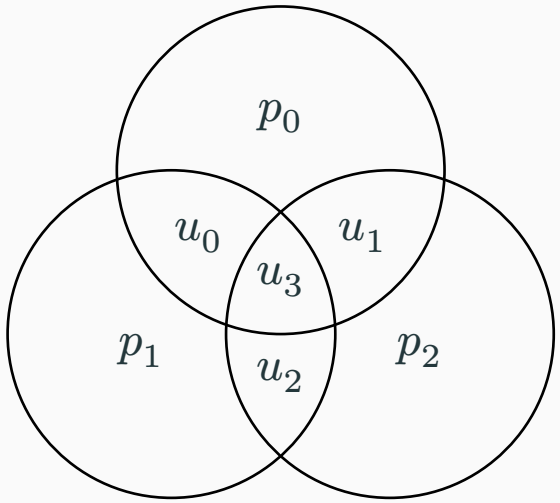
u	v
00	000
10	101
01	011
11	110

Decodificador

b	$\hat{u}$
000	00
100	$\perp$
010	$\perp$
110	11
001	$\perp$
101	10
011	01
111	$\perp$

Exemplo: Código de Hamming $(7, 4)$

- Proposto por Richard W. Hamming em 1950.
- Mensagem: $u = (u_0, u_1, u_2, u_3)$.
- Palavra-código: $v = (u_0, u_1, u_2, u_3, p_0, p_1, p_2)$.
- Bits de paridade: paridade **par** em cada círculo.





Richard W. Hamming (1915–1998).
Fonte: UChicago, PSD Trailblazers.
Foto: cortesia da Nokia Bell Labs.

Exemplo: Código de Hamming $(7, 4)$

Tabela de codificação ($2^4 = 16$ entradas):

u	v	u	v
0000	0000000	0001	0001111
1000	1000110	1001	1001001
0100	0100101	0101	0101010
1100	1100011	1101	1101100
0010	0010011	0011	0011100
1010	1010101	1011	1011010
0110	0110110	0111	0111001
1110	1110000	1111	1111111

Exemplo: Código de Hamming $(7, 4)$

Decodificação:

1. Calcule a paridade de cada círculo.
2. Se todas forem pares, declare que não houve erro.
3. Senão, identifique o bit invertido e corrija.

- *Corrige* qualquer inversão de bit única.
- Taxa: $R = 4/7$.

Observação: Uma tabela de decodificação teria $2^7 = 128$ entradas.

Exercício 1

Considere o código de Hamming $(7, 4)$. Suponha que a mensagem seja $u = \texttt{1011}$.

(a) Determine a palavra-código v.

(b) Suponha que o canal inverta o bit da posição 1 (lembre que o bit mais à esquerda é o de índice 0). Determine o padrão de erro e, a palavra recebida b e a mensagem decodificada $\hat{u}$.

(c) Repita o item anterior, supondo agora que o canal, além de inverter o bit da posição 1, inverta também o bit da posição 5.

Exercício 1

Considere o código de Hamming $(7, 4)$. Suponha que a mensagem seja $u = \texttt{1011}$.

(a) Determine a palavra-código v.

(b) Suponha que o canal inverta o bit da posição 1 (lembre que o bit mais à esquerda é o de índice 0). Determine o padrão de erro e, a palavra recebida b e a mensagem decodificada $\hat{u}$.

(c) Repita o item anterior, supondo agora que o canal, além de inverter o bit da posição 1, inverta também o bit da posição 5.

Respostas:

(a) $v = \texttt{1011010}$.

(b) $e = \texttt{0100000}$, $b = \texttt{1111010}$, $\hat{u} = \texttt{1011}$.

(c) $e = \texttt{0100010}$, $b = \texttt{1111000}$, $\hat{u} = \texttt{1110}$.

Codificadores sistemáticos

Um codificador de um código de bloco é dito ser **sistemático** se os bits da mensagem aparecem inalterados na palavra-código, em posições pré-determinadas.

Codificadores sistemáticos

Um codificador de um código de bloco é dito ser **sistemático** se os bits da mensagem aparecem inalterados na palavra-código, em posições pré-determinadas.

Vantagem: Recuperar a mensagem a partir da palavra-código é trivial: basta descartar os bits de paridade.

Exercício 2

Codificador 1

u	v
00	00000
10	10010
01	00001
11	10011

Codificador 2

u	v
00	00000
10	10011
01	10010
11	00001

(a) Determine o conjunto de palavras-código de cada codificador.

(b) Quais dos codificadores são sistemáticos?

(c) Os três exemplos vistos anteriormente — código de repetição $(3, 1)$, código de paridade simples $(3, 2)$ e código de Hamming $(7, 4)$ — são sistemáticos?

Exercício 2

Codificador 1

u	v
00	00000
10	10010
01	00001
11	10011

Codificador 2

u	v
00	00000
10	10011
01	10010
11	00001

(a) Determine o conjunto de palavras-código de cada codificador.

(b) Quais dos codificadores são sistemáticos?

(c) Os três exemplos vistos anteriormente — código de repetição $(3, 1)$, código de paridade simples $(3, 2)$ e código de Hamming $(7, 4)$ — são sistemáticos?

Respostas:

(a) Ambos: $\mathcal{C} = \{\texttt{00000}, \texttt{00001}, \texttt{10010}, \texttt{10011}\}$.

(b) Apenas o codificador 1 (por exemplo, nas duas últimas posições).

(c) Sim (todos nas k primeiras posições).

Códigos de bloco lineares

Definição

Um código de bloco é dito ser **linear** se a soma de quaisquer duas palavras-código é também uma palavra-código.

Definição

Um código de bloco é dito ser **linear** se a soma de quaisquer duas palavras-código é também uma palavra-código.

Observações:

- Em outras palavras, o conjunto de palavras-código é um subespaço k-dimensional do espaço vetorial $\{0,1\}^n$. Agora o nome *dimensão* para k se justifica.
- Duas consequências: a palavra nula é sempre palavra-código de um código linear; o número de palavras-código de um código linear é sempre uma potência de 2.

Exercício 3

Determine se os códigos abaixo são lineares. Justifique.

(a) O código cujo conjunto de palavras-código é $\{\texttt{10010}, \texttt{00001}, \texttt{10011}\}$.

(b) O código cujo conjunto de palavras-código é $\{\texttt{000}, \texttt{100}, \texttt{010}, \texttt{001}\}$.

(c) Os códigos de repetição $(3, 1)$ e paridade simples $(3, 2)$.

(d) O código de Hamming $(7, 4)$.

Exercício 3

Determine se os códigos abaixo são lineares. Justifique.

(a) O código cujo conjunto de palavras-código é $\{\texttt{10010}, \texttt{00001}, \texttt{10011}\}$.

(b) O código cujo conjunto de palavras-código é $\{\texttt{000}, \texttt{100}, \texttt{010}, \texttt{001}\}$.

(c) Os códigos de repetição $(3, 1)$ e paridade simples $(3, 2)$.

(d) O código de Hamming $(7, 4)$.

Respostas:

(a) Não.

(b) Não.

(c) Sim, ambos: por inspeção das palavras-código.

(d) Sim: se v e v' têm paridade par em cada círculo, então $v + v'$ também terá.

Matriz geradora

Uma matriz binária G de tamanho $k \times n$ é dita ser uma **matriz geradora** de um código se

$$v \text{ é palavra-código} \iff v = uG.$$

Observações:

- Dado o conjunto de palavras-código, a matriz geradora não é, em geral, única.
- A matriz G define um codificador específico: $v = uG$.
- Para codificadores sistemáticos, G contém a identidade I_k como submatriz.

Exercício 4

Considere o código de bloco linear com matriz geradora

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

e codificador $v = uG$. Construa a tabela de codificação, isto é, determine a palavra-código correspondente a cada uma das 2^k mensagens.

Exercício 4

Considere o código de bloco linear com matriz geradora

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

e codificador $v = uG$. Construa a tabela de codificação, isto é, determine a palavra-código correspondente a cada uma das 2^k mensagens.

Resposta:

u	000	100	010	110	001	101	011	111
v	000000	100011	010101	110110	001110	101101	011011	111000

Exercício 5

Determine uma matriz geradora para cada código abaixo.

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Exercício 5

Determine uma matriz geradora para cada código abaixo.

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Respostas:

(a) $G = [1 \quad 1 \quad 1]$.

(b) $G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$.

(c) $G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$.

Matriz verificadora

Uma matriz binária H de tamanho $m \times n$ é dita ser uma **matriz verificadora** de um código se

$$v \text{ é palavra-código} \iff Hv^{\mathsf{T}} = 0.$$

Observações:

- Lembrando que $m = n - k$ é a redundância do código.
- Na equação acima, 0 é o vetor coluna nulo de tamanho m.

Relação entre as matrizes geradora e verificadora

As matrizes G e H satisfazem

$$GH^{\mathsf{T}} = 0.$$

Em geral, uma pode ser obtida a partir da outra por eliminação gaussiana. No entanto, no caso sistemático, vale a proposição abaixo.

Fato: Se $G = [I_k \;\; P]$, então $H = [P^{\mathsf{T}} \;\; I_m]$, onde I_a é a matriz identidade $a \times a$ e P, de tamanho $k \times m$, é chamada de **submatriz de paridade**.

Relação entre as matrizes geradora e verificadora

As matrizes G e H satisfazem

$$GH^{\mathsf{T}} = 0.$$

Em geral, uma pode ser obtida a partir da outra por eliminação gaussiana. No entanto, no caso sistemático, vale a proposição abaixo.

Fato: Se $G = [I_k \;\; P]$, então $H = [P^{\mathsf{T}} \;\; I_m]$, onde I_a é a matriz identidade $a \times a$ e P, de tamanho $k \times m$, é chamada de **submatriz de paridade**.

Demonstração:

$$GH^{\mathsf{T}} = [I_k \;\; P] \begin{bmatrix} P \\ I_m \end{bmatrix} = I_k P + P I_m = P + P = 0.$$

Exercício 6

Determine uma matriz verificadora para cada código abaixo. Interprete a equação $Hv^{\mathsf{T}} = 0$.

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Exercício 6

Determine uma matriz verificadora para cada código abaixo. Interprete a equação $Hv^{\mathsf{T}} = 0$.

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Respostas:

(a) $H = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$.

$v_0 = v_1 = v_2$.

(b) $H = [1 \quad 1 \quad 1]$.

$v_0 + v_1 + v_2 = 0$.

(c) $H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$.

Paridade par em cada círculo.

Exercício 7

Considere o código de bloco linear com matriz verificadora

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Determine uma matriz geradora, as palavras-código e a taxa do código.

Exercício 7

Considere o código de bloco linear com matriz verificadora

$$H = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Determine uma matriz geradora, as palavras-código e a taxa do código.

Respostas:

$$G = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}, \qquad \mathcal{C} = \{\texttt{00000}, \texttt{01101}, \texttt{10110}, \texttt{11011}\} \qquad \text{e} \qquad R = 2/5.$$

Distância mínima

Distância e peso de Hamming

Definições:

- **Distância de Hamming** $\mathrm{d_H}(x, y)$: número de posições em que x e y diferem.
- **Peso de Hamming** $\mathrm{w_H}(x)$: número de bits 1 em x.

Fatos: $\mathrm{d_H}(x, y) = \mathrm{w_H}(x + y)$ e $\mathrm{d_H}(0, x) = \mathrm{w_H}(x)$.

Distância mínima

A **distância mínima** de um código de bloco, denotada por d, é a menor distância de Hamming entre duas palavras-código distintas.

Distância mínima

A **distância mínima** de um código de bloco, denotada por d, é a menor distância de Hamming entre duas palavras-código distintas.

Fato: Para códigos lineares, d é o menor peso de Hamming das palavras-código não nulas.

Distribuição de peso das palavras-código

A **distribuição de peso das palavras-código** é a sequência $A_0, A_1, ..., A_n$, onde A_j é o número de palavras-código de peso j.

Distribuição de peso das palavras-código

A **distribuição de peso das palavras-código** é a sequência $A_0, A_1, ..., A_n$, onde A_j é o número de palavras-código de peso j.

Propriedades:

- $A_0 + A_1 + \cdots + A_n = 2^k$.
- $A_0 = 1$.
- $A_j = 0$ para $1 \leq j \leq d - 1$.
- $A_d > 0$.

Exercício 8

Determine a distância mínima e a distribuição de peso das palavras-código do:

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Exercício 8

Determine a distância mínima e a distribuição de peso das palavras-código do:

(a) Código de repetição $(3, 1)$.

(b) Código de paridade simples $(3, 2)$.

(c) Código de Hamming $(7, 4)$.

Respostas:

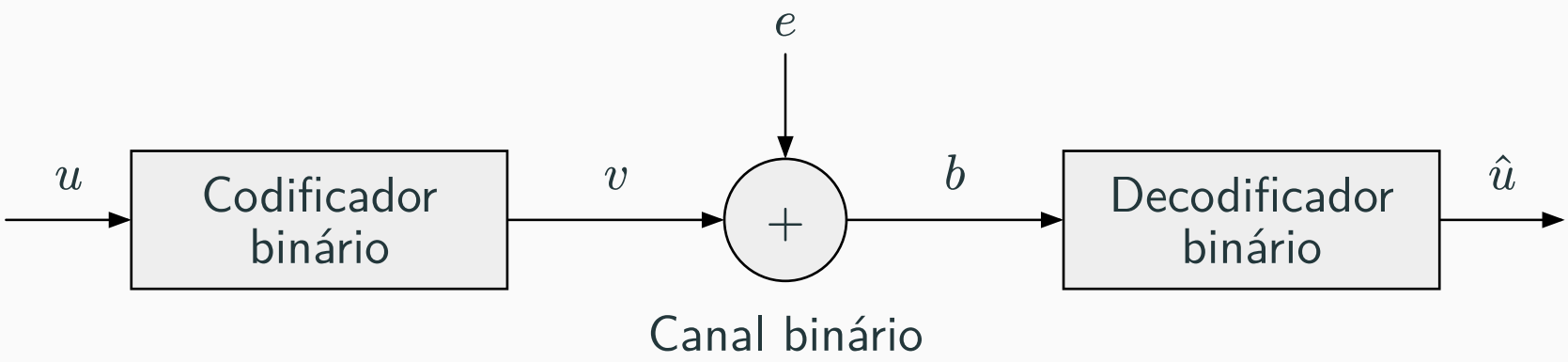
(a) $d = 3$ e $(A_0, ..., A_3) = (1, 0, 0, 1)$.

(b) $d = 2$ e $(A_0, ..., A_3) = (1, 0, 3, 0)$.

(c) $d = 3$ e $(A_0, ..., A_7) = (1, 0, 0, 7, 7, 0, 0, 1)$.

Decodificação de mínima distância

O **decodificador de mínima distância** seleciona a palavra-código mais próxima da palavra recebida, de acordo com a distância de Hamming.



Exercício 9

Considere um código cujas palavras-código são

$$\mathcal{C} = \{\texttt{00000}, \texttt{01110}, \texttt{10011}, \texttt{11101}\}.$$

Suponha que foi transmitida a palavra-código $v = \texttt{10011}$. Efetue a decodificação de mínima distância, supondo:

(a) $e = \texttt{00000}$, ou seja, $b = \texttt{10011}$.

(b) $e = \texttt{00010}$, ou seja, $b = \texttt{10001}$.

(c) $e = \texttt{10010}$, ou seja, $b = \texttt{00001}$.

(d) $e = \texttt{10100}$, ou seja, $b = \texttt{00111}$.

Exercício 9: Respostas

(a) $\hat{v} = \texttt{10011}$ (correto).

(b) $\hat{v} = \texttt{10011}$ (correto).

(c) $\hat{v} = \texttt{00000}$ (erro de decodificação).

(d) Empate: `01110` e `10011` estão ambas a distância 2 de b. O resultado depende da regra de desempate.

Capacidade de correção de erros

Teorema. Considere um código de bloco (linear ou não) com distância mínima d e seja

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor,$$

onde $\lfloor x \rfloor$ denota o maior inteiro menor ou igual a x. Com decodificação de mínima distância:

- Todos os padrões de erro de peso t ou menos pode ser corrigidos.
- Existe pelo menos um padrão de erro de peso $t+1$ que não pode ser corrigido.

O parâmetro t é chamado de **capacidade de correção de erros** do código.

Observação: Apesar do resultado acima, um código, em geral, é capaz de corrigir padrões de erro de peso $> t$. Um código linear, por exemplo, pode corrigir 2^{n-k} padrões de erro (como será visto mais adiante).

Demonstração

Antes de iniciar a demonstração, note que a expressão para o valor de t é equivalente a

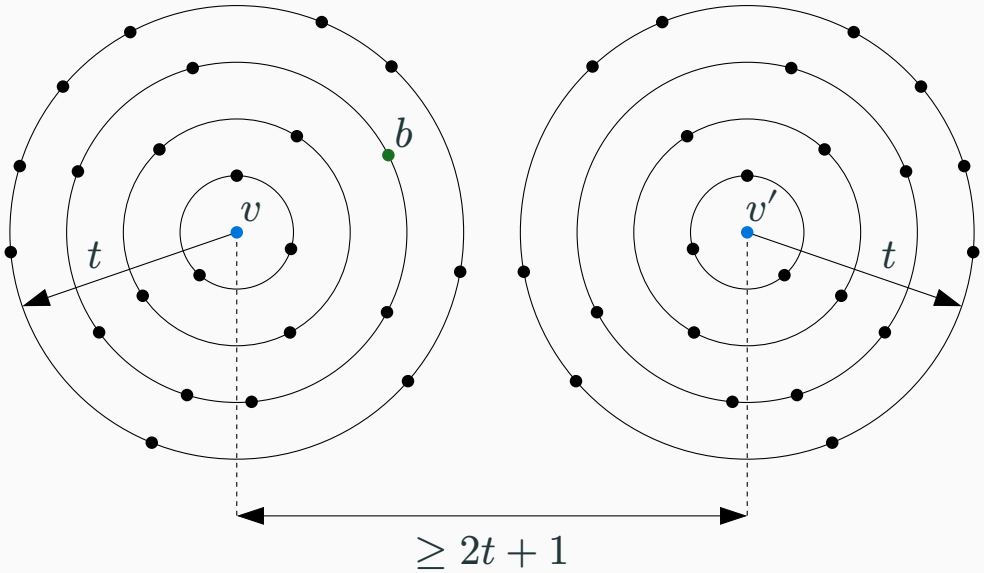
$$d = \begin{cases} 2t+1, & \text{se } d \text{ é ímpar}, \\ 2t+2, & \text{se } d \text{ é par}. \end{cases}$$

Portanto, $\mathrm{d_H}(v, v') \geq 2t+1$ para quaisquer duas palavras-código distintas v e v'.

Parte 1. Assuma primeiramente que o padrão de erro e tenha peso t ou menos. Suponha que v seja a palavra-código transmitida e $b = v + e$ seja a palavra recebida. Temos que

$$\mathrm{d_H}(v, b) = \mathrm{w_H}(e) \leq t.$$

Considere agora uma palavra-código $v' \neq v$.



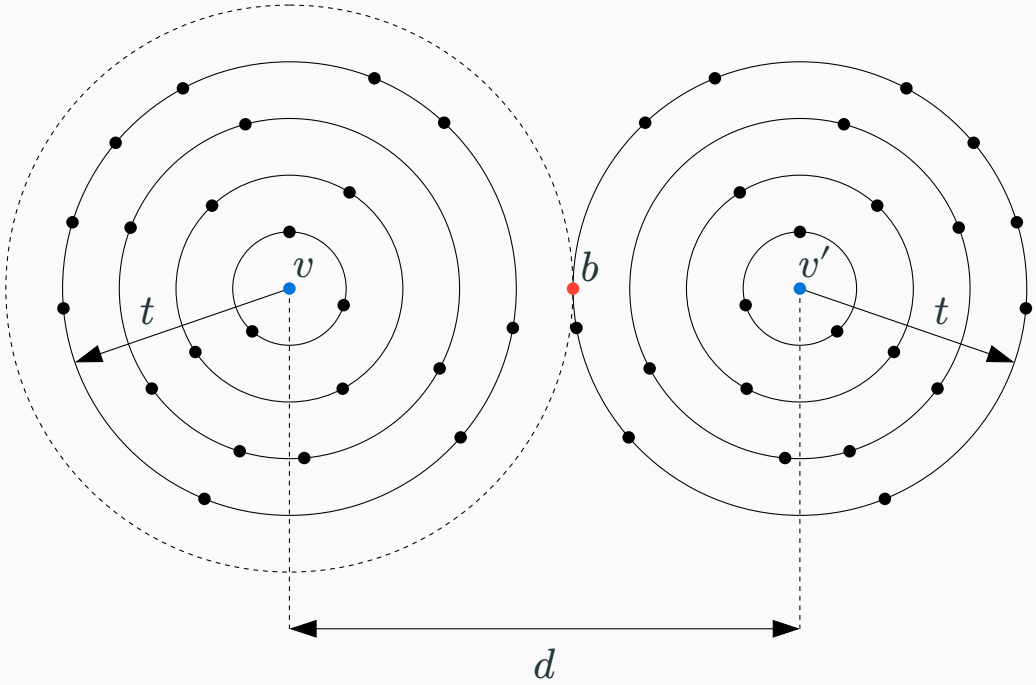
Demonstração

Pela desigualdade triangular, temos que $\mathrm{d_H}(v, v') \leq \mathrm{d_H}(v, b) + \mathrm{d_H}(v', b)$, de modo que

$$\mathrm{d_H}(v', b) \geq \underbrace{\mathrm{d_H}(v, v')}_{\geq 2t+1} - \underbrace{\mathrm{d_H}(v, b)}_{\leq t} \geq t + 1.$$

Conclusão: a palavra recebida b está mais próxima da palavra-código transmitida v do que de qualquer outra palavra-código $v' \neq v$ e, portanto, o decodificador de mínima distância irá selecionar a palavra-código correta v.

Parte 2. Agora, sejam v e v' tais que $\mathrm{d_H}(v, v') = d$ e defina $\delta = v + v'$. O peso de δ é $\mathrm{w_H}(\delta) = \mathrm{w_H}(v + v') = \mathrm{d_H}(v, v') = d$, ou seja, $2t + 1$ se d é ímpar, ou $2t + 2$, se d é par.



Seja e de peso $t+1$ com bits 1 apenas onde δ também possui bits 1. Suponha que v seja a palavra-código transmitida e $b = v + e$ seja a palavra recebida. Temos que

$$\mathrm{d_H}(v, b) = \mathrm{w_H}(e) = t + 1,$$

ao passo que

$$\mathrm{d_H}(v', b) = \mathrm{d_H}(v + \delta, v + e) = \mathrm{d_H}(\delta, e) = \begin{cases} t, & d \text{ ímpar}, \\ t+1, & d \text{ par}. \end{cases}$$

Assim, e é um padrão de erro de peso $t+1$ que não pode ser corrigido, pois:

- Para d ímpar, a palavra recebida b está mais próxima de v' do que de v. Nesse caso, o decodificador de mínima distância *irá com certeza selecionar* a palavra-código errada v'.
- Para d par, a palavra recebida b está equidistante de v e v'. Nesse caso, o decodificador de mínima distância *poderá selecionar* a palavra-código errada v'. ■

Capacidades de correção de erro de alguns códigos

Código	d	t
Paridade simples $(3, 2)$	2	0
Repetição $(3, 1)$	3	1
Hamming $(7, 4)$	3	1
Golay $(23, 12)$	7	3
Reed–Muller $(32, 6)$	16	7

Curiosidade: É possível demonstrar que, para um código *linear* de comprimento $n = 12$ e dimensão $k = 5$ (ou seja, com 32 palavras-código), a melhor distância mínima possível é $d = 4$ (resultando em $t = 1$). No entanto, em 1962, M. Nadler publicou a construção de um código *não-linear* de mesmo comprimento e mesmo número de palavras-código, mas com $d = 5$ (resultando em $t = 2$).

Referências

Referências

[1] S. Haykin, *Communication Systems*, 4th ed. John Wiley & Sons, 2001.

[2] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[3] S. Lin and D. J. Costello Jr., *Error Control Coding*, 2nd ed. Pearson Prentice Hall, 2004.