

2 Codificação

Este capítulo apresenta o codificador turbo proposto em [4], cuja estrutura era composta por dois codificadores convolucionais sistemáticos recursivos concatenados em paralelo, com um entrelaçador entre eles. Também é abordada a técnica de puncionamento utilizada em [4], com intuito de se obter codificadores turbo com diferentes taxas a partir de um mesmo codificador.

O objetivo deste capítulo é portanto revisar a teoria e conceitos fundamentais relacionados aos elementos relativos à codificação — codificadores, entrelaçadores, técnica de puncionamento — assim como os tipos de esquema, nos quais esse elementos se estruturam — concatenação série ou paralelo. O capítulo é organizado da seguinte forma. Na Seção 2.1 são introduzidas definições fundamentais sobre a teoria de códigos convolucionais.

A técnica de puncionamento é explicada na Seção 2.2. Alguns entrelaçadores apresentados comumente na literatura [6] [4] [14], são descritos na Seção 2.3.1. Os princípios da concatenação de codificadores são abordados nas seções 2.3.3 e 2.3.2 respectivamente. Por fim, o codificador turbo é apresentado na Seção 2.4.

2.1 Códigos Convolucionais

Os códigos convolucionais foram introduzidos por Elias em 1955. Por apresentarem um ótimo desempenho, quando comparados aos códigos de blocos de mesma complexidade, e uma fácil implementação, os códigos convolucionais tornaram-se presentes na etapa de codificação proposta em vários padrões de telecomunicações.

No desenvolvimento deste trabalho, foram utilizados códigos convolucionais binários, como proposto em [4], sendo os elementos e as operações considerados no processo de codificação pertencentes ao corpo \mathbb{F}_2 .

2.1.1

Definições

Considere que um *codificador convolucional* recebe a cada instante de tempo t , um bloco de k bits na entrada, ou seja, o vetor

$$\mathbf{m}_t = \begin{pmatrix} m_t^{(0)} \\ \vdots \\ m_t^{(k-1)} \end{pmatrix}$$

e, produz um bloco de n bits de saída, ou seja, o vetor

$$\mathbf{c}_t = \begin{pmatrix} c_t^{(0)} \\ \vdots \\ c_t^{(n-1)} \end{pmatrix}.$$

A taxa de codificação é $R = k/n$. As seqüências de entrada e saída, para os instantes $t = 0, \dots, N - 1$, são representadas por

$$\begin{aligned} \mathbf{m} &= \left(\mathbf{m}_0 \quad \cdots \quad \mathbf{m}_{N-1} \right) \\ \mathbf{c} &= \left(\mathbf{c}_0 \quad \cdots \quad \mathbf{c}_{N-1} \right). \end{aligned} \quad (2-1)$$

Pode-se destacar ainda as seqüências

$$\begin{aligned} \mathbf{m}^{(i)} &= \left(m_0^{(i)} \quad \cdots \quad m_{N-1}^{(i)} \right) \\ \mathbf{c}^{(j)} &= \left(c_0^{(j)} \quad \cdots \quad c_{N-1}^{(j)} \right). \end{aligned} \quad (2-2)$$

Um codificador convolucional pode ser visualizado como um circuito ou um conjunto de registradores com memória ν , onde as seqüências geradas são obtidas através de operações lógicas¹ entre os bits armazenados nos registradores (bits passados) e os bits de entrada (bits atuais). A partir deste comentário vale ressaltar a seguinte definição:

Definição 2.1 (Memória) A memória ν de um codificador é igual ao número de bits de entrada que podem ser armazenados em seus registradores. ■

O codificador convolucional da Fig. 2.1, por exemplo, recebe, a cada

¹No caso binário, ou-exclusivo.

instante t , um bit² de entrada m_t , e gera como saída $c_t^{(0)}$ e $c_t^{(1)}$, conforme explicitam as equações (2-3) e (2-4).

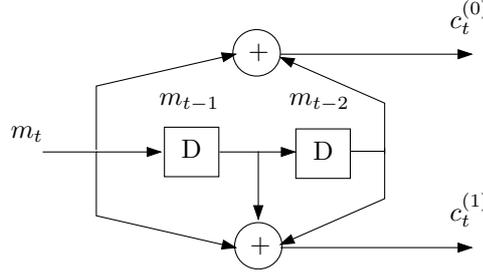


Figura 2.1: Codificador convolucional com $R = 1/2$.

$$c_t^{(0)} = m_t \oplus m_{t-2} \quad (2-3)$$

e,

$$c_t^{(1)} = m_t \oplus m_{t-1} \oplus m_{t-2}. \quad (2-4)$$

A título de ilustração, considere a seqüência de entrada \mathbf{m} , de tamanho $N = 7$

$$\mathbf{m} = \left(1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \right). \quad (2-5)$$

A seqüência de vetores na saída, considerando que o registrador é inicializado com zeros, é

$$\begin{aligned} \mathbf{c}^{(0)} &= \left(1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \right) \\ \mathbf{c}^{(1)} &= \left(1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \right). \end{aligned} \quad (2-6)$$

Os códigos convolucionais são códigos lineares, o que permite analisar sua estrutura utilizando várias técnicas da teoria de sistemas lineares. O codificador, neste contexto, pode ser representado como um conjunto de filtros digitais binários, e cada seqüência binária de saída $\mathbf{c}^{(j)}$, $j = 0, \dots, n-1$, se relaciona a respectiva seqüência de entrada $\mathbf{m}^{(i)}$, $i = 0, \dots, k-1$, através de uma função transferência

$$\mathbf{g}_i^{(j)} = \left(g_{i0}^{(j)} \ \cdots \ g_{ir_i}^{(j)} \right) \quad (2-7)$$

de comprimento $r_i + 1$.

A função transferência $\mathbf{g}_i^{(j)}$ é a resposta $\mathbf{c}^{(j)}$, ao impulso $\mathbf{m}^{(i)} = (1 \ 0 \ 0 \ \cdots)$. A função transferência $\mathbf{g}_i^{(j)}$ é também conhecida como

²Para codificadores convolucionais com taxa $R = 1/n$, optou-se por omitir o índice (0), a fim de simplificar a notação.

seqüência geradora, e pode ser facilmente obtida através da conferência das ligações entrada/saída implementadas no codificador.

No caso do codificador da Fig. 2.1, as seqüências geradoras são

$$\mathbf{g}^{(0)} = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \quad (2-8)$$

e

$$\mathbf{g}^{(1)} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}. \quad (2-9)$$

(Observe que o índice i foi omitido, pois o codificador da Fig. 2.1 possui uma única entrada.) Note que as seqüências $\mathbf{g}^{(0)}$ e $\mathbf{g}^{(1)}$ são finitas — codificadores convolucionais em que todas as seqüências $\mathbf{g}_i^{(j)}$ são finitas, recebem a denominação de *codificadores FIR* (*Finite Impulse Response*).

Na literatura, é comum representar as polinômios geradores na forma octal, onde cada trio de bits representa um inteiro de 0 a 7, desta maneira, $g^{(0)} = 5$ e $g^{(1)} = 7$.

Considerando a função transferência $\mathbf{g}_i^{(j)} = (g_{i0}^{(j)}, \dots, g_{ir_i}^{(j)})$, a saída $c_t^{(j)}$ em determinado instante t correspondente a entrada $m_t^{(i)}$ é dada por

$$c_t^{(j)} = \sum_{i=0}^{k-1} \sum_{l=0}^{r_i} m_{t-l}^{(i)} g_{il}^{(j)} \quad (2-10)$$

que consiste em uma convolução entre $\mathbf{m}^{(i)}$ e $\mathbf{g}_i^{(j)}$, justificando a denominação do código. Um conceito importante para codificação convolucional deriva de (2-7)

Definição 2.2 (*Constraint Length*) ³ A *constraint length* K de um codificador é definida como

$$K = 1 + \max_i(r_i). \blacksquare \quad (2-11)$$

Um codificador convolucional com taxa $R = k/n$ e *constraint length* K é representado neste trabalho pela notação (n, k, K) . Um *código convolucional* \mathcal{C} com parâmetros (n, k, K) , também denotado por $\mathcal{C} = (n, k, K)$, é definido como o conjunto de todas as seqüências de saída (palavras-código) produzidas por um codificador convolucional (n, k, K) .

Pode-se usualmente expressar a codificação convolucional no anel⁴ polinomial $\mathbb{F}_2[x]$, onde as seqüências de entrada, de saída e geradora são

³As definições 2.1 e 2.2 variam de autor a autor, por exemplo vide p. 55, 73 em [15].

⁴A notação $\mathbb{F}_2[x]$ é utilizada para denotar o conjunto de todos os polinômios $m_0 + m_1x + m_2x^2 + \dots + m_nx^n$ de grau arbitrário, com os coeficientes $\{m_i\}$ pertencentes ao corpo finito \mathbb{F}_2 .

representadas como polinômios. O vetor

$$\mathbf{m} = \left(m_0 \ m_1 \ m_2 \ m_3 \ m_4 \ \cdots \right)$$

seria representado, neste anel, como

$$m(x) = \sum_{l=0}^{\infty} m_l x^l. \quad (2-12)$$

Ainda para o caso do codificador da Fig. 2.1, tem-se que

$$m(x) = 1 + x + x^4 + x^6 \longleftrightarrow \mathbf{m} = \left(1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \right) \quad (2-13)$$

e

$$\begin{aligned} g^{(0)}(x) &= 1 + x^2 \\ g^{(1)}(x) &= 1 + x + x^2. \end{aligned} \quad (2-14)$$

As saídas são

$$\begin{aligned} c^{(0)}(x) &= m(x)g^{(0)}(x) = 1 + x + x^2 + x^3 + x^4 + x^8 \\ c^{(1)}(x) &= m(x)g^{(1)}(x) = 1 + x^3 + x^4 + x^5 + x^7 + x^8. \end{aligned} \quad (2-15)$$

Os polinômios geradores podem ser organizados em uma matriz $k \times n$, denominada matriz geradora, ou matriz função de transferência, que para codificador da Fig. 2.1 é

$$\mathbf{G}(x) = \left(g^{(0)}(x) \ g^{(1)}(x) \right) = \left(1 + x^2 \ 1 + x + x^2 \right). \quad (2-16)$$

Para codificadores com taxa $R = \frac{k}{n}$, $k > 1$, tem-se

$$\mathbf{m}(x) = \begin{pmatrix} m^{(0)}(x) \\ \vdots \\ m^{(k-1)}(x) \end{pmatrix} \quad (2-17)$$

e

$$\mathbf{G}(x) = \begin{pmatrix} g_0^{(0)}(x) & g_0^{(1)}(x) & \cdots & g_0^{(n-1)}(x) \\ g_1^{(0)}(x) & g_1^{(1)}(x) & \cdots & g_1^{(n-1)}(x) \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1}^{(0)}(x) & g_{k-1}^{(1)}(x) & \cdots & g_{k-1}^{(n-1)}(x) \end{pmatrix} \quad (2-18)$$

com

$$\mathbf{c}(x) = \begin{pmatrix} c^{(0)}(x) \\ \vdots \\ c^{(n-1)}(x) \end{pmatrix} = \mathbf{m}^\top(x) \mathbf{G}(x). \quad (2-19)$$

A matriz geradora $\mathbf{G}(x)$, sendo um parâmetro importante na classificação dos codificadores convolucionais, é examinada mais detalhadamente a seguir [31].

Definição 2.3 (Codificador Polinomial ou Não-Recursivo) O codificador que corresponde a uma matriz geradora $\mathbf{G}(x)$ que possui apenas elementos polinomiais é dito polinomial ou não-recursivo. ■

Como exemplo, vide o codificador da matriz da Eq. (2-16).

Definição 2.4 (Codificador Sistemático) O codificador que corresponde a uma matriz geradora $\mathbf{G}(x)$ é dito sistemático, se esta matriz puder ser escrita na forma $\mathbf{G}(x) = \begin{pmatrix} \mathbf{I} & \mathbf{P}(x) \end{pmatrix}$, onde \mathbf{I} é a matriz identidade. ■

A matriz correspondente a este codificador é também dita sistemática. Por exemplo, a matriz geradora

$$\mathbf{G}(x) = \begin{pmatrix} 1 & 0 & 1 + x^2 + x^3 \\ 0 & 1 & x + x^2 \end{pmatrix} \quad (2-20)$$

caracteriza um codificador convolucional sistemático de taxa $R = 2/3$, com

$$\begin{aligned} c^{(0)}(x) &= m^{(0)}(x) \\ c^{(1)}(x) &= m^{(1)}(x) \\ c^{(2)}(x) &= m^{(0)}(x)(1 + x^2 + x^3) + m^{(1)}(x)(x + x^2). \end{aligned} \quad (2-21)$$

de acordo com a Eq. (2-19).

Outro tipo interessante de codificador é ilustrado na Fig. 2.2. O conteúdo do registrador deste codificador é realimentado com os bits $m_{t-3}^{(0)}$ e $m_{t-3}^{(1)}$. Denomina-se tal codificador de codificador recursivo. Este codificador pode ser caracterizado por uma matriz geradora que possui elementos racionais diretamente relacionados às ligações de realimentação implementadas. A definição de codificador convolucional recursivo é dada abaixo:

Definição 2.5 (Codificador Racional ou Recursivo) O codificador que corresponde a uma matriz geradora $\mathbf{G}(x)$ que possui elementos racionais é dito racional ou recursivo. ■

Por exemplo, a matriz referente ao codificador da Fig. 2.2 é dada por

$$\mathbf{G}(x) = \begin{pmatrix} 1 & 0 & \frac{x}{1+x^3} \\ 0 & 1 & \frac{x^2}{1+x^4} \end{pmatrix} \quad (2-22)$$

conforme explicita-se.

Os elementos da matriz geradora de um codificador convolucional recursivo são da forma $g_i^{(j)}(x) = n(x)/d(x)$, onde o numerador $n(x)$ está associado às ligações que determinam a saída, e o denominador $d(x)$ às ligações que determinam os bits realimentados.

Para a seqüência $g_0^{(2)}(x)$ da matriz geradora $\mathbf{G}(x)$ da Eq. (2-22) tem-se que

$$\begin{aligned} n(x) = x &\longleftrightarrow \mathbf{n} = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix} \\ d(x) = 1 + x^3 &\longleftrightarrow \mathbf{d} = \begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix} \end{aligned} \quad (2-23)$$

o mesmo vale para $g_1^{(2)}(x)$.

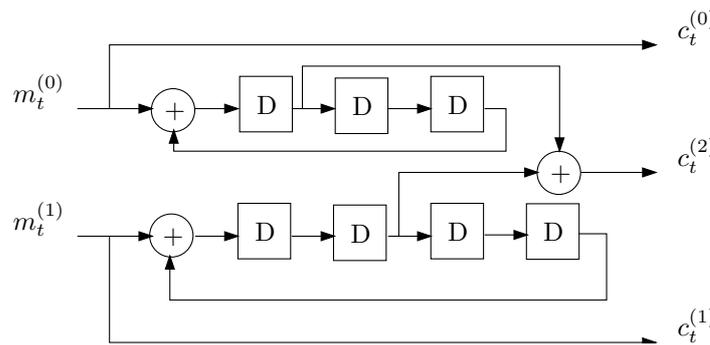


Figura 2.2: Codificador convolucional correspondente à matriz em (2-22), $R = 2/3$.

Note que, se as divisões dos polinômios geradores $g_0^{(2)}(x)$ e $g_1^{(2)}(x)$ forem efetuadas, o resultado corresponderá a uma seqüência infinita para ambas, por essa razão, codificadores convolucionais recursivos são também denominados de *codificadores IIR (Infinite Impulse Response)*.

As seqüências codificadas para codificadores convolucionais recursivos também são obtidas usando a Eq. (2-19).

O codificador da Fig. 2.2, além de recursivo é sistemático, visto que é possível identificar uma matriz identidade em sua matriz geradora.

Outra definição importante na caracterização de codificadores convolucionais é

Definição 2.6 (Distância Livre) Considere um código convolucional \mathcal{C} com parâmetros (n, k, K) truncado em $t = N - 1$, isto é, as seqüências de

entrada $\mathbf{m}^{(i)} = \left(m_0^{(i)} \ \dots \ m_{N-1}^{(i)} \right)$ e de saída $\mathbf{c}^{(j)} = \left(c_0^{(j)} \ \dots \ c_{N-1}^{(j)} \right)$, para $i = 0, \dots, k-1$ e $j = 0, \dots, n-1$, possuem tamanho finito N . Sendo a distância livre dada por

$$d_{\text{free}} = \lim_{N \rightarrow \infty} (d_N) \quad (2-24)$$

com d_N

$$\begin{aligned} d_N &= \min\{d_H(\mathbf{c}, \mathbf{c}') | \mathbf{c} \neq \mathbf{c}'\} \\ &= \min\{w(\mathbf{c}) | \mathbf{c} \neq \mathbf{0}\} \end{aligned} \quad (2-25)$$

onde $d_H(\mathbf{c}, \mathbf{c}')$ é a distância de *Hamming*⁵ entre \mathbf{c} e \mathbf{c}' , e $w(\mathbf{c})$ é o peso⁶ da palavra código \mathbf{c} , onde $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$. ■

2.1.2 Representações em Grafos

A utilização de grafos na representação do processo de codificação/decodificação consiste em uma ferramenta útil para a compreensão e desenvolvimento de algoritmos de decodificação. No caso dos codificadores convolucionais, o grafo utilizado é denominado de *treliça*.

A discussão da estrutura de uma treliça, introduz alguns conceitos fundamentais relativos a teoria de grafos [31]:

Definição 2.7 (Grafo) Um grafo \mathcal{G} é definido por um par de conjuntos (V, E) , onde V é um conjunto finito, não-vazio, de vértices⁷ designados por v_j , $j = 1, \dots, |V|$, denominado *conjunto de vértices*, e E é uma família finita de pares não-ordenados dos elementos de V . Estes pares são denominados de *ramos* e designados por $e(v_j, v_k)$, $v_j, v_k \in V$. ■

Na Fig. 2.3, é ilustrado o grafo com $V = \{a, b, c, d\}$ e $E = \{e(a, b), e(a, c), e(b, c), e(c, d), e(d, d)\}$.

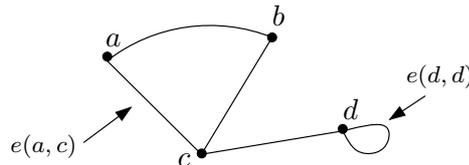


Figura 2.3: Grafo genérico.

⁵ $d_H(\mathbf{c}_1, \mathbf{c}_2) = |\{t | c_{1t}^{(j)} \neq c_{2t}^{(j)}, t = 0, 1, \dots, N-1; j = 0, \dots, n-1\}|$.

⁶ $w(\mathbf{c}) = |\{t | c_t \neq 0, t = 0, 1, \dots, N-1\}|$.

⁷Vértices podem usualmente receber a denominação de nós.

Definição 2.8 (Vértices Adjacentes) Diz-se que dois vértices v_j e v_k de \mathcal{G} são adjacentes se há um ramo $e(v_j, v_k)$ conectando estes vértices. ■

É comum também a denominação de vértices ou nós *vizinhos* para vértices adjacentes.

Definição 2.9 (Ramos Adjacentes) Diz-se que dois ramos são adjacentes se eles possuem no mínimo um vértice em comum. ■

Definição 2.10 (Grau de um Vértice) O grau de um vértice é definido como o número total de ramos incidentes nele. ■

Definição 2.11 (Percurso em um Grafo) Um percurso em um grafo \mathcal{G} é uma seqüência finita de ramos distintos representada por $\{e(v_0, v_1), e(v_1, v_2), \dots, e(v_{m-1}, v_m)\}$, com início no nó v_0 e término no nó v_m . ■

Definição 2.12 (Circuito ou Ciclo) Se $v_m = v_0$ o percurso é denominado de circuito ou ciclo. ■

Por exemplo, no grafo da Fig. 2.3, $\{e(a, b), e(b, c), e(c, d)\}$ é um percurso, e $\{e(a, b), e(b, c), e(c, a)\}$ é um ciclo.

Definição 2.13 (Grafo Conectado) Um grafo conectado é aquele que possui um percurso entre quaisquer vértices $v_j, v_k \in V$, $j \neq k$. ■

Definição 2.14 (Árvore ou Grafo sem ciclos) Uma árvore é um grafo conectado que não contém circuitos. Um vértice de uma árvore é denominado de *folha* ou *nó terminal*, se seu grau é igual a 1. ■

Definição 2.15 (Grafo Bi-particionado) É o grafo no qual o conjunto de vértices pode ser particionado em dois conjuntos disjuntos, V_1 e V_2 , tal que todo ramo pertencente ao grafo \mathcal{G} une um vértice de V_1 a um vértice de V_2 . Os vértices contidos em V_1 não estão conectados entre si, o mesmo vale para os vértices de V_2 . ■

Definição 2.16 (Grafo Orientado de Ramo Rotulado) ⁸ É um trio (V, E, \mathcal{A}) , consistindo em um conjunto de vértices V , um conjunto finito \mathcal{A} , denominado de alfabeto, e um conjunto de ramos E , onde cada ramo é associado a um trio ordenado (v, a, v') , com $v, v' \in V$ e $a \in \mathcal{A}$. Um ramo $e(v, a, v') \in E$ inicia no vértice v , termina no vértice v' , e é rotulado de a . ■

⁸A Def. 2.16 pode ser generalizada para uma ênupla $(V, E, \mathcal{A}^0, \dots, \mathcal{A}^n)$.

Definição 2.17 (Treliza) Uma treliza $\mathcal{T}(V, E, \mathcal{A}^m, \mathcal{A}^c)$ de profundidade N , para um código convolucional, é um grafo orientado de ramo rotulado com a seguinte propriedade:

$$V = V_0 \cup V_1 \cup \dots \cup V_N \quad (2-26)$$

de modo que cada ramo de \mathcal{T} inicia em um vértice de V_t e termina em um vértice de V_{t+1} , para $t = 0, 1, \dots, N - 1$. Os conjuntos V_0, V_1, \dots, V_N são denominados de *classes de vértices* de \mathcal{T} . O conjunto de índices ordenados $\{0, 1, \dots, N\}$, induzido pela partição da Eq. (2-26), é denominado de *eixo de instantes de tempo* para \mathcal{T} . ■

Considere um codificador convolucional (n, k, K) cujas entradas e saídas no instante t são dadas por \mathbf{m}_t e \mathbf{c}_t respectivamente. O codificador pode ser visualizado como uma máquina de estados, onde cada estado σ_t corresponde aos bits armazenados nos registradores no instante t . Sendo o código binário, o número total de possíveis estados é $Q = 2^{K-1}$, ou seja, $\sigma_t \in \{0, 1, \dots, Q - 1\} \forall t$.

Uma treliza \mathcal{T} pode ser vista como a união de N seções, T_0, \dots, T_{N-1} , onde cada seção é definida como um subconjunto $T_t \subset V_t \times A_t^m \times A_t^c \times V_{t+1}$ que especifica as possíveis combinações $(v_t, \mathbf{m}_t, \mathbf{c}_t, v_{t+1})$; onde os vértices $v_t \in V_t$ e $v_{t+1} \in V_{t+1}$ correspondem aos estados σ_t e σ_{t+1} , respectivamente; enquanto que A_t^m e A_t^c correspondem aos alfabetos de \mathbf{m}_t e \mathbf{c}_t .

Note que cada seção T_t da treliza corresponde a um subgrafo $\mathcal{T}_t(V_t, E_t)$, na qual os vértices correspondem aos possíveis estados, e os ramos todas as possíveis transições de estados (σ_t, σ_{t+1}) . Cada ramo é rotulado respectivamente com a entrada \mathbf{m}_t que provoca a transição (σ_t, σ_{t+1}) , e a saída \mathbf{c}_t gerada pela condição (\mathbf{m}_t, σ_t) .

A treliza do código, quando utilizada como ferramenta para solução dos problemas de decodificação, pode ser vista como uma descrição causal do processo probabilístico responsável pela geração de uma palavra código. Considera-se que o tempo flui da esquerda para direita, e em cada instante de tempo os bits mensagem, gerados por um fonte binária aleatória, determinam que direção seguir, descrevendo um percurso na treliza, equivalente a uma palavra do código.

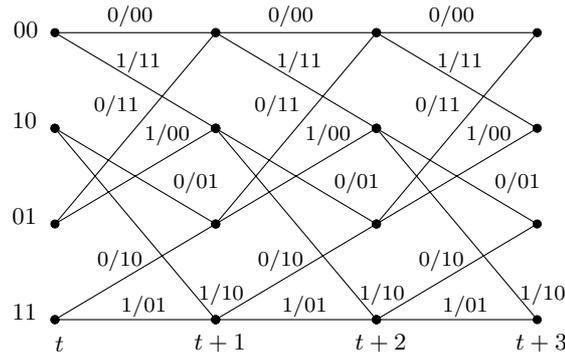


Figura 2.4: Quatro estágios da treliça para o codificador da Fig. 2.1.

2.2

Códigos Convolucionais Modificados por Puncionamento

Dado um código $\mathcal{C} = (n, k, K)$, códigos de altas taxas são obtidos a partir de \mathcal{C} , através da supressão⁹ periódica de bits em uma ou mais seqüências de saída de \mathcal{C} [14]. Essa técnica é denominada de puncionamento.

Por exemplo, dado um código $\mathcal{C} = (2, 1, K)$, onde a seqüência de entrada \mathbf{m} e a palavra código correspondente \mathbf{c} são dadas por

$$\mathbf{m} = \begin{pmatrix} m_0 & m_1 & m_2 & m_3 & m_4 & \dots \end{pmatrix} \quad (2-27)$$

$$\mathbf{c} = \begin{pmatrix} c_0^{(0)} & c_1^{(0)} & c_2^{(0)} & c_3^{(0)} & c_4^{(0)} & \dots \\ c_0^{(1)} & c_1^{(1)} & c_2^{(1)} & c_3^{(1)} & c_4^{(1)} & \dots \end{pmatrix}.$$

Se a cada três bits, o quarto é apagado, a nova palavra código associada a \mathbf{m} seria

$$\mathbf{c} = \begin{pmatrix} c_0^{(0)} & c_1^{(0)} & c_2^{(0)} & c_3^{(0)} & c_4^{(0)} & \dots \\ c_0^{(1)} & - & c_2^{(1)} & - & c_4^{(1)} & \dots \end{pmatrix} \quad (2-28)$$

onde o símbolo “-” foi inserido para indicar que aquele bit não foi transmitido.

O puncionamento pode ser representado por uma matriz \mathbf{P} de n linhas e período T

$$\mathbf{P} = \begin{pmatrix} p_{0,0} & \dots & p_{0,T-1} \\ \vdots & \ddots & \vdots \\ p_{n-1,0} & \dots & p_{n-1,T-1} \end{pmatrix} \quad (2-29)$$

com $p_{i,j} \in \{0, 1\}$ para $i = 0, \dots, n-1$ e $j = 0, \dots, T-1$, onde 0 implica que o bit foi puncionado e conseqüentemente não transmitido.

⁹Neste contexto supressão significa não transmissão de determinado bit.

Cada linha de \mathbf{P} refere-se a uma seqüência de saída, enquanto que as colunas determinam quais bits são puncionados em subseqüências de comprimento T de \mathbf{c} .

A taxa R_p do código obtido através do puncionamento, é dada por

$$R_p = \frac{T}{w(\mathbf{P})}. \quad (2-30)$$

Para o exemplo acima tem-se

$$\mathbf{P} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad (2-31)$$

com $T = 2$ e $R_p = \frac{2}{3}$.

O mesmo decodificador usado para \mathcal{C} , denominado neste contexto de *código-mãe*, pode ser utilizado na decodificação do código puncionado \mathcal{C}_p , basta que para isso, o decodificador seja projetado para inserir no lugar dos bits puncionados, uma informação neutra ao algoritmo de decodificação.

A representação do código puncionado \mathcal{C}_p através da treliça do código-mãe \mathcal{C} é mostrada na Fig. 2.5.

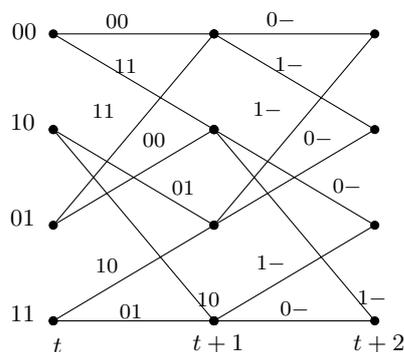


Figura 2.5: Dois estágios da treliça do código-mãe representando um estágio da treliça do código puncionado.

Os dois principais benefícios de se utilizar um código convolucional puncionado \mathcal{C}_p , obtido a partir do código-mãe \mathcal{C} são:

- Redução da complexidade na decodificação para códigos de altas taxas, visto que o decodificador utilizado é o do código mãe, mais simples do que um decodificador que usasse diretamente a treliça do código \mathcal{C}_p ;
- Possibilidade da geração de códigos de taxas diferentes a partir de um mesmo codificador-mãe.

Esses dois benefícios fizeram com que a técnica de puncionamento se tornasse presente em vários padrões das telecomunicações, sendo inclusive empregada na codificação turbo.

2.3

Codificadores Concatenados

Sistemas concatenados consistem em um arranjo de dois ou mais codificadores com o intuito de gerar um código com palavras códigos maiores com boas propriedades, e com complexidade de decodificação menor, se comparado a um código de um único estágio, que produzisse palavras códigos de mesmo tamanho.

Sistemas concatenados são uma tentativa de se alcançar o que foi proposto por Shannon, a um nível de complexidade praticável.

Os codificadores neste arranjo podem ser concatenados de duas formas: em série ou em paralelo. Em qualquer um dos casos, pode-se ainda utilizar um entrelaçador entre codificadores, com o objetivo de melhorar o desempenho do esquema.

Nesta seção, primeiramente, são revisados os principais conceitos sobre entrelaçamento, são abordados também alguns entrelaçadores, que fornecem noções interessantes sobre o desempenho de codificadores turbo. Depois, são introduzidas as estruturas dos dois esquemas, discutindo-se brevemente o papel do entrelaçador em cada um deles.

2.3.1

Entrelaçador

Um entrelaçador Π é um dispositivo de uma única entrada e uma única saída, cuja função é rearranjar a seqüência de entrada em uma ordem temporal diferente na saída. O inverso desse processo é realizado pelo desentrelaçador Π^{-1} , sendo este, então, responsável por restaurar a seqüência de entrada.

Considere a seqüência de entrada de tamanho N

$$\mathbf{m} = \left(m_0 \quad m_1 \quad \cdots \quad m_{N-1} \right) \quad (2-32)$$

onde $m_t \in \{0, 1\}$, $0 \leq t \leq N - 1$. O entrelaçador Π , ilustrado pelo diagrama de blocos da Fig. 2.6, permuta \mathbf{m} resultando em

$$\tilde{\mathbf{m}} = \begin{pmatrix} \tilde{m}_0 & \tilde{m}_1 & \cdots & \tilde{m}_{N-1} \end{pmatrix} \quad (2-33)$$

onde $\tilde{m}_{t'} \in \{0, 1\}$, $0 \leq t' \leq N - 1$.



Figura 2.6: Modelo de entrelaçador.

Definindo um conjunto $\mathbb{Z}_N = \{0, 1, \dots, N - 1\}$. O entrelaçador pode ser definido por uma função de mapeamento dada por [20]

$$\pi(\mathbb{Z}_N \rightarrow \mathbb{Z}_N) : t' = \pi(t), t, t' \in \mathbb{Z}_N \quad (2-34)$$

desse modo $m_t = \tilde{m}_{t'}$.

A função mapeamento pode ser representada através do vetor de entrelaçamento $\boldsymbol{\pi}$ que guarda para m_t a sua nova posição em $\tilde{\mathbf{m}}$, e é representado por

$$\boldsymbol{\pi} = \begin{pmatrix} \pi(0) & \pi(1) & \cdots & \pi(N - 1) \end{pmatrix}. \quad (2-35)$$

Considerando que Π tenha tamanho $N = 5$ e apresente na sua saída a seqüência

$$\tilde{\mathbf{m}} = \begin{pmatrix} \tilde{m}_0 & \tilde{m}_1 & \tilde{m}_2 & \tilde{m}_3 & \tilde{m}_4 \end{pmatrix} = \begin{pmatrix} m_1 & m_3 & m_0 & m_4 & m_2 \end{pmatrix}. \quad (2-36)$$

O vetor de entrelaçamento é dado por

$$\boldsymbol{\pi} = \begin{pmatrix} \pi(0) & \pi(1) & \pi(2) & \pi(3) & \pi(4) \end{pmatrix} = \begin{pmatrix} 2 & 0 & 4 & 1 & 3 \end{pmatrix}. \quad (2-37)$$

o que é ilustrado na Fig. 2.7.

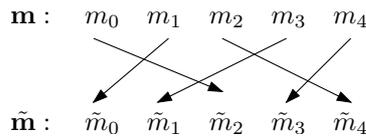


Figura 2.7: Função de entrelaçamento.

Dentre os vários tipos de entrelaçadores existentes, optou-se por utilizar nas simulações deste trabalho, os entrelaçadores: de bloco, pseudo-

aleatório e pseudo-aleatório com parâmetro S cujas estruturas são detalhadas a seguir.

Entrelaçador de Bloco

No entrelaçador de bloco, a seqüência de entrada \mathbf{m} de tamanho N , é armazenada em uma matriz \mathbf{B} de L linhas e C colunas, tal que $N = L \times C$.

No entrelaçamento de blocos, tudo se passa como se a seqüência \mathbf{m} fosse escrita em \mathbf{B} no sentido das linhas, e a seqüência $\tilde{\mathbf{m}}$ obtida a partir de uma leitura no sentido das colunas. A função mapeamento para este entrelaçador pode ser representada por:

$$\pi(t) = [(t - 1) \bmod C] \times L + \lfloor (t - 1) / C \rfloor + 1, \quad t \in \mathbb{Z}_N. \quad (2-38)$$

O desentrelaçador, como dito anteriormente, tem a mesma estrutura e realiza a operação inversa. Costuma-se chamar o número de linhas da matriz \mathbf{B} de *profundidade do entrelaçador*, enquanto o número de colunas é denominado de *extensão do entrelaçador*.

Entrelaçador Pseudo-Aleatório

No entrelaçador pseudo-aleatório a seqüência de entrada \mathbf{m} é escrita e lida em um bloco de tamanho N de maneira aleatória. O vetor de entrelaçamento $\boldsymbol{\pi}$, pode ser obtido a partir do seguinte algoritmo de N passos:

Algoritmo 2.1 (Entrelaçador Pseudo-Aleatório)

- 1) Para $t = 0$, escolha aleatoriamente um inteiro a_0 do conjunto \mathbb{Z}_N , de acordo com uma distribuição uniforme, com probabilidade $p(a_0) = \frac{1}{N}$. Faz-se $\pi(0) = a_0$ e especifica-se o conjunto $A_0 = \{a_0\}$.
- 2) Para $0 < t < N - 1$, escolha aleatoriamente um inteiro a_t pertencente ao conjunto $A_t = \mathbb{Z}_N \cap \bar{A}_{t-1}$, de acordo com uma distribuição uniforme, com probabilidade $p(a_t) = \frac{1}{N-t}$. O inteiro escolhido a_t é atribuído a $\pi(t)$.
- 3) Para $t = N - 1$, o último inteiro restante a_{N-1} é atribuído a $\pi(N - 1)$.
- 4) Fim. ■

Entrelaçador Pseudo-Aleatório com Parâmetro S

O entrelaçador aleatório com parâmetro S foi proposto por Divsalar e Pollara em [6]. O vetor de entrelaçamento desse entrelaçador é constituído a partir da geração aleatória de N inteiros pertencentes ao conjunto \mathbb{Z}_N , associada a um parâmetro S , denominado de *distância mínima de entrelaçamento*.

O vetor de entrelaçamento é obtido da seguinte forma: cada inteiro a_t , selecionado, é comparado aos S inteiros, $\{a_{t-1}, a_{t-2}, \dots, a_{t-S}\}$, previamente selecionados. Se $|a_t - a_{t-k}| < S$ para $k = 1, 2, \dots, S$, a_t é descartado e um novo inteiro é selecionado para ser testado. Esse processo é repetido até que os N inteiros tenham sido selecionados.

O tempo de busca gasto na composição do vetor de entrelaçamento aumenta com S , foi observado em [6], que escolher $S < \sqrt{N/2}$, apresenta um bom custo-benefício entre tempo de busca e o fator de dispersão¹⁰. Note que se $S = 1$, o entrelaçador obtido é o entrelaçador pseudo-aleatório.

Mais informações sobre a influência que um entrelaçador possui no desempenho dos códigos turbo são abordadas no Capítulo 5.

2.3.2 Codificadores Concatenados em Série

Codificadores concatenados em série foram introduzidos por Forney em 1965. Nesse tipo de concatenação, os codificadores são justapostos serialmente. Por exemplo, no diagrama de blocos da Fig. 2.8.

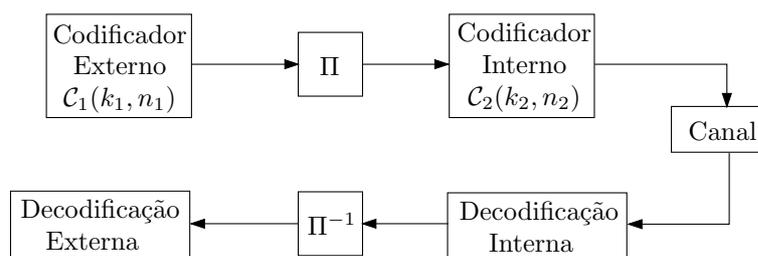


Figura 2.8: Modelo de codificadores concatenados em série.

A seqüência de entrada é primeiramente codificada pelo *codificador externo*, que recebe blocos de k_1 símbolos e produz blocos de n_1 símbolos. Após essa etapa, o *codificador interno* recebe blocos de $k_2 = n_1$ símbolos,

¹⁰O fator de dispersão está associado à distância entre as posições $\pi(t)$ e $\pi(t+1)$, para $t = 0, \dots, N-1$. Quanto maior a distância, maior será a dispersão provocada pelo entrelaçador.

provenientes do codificador interno, e gera em sua saída blocos de n_2 símbolos.

Nesse esquema, os bits são entrelaçados na transmissão, para que possam ser desentrelaçados, “embaralhados”, na recepção. O desentrelaçador, nesse caso, tem como função tentar dispersar possíveis rajadas de erro provenientes da decodificação interna, distribuindo os bits errados ao longo da seqüência recebida, ajudando dessa forma a decodificação externa.

2.3.3 Codificadores Concatenados em Paralelo

A concatenação em paralelo foi introduzida em [4], e consiste em um arranjo no qual dois ou mais codificadores são justapostos paralelamente com entrelaçadores entre eles. A diferença entre a concatenação série e a paralela, está no fato de que nesta última todos os codificadores recebem em suas entradas versões permutadas de uma mesma seqüência de entrada. Os códigos correspondentes aos codificadores utilizados neste esquema são usualmente denominados de *códigos componentes*.

2.4 Códigos Turbo

Códigos turbo podem ser representados por codificadores que são construídos a partir da concatenação em série ou em paralelo de um ou mais codificadores, empregando entrelaçadores entre eles. Neste trabalho, o codificador turbo estudado e implementado foi o proposto em [4], o qual é composto por dois codificadores convolucionais sistemáticos recursivos concatenados em paralelo com um entrelaçador entre eles.

A vantagem de se utilizar codificadores recursivos, reside no fato de que este tipo de codificador produz seqüências codificadas de alto peso, mesmo quando as seqüências de entrada possuem baixo peso, o que contribui para melhorar o desempenho do código. Detalhes sobre como os pesos das seqüências de entrada e codificadas influenciam no desempenho dos códigos turbo são vistos no Capítulo 5.

Os códigos turbo são códigos de bloco cujo tamanho do bloco de entrada é igual ao tamanho do entrelaçador utilizado, enquanto que o tamanho das palavras códigos, dependerá obviamente da taxa do código.

O entrelaçador utilizado desempenha duas funções na codificação turbo: a primeira é descorrelacionar as entradas dos decodificadores dos

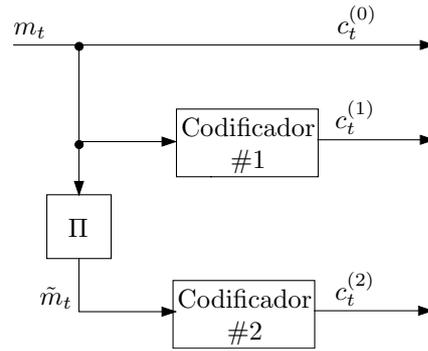


Figura 2.9: Codificador turbo genérico.

códigos componentes, de modo a possibilitar a utilização de um algoritmo de decodificação iterativo, baseado na troca de informação “descorrelatada” entre os decodificadores; e a segunda consiste em melhorar perfil de peso das palavras códigos, como será visto no Capítulo 5.

No codificador turbo, o primeiro codificador é responsável por codificar a seqüência de entrada ou mensagem \mathbf{m} , enquanto que o segundo codificador é responsável por codificar uma versão permutada da mesma, $\tilde{\mathbf{m}}$, conforme Fig. 2.9. Ambos os codificadores convolucionais têm parâmetros $(2, 1, K)$ e possuem função de transferência¹¹ dada por

$$\mathbf{G}(x) = \begin{pmatrix} 1 & \frac{h_0(x)}{h_1(x)} \end{pmatrix}. \quad (2-39)$$

O codificador turbo com a estrutura ilustrada na Fig. 2.9, é representado neste trabalho pela notação (h_0, h_1, N) , onde h_0 e h_1 são as representações em octal¹² de $h_0(x)$ e $h_1(x)$, respectivamente, e N é o tamanho do entrelaçador empregado. Um código turbo $\mathcal{C} = (h_0, h_1, N)$ é definido como o conjunto de todas as seqüências de saída produzidas por um codificador turbo (h_0, h_1, N) .

Na saída do codificador turbo a seqüência de saída \mathbf{c} é composta por:

- uma cópia da mensagem, $\mathbf{c}^{(0)} = \mathbf{m}$;
- a seqüência de paridade $\mathbf{c}^{(1)}$ gerada pelo primeiro codificador;
- a seqüência de paridade $\mathbf{c}^{(2)}$ gerada pelo segundo codificador.

¹¹A matriz $\mathbf{G}(x)$ não precisa necessariamente ser a mesma para os dois codificadores empregados, entretanto, não foi justificada na literatura nenhuma razão para que se utilizasse codificadores com matrizes geradoras diferentes.

¹²A representação em octal para um polinômio binário $f(x) = b_0 + b_1x + \dots + b_nx^n$ de grau n é obtida primeiro formando-se o vetor binário (b_0, b_1, \dots, b_n) e em seguida particionando-se este vetor em trios da direita para a esquerda, isto é, a_i corresponde ao valor em octal para o trio $(b_{n-3i-2}, b_{n-3i-1}, b_{n-3i})$, $i = 0, \dots, z$, onde $z = \lceil n/3 \rceil - 1$ e $b_i = 0$ para $i < 0$; a representação em octal é então dada pela concatenação $a_z \dots a_0$.

A taxa deste codificador turbo é $R = \frac{1}{3}$, sendo obviamente possível construir codificadores com taxas maiores através do uso de punção das seqüências de saída. Note que a seqüência $\tilde{\mathbf{m}}$ não é transmitida, pois é redundante em relação a \mathbf{m} .

A seqüência de saída \mathbf{c} é dada por

$$\begin{aligned} \mathbf{c} &= \begin{pmatrix} c_0^{(0)} & c_1^{(0)} & \cdots & c_{N-1}^{(0)} \\ c_0^{(1)} & c_1^{(1)} & \cdots & c_{N-1}^{(1)} \\ c_0^{(2)} & c_1^{(2)} & \cdots & c_{N-1}^{(2)} \end{pmatrix} \\ &= \left(\mathbf{c}_0 \quad \mathbf{c}_1 \quad \cdots \quad \mathbf{c}_{N-1} \right). \end{aligned} \quad (2-40)$$

Considere o codificador turbo ilustrado na Fig. 2.10 [31]. A matriz geradora de cada componente de codificação é dada por

$$\mathbf{G}(x) = \left(1 \quad \frac{1}{1+x^2} \right) \quad (2-41)$$

e o vetor de entrelaçamento de Π dado por $\boldsymbol{\pi} = (8 \ 3 \ 7 \ 6 \ 9 \ 0 \ 2 \ 5 \ 1 \ 4)$, para o bloco de entrada $\mathbf{m} = (1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1)$ tem-se que

$$\begin{aligned} \mathbf{c}^{(0)} &= \left(1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \right) = \mathbf{m} \\ \mathbf{c}^{(1)} &= \left(1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \right) \\ \tilde{\mathbf{m}} &= \left(1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \right) \\ \mathbf{c}^{(2)} &= \left(1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \right). \end{aligned} \quad (2-42)$$

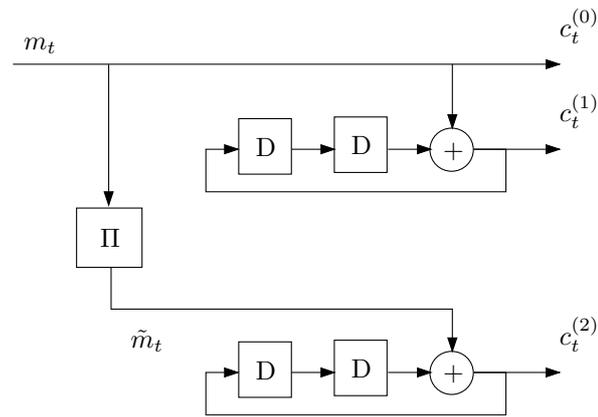


Figura 2.10: Codificador turbo.

Nesse capítulo foram revisados os principais conceitos e características relacionados aos componentes que compõem um codificador turbo, assim como seu funcionamento. No próximo capítulo, é abordado o algoritmo de decodificação turbo proposto em [4], segundo uma abordagem que denominamos neste trabalho de abordagem convencional.