

3 Decodificação Turbo

A decodificação turbo consiste em um método de decodificação subótimo, mas que no entanto apresenta um desempenho próximo ao limite proposto por Shannon. Do ponto de vista da decodificação, este excelente desempenho é resultado da combinação de dois fatores:

- 1) A decodificação turbo é realizada de modo iterativo;
- 2) Os decodificadores dos códigos componentes são do tipo SISO (*Soft Input Soft Output*).

A decodificação turbo iterativa consiste em dois decodificadores SISO concatenados serialmente através de entrelaçadores, conforme ilustrado na Fig. 3.1. Um decodificador, ao longo das iterações, produz em sua saída valores reais (*soft*), denominados de *informação extrínseca*, que são repassados à entrada do outro decodificador. Ao receber a informação extrínseca, um decodificador está recebendo a inferência que o outro decodificador fez sobre os bits transmitidos, o que contribui positivamente para sua própria inferência, aprimorando a cada nova iteração a estimativa dos bits transmitidos.

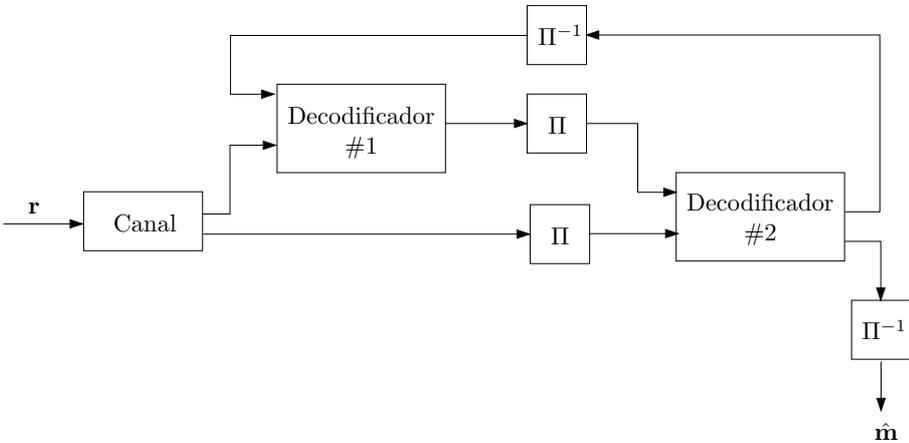


Figura 3.1: Diagrama de blocos simplificado de um decodificador turbo.

Atingido o número pré-definido de iterações, o segundo decodificador entrega em sua saída uma decisão *hard* dos bits transmitidos, finalizando assim a decodificação turbo.

Para os decodificadores SISO foi proposto em [4] o uso do algoritmo BCJR ¹. Este algoritmo é usualmente conhecido, em artigos e livros que abordam decodificação turbo, por algoritmo MAP (*Maximum a Posteriori*).

Para a decodificação turbo, foram propostas outras variações do algoritmo MAP no domínio logarítmico [7], assim como uma versão subótima do algoritmo de Viterbi conhecida como algoritmo SOVA (*Soft Output Viterbi Algorithm*) [26]. Dentre os vários algoritmos citados, optou-se, neste trabalho, por abordar apenas o algoritmo MAP e suas variações.

Neste capítulo, o algoritmo MAP e a decodificação turbo iterativa são desenvolvidos através da abordagem convencional, amplamente utilizada na literatura, na qual os algoritmos de decodificação são desenvolvidos baseados na treliça dos códigos componentes utilizados. As versões do algoritmo MAP no domínio logarítmico são abordados no Capítulo 4.

Este capítulo está organizado da seguinte forma. Na Seção 3.1 é apresentado o desenvolvimento do algoritmo MAP para codificadores convolucionais genéricos. Na Seção 3.1.1, este algoritmo é adaptado para o caso específico em que o codificador convolucional é sistemático recursivo. A decodificação turbo iterativa é abordada na Seção 3.2.

3.1 Algoritmo MAP(BCJR)

O algoritmo MAP é apropriado para a estimação dos estados ou saídas de um processo de Markov discreto no tempo com número de estados finitos. Como a saída de codificadores convolucionais ao passarem por canais discretos (tais como aqueles derivados de um canal AWGN) correspondem a uma seqüência de Markov, o algoritmo MAP pode então ser utilizado em sua decodificação.

Considerando o diagrama de blocos da Fig. 1.2, o algoritmo MAP pode ser utilizado na construção de um decodificador que minimiza a probabilidade de erro de símbolo², e disponibiliza uma estimação dos bits transmitidos m_t (*hard output*) e as probabilidades *a posteriori* dos mesmos (*soft output*), ambos baseados na seqüência recebida \mathbf{r} .

¹BCJR corresponde à sigla dos nomes de seus criadores: Bahl, Cocke, Jelinek e Raviv.

²Probabilidade de bit para comunicação binária.

Por simplicidade, considere um codificador convolucional $(n, 1, K)$ binário, de uma entrada, com *constraint length* K e taxa $R = \frac{1}{n}$.

Assume-se que a seqüência de entrada \mathbf{m}

$$\mathbf{m} = \left(m_0 \quad \cdots \quad m_t \quad \cdots \quad m_{N-1} \right) \quad (3-1)$$

é formada por símbolos equiprováveis e independentes entre si, sendo m_t o símbolo da seqüência de entrada no instante t .

A seqüência codificada é representada por

$$\mathbf{c} = \left(\mathbf{c}_0 \quad \cdots \quad \mathbf{c}_t \quad \cdots \quad \mathbf{c}_{N-1} \right) \quad (3-2)$$

onde

$$\mathbf{c}_t = \begin{pmatrix} c_t^{(0)} \\ \vdots \\ c_t^{(n-1)} \end{pmatrix}. \quad (3-3)$$

A seqüência de estados descrita pelos registradores do codificador, entre os instantes de 0 a N , é

$$\boldsymbol{\sigma} = \left(\sigma_0 \quad \cdots \quad \sigma_N \right) \quad (3-4)$$

onde σ_t pode assumir valores pertencentes ao conjunto $\{0, 1, \dots, Q-1\}$, com $Q = 2^{K-1}$ é o número total de estados.

Como foi abordado na Seção 2.1.2, uma transição de estado, ou seja, um ramo da treliça do código é representado pela quadrúpla $(\sigma_t, m_t, \mathbf{c}_t, \sigma_{t+1})$, conforme ilustrado na Fig. 3.2.

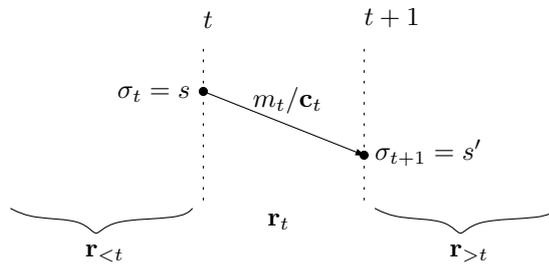


Figura 3.2: Transição de estado $(\sigma_t = s, \sigma_{t+1} = s')$ na treliça do código.

Considera-se que o estado inicial do codificador $\sigma_0 = 0$, enquanto que o estado final σ_N pode ser assumido como desconhecido. Na prática, força-se os registradores do codificador a retornar ao estado zero no instante $t = N$, ou seja, $\sigma_N = 0$. Quando isso acontece diz-se que o codificador foi terminado. A terminação do codificador é realizada impondo-se valores aos ν bits do

final de \mathbf{m} , de modo que σ_N seja zero, a esses bits dá-se a denominação de bits de cauda (“*tail bits*”).

No caso de um canal AWGN com modulador BPSK, cada símbolo \mathbf{x}_t é representado por

$$\mathbf{x}_t = \begin{pmatrix} x_t^{(0)} \\ \vdots \\ x_t^{(n-1)} \end{pmatrix} \quad (3-5)$$

com

$$\begin{aligned} x_t^{(i)} &= \mathcal{M}(c_t^{(i)}) \\ &= \sqrt{E_c} \left(2c_t^{(i)} - 1 \right), \quad t = 0, \dots, N-1, \quad i = 0, \dots, n-1 \end{aligned} \quad (3-6)$$

onde $E_c = \frac{E_b}{R}$, a energia alocada a cada bit codificado c_t , é obtida a partir de E_b , a energia alocada a cada bit m_t de \mathbf{m} .

Tendo em vista que o canal utilizado é um canal gaussiano, na recepção, o usuário observa a seqüência recebida

$$\mathbf{r} = \left(\mathbf{r}_0 \quad \cdots \quad \mathbf{r}_t \quad \cdots \quad \mathbf{r}_{N-1} \right) \quad (3-7)$$

onde

$$\mathbf{r}_t = \mathbf{x}_t + \mathbf{n}_t. \quad (3-8)$$

As componentes do vetor gaussiano \mathbf{n}_t são variáveis aleatórias gaussianas independentes com função densidade de probabilidade $\mathcal{N} \left(0, \frac{N_0}{2} \right)$.

Considere que a seqüência recebida seja particionada em três componentes $\mathbf{r} = \mathbf{r}_{<t} \mathbf{r}_t \mathbf{r}_{>t}$, onde $\mathbf{r}_{<t} = \{\mathbf{r}_l : l < t\}$, é o conjunto das observações passadas, \mathbf{r}_t é a observação atual, e $\mathbf{r}_{>t} = \{\mathbf{r}_l : l > t\}$ refere-se às observações futuras.

O objetivo do decodificador é determinar as probabilidades *a posteriori* $P(m_t = m | \mathbf{r})$. O primeiro passo para o cálculo destas probabilidades é obter as probabilidades *a posteriori* em função das transição de estados, estando cada transição (σ_t, σ_{t+1}) associada de maneira única ao par (m_t, \mathbf{c}_t) , vide Def. 2.17, torna-se trivial a determinar as probabilidades $P(m_t = m | \mathbf{r})$.

A probabilidade *a posteriori* da transição de estado $(\sigma_t = s, \sigma_{t+1} = s')$, vide Fig. 3.2, dada a seqüência observada \mathbf{r} , tendo em vista que o canal é sem memória (as variáveis $\mathbf{n}_{<t}, \mathbf{n}_t$ e $\mathbf{n}_{>t}$ são estatisticamente independentes),

e que \mathbf{r} é um processo de Markov, é

$$\begin{aligned} P(\sigma_t = s, \sigma_{t+1} = s' | \mathbf{r}) &= p(\sigma_t = s, \sigma_{t+1} = s', \mathbf{r}) / p(\mathbf{r}) & (3-9) \\ &= p(\sigma_t = s, \sigma_{t+1} = s', \mathbf{r}_{<t}, \mathbf{r}_t, \mathbf{r}_{>t}) / p(\mathbf{r}) \\ &= \frac{p(\sigma_t = s, \sigma_{t+1} = s', \mathbf{r}_{<t}, \mathbf{r}_t)}{p(\mathbf{r})} \times \\ &\quad p(\mathbf{r}_{>t} | \sigma_t = s, \sigma_{t+1} = s', \mathbf{r}_{<t}, \mathbf{r}_t). \end{aligned}$$

De acordo com a propriedade de Markov, o conhecimento do estado no instante $t + 1$ não informa nada relevante sobre o estado t ou estados anteriores, tem-se

$$p(\mathbf{r}_{>t} | \sigma_t = s, \sigma_{t+1} = s', \mathbf{r}_{<t}, \mathbf{r}_t) = p(\mathbf{r}_{>t} | \sigma_{t+1} = s') \quad (3-10)$$

e

$$\begin{aligned} p(\sigma_t = s, \sigma_{t+1} = s', \mathbf{r}_{<t}, \mathbf{r}_t) &= p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = s, \mathbf{r}_{<t}) p(\sigma_t = s, \mathbf{r}_{<t}) & (3-11) \\ &= p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = p) p(\sigma_t = s, \mathbf{r}_{<t}). \end{aligned}$$

Substituindo-se (3-10) e (3-11) em (3-9)

$$P(\sigma_t = s, \sigma_{t+1} = s' | \mathbf{r}) = \frac{p(\sigma_t = s, \mathbf{r}_{<t}) p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = p) p(\mathbf{r}_{>t} | \sigma_{t+1} = s')}{p(\mathbf{r})}. \quad (3-12)$$

Os fatores em (3-12) podem ser representados por $\alpha_t(s)$, $\gamma_t(s, s')$, $\beta_{t+1}(s')$. O fator

$$\alpha_t(s) = p(\sigma_t = s, \mathbf{r}_{<t}) \quad (3-13)$$

denominado de *métrica progressiva*, é proporcional à probabilidade do conjunto de percursos na treliça do código cuja seqüência de estados que os caracterizam, possuem o mesmo estado $\sigma_t = s$ no instante t , ou seja, percursos que finalizam no estado s no instante t . O fator

$$\gamma_t(s, s') = p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = s) \quad (3-14)$$

é proporcional à probabilidade da transição de estado ($\sigma_t = s, \sigma_{t+1} = s'$), considerando a observação \mathbf{r}_t , e por fim, o fator

$$\beta_{t+1}(s') = p(\mathbf{r}_{>t} | \sigma_{t+1} = s') \quad (3-15)$$

denominado de *métrica regressiva*, é proporcional à probabilidade do conjunto de percursos na treliça do código cuja seqüência de estados que os caracterizam, possuem o mesmo estado $\sigma_{t+1} = s'$ no instante $t + 1$, ou seja, percursos que iniciam no estado s' no instante $t + 1$.

Para obter as probabilidades $P(m_t = m|\mathbf{r})$ a partir da Eq. (3-12), basta apenas realizar o somatório de todas as probabilidades $P(\sigma_t = s, \sigma_{t+1} = s'|\mathbf{r})$ correspondentes às transições de estados ($\sigma_t = s, \sigma_{t+1} = s'$) associadas à entrada $m_t = m$.

Considere os conjuntos de transições de estados definidos por: $\mathcal{S}_0 \triangleq \{(s, s') : m_t = 0\}$ e $\mathcal{S}_1 \triangleq \{(s, s') : m_t = 1\}$, dessa forma

$$\begin{aligned} P(m_t = 0|\mathbf{r}) &= \sum_{(s,s') \in \mathcal{S}_0} P(\sigma_t = s, \sigma_{t+1} = s'|\mathbf{r}) & (3-16) \\ &= \frac{1}{p(\mathbf{r})} \sum_{(s,s') \in \mathcal{S}_0} \alpha_t(s) \gamma_t(s, s') \beta_{t+1}(s') \end{aligned}$$

e

$$\begin{aligned} P(m_t = 1|\mathbf{r}) &= \sum_{(s,s') \in \mathcal{S}_1} P(\sigma_t = s, \sigma_{t+1} = s'|\mathbf{r}) & (3-17) \\ &= \frac{1}{p(\mathbf{r})} \sum_{(s,s') \in \mathcal{S}_1} \alpha_t(s) \gamma_t(s, s') \beta_{t+1}(s'). \end{aligned}$$

Observe que $\frac{1}{p(\mathbf{r})}$ é um fator comum às equações (3-16) e (3-17) e não precisa ser explicitamente calculado. Considerando-o apenas como um fator de normalização, a probabilidade $P(m_t = m|\mathbf{r})$ pode ser reescrita como

$$P(m_t = m|\mathbf{r}) = \frac{\tilde{p}(m_t = m|\mathbf{r})}{\sum_m \tilde{p}(m_t = m|\mathbf{r})} \quad (3-18)$$

onde

$$\tilde{p}(m_t = m|\mathbf{r}) = \sum_{(s,s') \in \mathcal{S}_m} \alpha_t(s) \gamma_t(s, s') \beta_{t+1}(s'). \quad (3-19)$$

As expressões para $\alpha_t(s)$, $\gamma_t(s, s')$ e $\beta_{t+1}(s')$ são desenvolvidas abaixo. Os cálculos de $\alpha_t(s)$ $\beta_{t+1}(s')$ são efetuados recursivamente de maneira eficiente.

Cálculo de α

Dado $\alpha_t(s)$ para $s \in \{0, \dots, Q - 1\}$, $\alpha_{t+1}(s')$ é obtida por

$$\begin{aligned}
\alpha_{t+1}(s') &= p(\sigma_{t+1} = s', \mathbf{r}_{<t+1}) = p(\sigma_{t+1} = s', \mathbf{r}_t, \mathbf{r}_{<t}) & (3-20) \\
&= \sum_{s=0}^{Q-1} p(\sigma_{t+1} = s', \mathbf{r}_t, \sigma_t = s, \mathbf{r}_{<t}) \\
&= \sum_{s=0}^{Q-1} p(\sigma_t = s, \mathbf{r}_{<t}) p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = s, \mathbf{r}_{<t}) \\
&= \sum_{s=0}^{Q-1} p(\sigma_t = s, \mathbf{r}_{<t}) p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = s) \\
&= \sum_{s=0}^{Q-1} \alpha_t(s) \gamma_t(s, s')
\end{aligned}$$

O cálculo da métrica α é efetuado no sentido de t crescente da treliça, ou seja, do instante $t = 0$ até $t = N - 1$, o que justifica sua denominação de métrica progressiva. Para $t = 0$, considera-se que os registradores do codificador são inicializados no estado zero, portanto α é inicializada com os valores

$$\left(\alpha_0(0) \quad \alpha_0(1) \quad \cdots \quad \alpha_0(Q-1) \right) = \left(1 \quad 0 \quad \cdots \quad 0 \right). \quad (3-21)$$

Cálculo de β

Dado $\beta_{t+1}(s')$ para $s' \in \{0, \dots, Q-1\}$, $\beta_t(s)$ é obtida por

$$\begin{aligned}
\beta_t(s) &= p(\mathbf{r}_{>t-1} | \sigma_t = s) = p(\mathbf{r}_{>t}, \mathbf{r}_t | \sigma_t = s) & (3-22) \\
&= \sum_{s'=0}^{Q-1} p(\mathbf{r}_{>t}, \mathbf{r}_t, \sigma_{t+1} = s' | \sigma_t = s) \\
&= \sum_{s'=0}^{Q-1} p(\mathbf{r}_t, \sigma_{t+1} = s' | \sigma_t = s) p(\mathbf{r}_{>t} | \mathbf{r}_t, \sigma_{t+1} = s', \sigma_t = s) \\
&= \sum_{s'=0}^{Q-1} p(\mathbf{r}_t, \sigma_{t+1} = s' | \sigma_t = s) p(\mathbf{r}_{>t} | \sigma_{t+1} = s') \\
&= \sum_{s'=0}^{Q-1} \gamma_t(s, s') \beta_{t+1}(s').
\end{aligned}$$

O cálculo da métrica β é efetuado no sentido de t decrescente da treliça, do instante $t = N - 1$ até $t = 0$, o que justifica sua denominação de

métrica reversa. A métrica β por ser calculada no sentido de t decrescente é inicializada em $t = N$ com os valores

$$\left(\beta_N(0) \quad \beta_N(1) \quad \cdots \quad \beta_N(Q-1) \right) = \left(1 \quad 0 \quad \cdots \quad 0 \right) \quad (3-23)$$

quando o codificador é terminado no estado zero, e

$$\left(\beta_N(0) \quad \beta_N(1) \quad \cdots \quad \beta_N(Q-1) \right) = \left(\frac{1}{Q} \quad \frac{1}{Q} \quad \cdots \quad \frac{1}{Q} \right). \quad (3-24)$$

quando considera-se que o codificador pode ser terminado em qualquer estado, desconhecido *a priori*, com distribuição uniforme.

Cálculo de γ

Como foi definido em (3-14)

$$\gamma_t(s, s') = p(\sigma_{t+1} = s', \mathbf{r}_t | \sigma_t = s)$$

utilizando o teorema de Bayes,

$$\begin{aligned} \gamma_t(s, s') &= \frac{p(\sigma_{t+1} = s', \mathbf{r}_t, \sigma_t = s)}{P(\sigma_t = s)} & (3-26) \\ &= p(\mathbf{r}_t | \sigma_t = s, \sigma_{t+1} = s') \frac{P(\sigma_t = s, \sigma_{t+1} = s')}{P(\sigma_t = s)} \\ &= p(\mathbf{r}_t | \sigma_t = s, \sigma_{t+1} = s') P(\sigma_{t+1} = s' | \sigma_t = s). \end{aligned}$$

Considerando que a transição $(\sigma_t = s, \sigma_{t+1} = s')$ é provocada pela entrada m_t e está associada a quádrupla $(\sigma_t, m_t, \mathbf{x}_t, \sigma_{t+1})$, onde \mathbf{c}_t foi substituído por \mathbf{x}_t , visto que ambos estão relacionados diretamente pelo mapeamento $\mathcal{M}(\mathbf{c}_t) = \mathbf{x}_t$, tem-se

$$\gamma_t(s, s') = p(\mathbf{r}_t | \mathbf{x}_t) P(m_t = m). \quad (3-27)$$

Para o caso em que o canal é sem memória, AWGN, o modulador é BPSK, e o código possui $R = \frac{1}{2}$, obtém-se

$$\begin{aligned} p(\mathbf{r}_t | \mathbf{x}_t) &= p(r_t^{(0)} | x_t^{(0)}) \cdot p(r_t^{(1)} | x_t^{(1)}) & (3-28) \\ &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(r_t^{(0)} - x_t^{(0)})^2\right) \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-1}{2\sigma^2}(r_t^{(1)} - x_t^{(1)})^2\right). \end{aligned}$$

3.1.1

Algoritmo MAP para Codificadores Sistemáticos

Os códigos componentes utilizados no codificador turbo proposto em [4], consistem de dois codificadores convolucionais recursivos sistemáticos com $R = \frac{1}{2}$. Nesta seção o algoritmo MAP desenvolvido na Seção 3.1 é adaptado para a decodificação referente a este codificador.

Para este caso, considerando um canal sem memória, AWGN com modulador BPSK, tem-se

$$\mathbf{c}_t = \begin{pmatrix} c_t^{(0)} \\ c_t^{(1)} \end{pmatrix} = \begin{pmatrix} m_t \\ c_t^{(1)} \end{pmatrix} \quad (3-29)$$

e o vetor de entrada no canal é

$$\mathbf{x}_t = \begin{pmatrix} x_t^{(0)} \\ x_t^{(1)} \end{pmatrix} = \sqrt{E_c} \cdot \begin{pmatrix} 2m_t - 1 \\ 2c_t^{(1)} - 1 \end{pmatrix} \quad (3-30)$$

O vetor recebido é representado por

$$\mathbf{r}_t = \begin{pmatrix} r_t^{(0)} \\ r_t^{(1)} \end{pmatrix} \quad (3-31)$$

onde

$$\begin{aligned} r_t^{(0)} &= x_t^{(0)} + n_t^{(0)} \\ r_t^{(1)} &= x_t^{(1)} + n_t^{(1)} \end{aligned} \quad (3-32)$$

com $n_t^{(0)}$ e $n_t^{(1)}$ sendo variáveis aleatórias gaussianas com função densidade de probabilidade $\mathcal{N}\left(0, \frac{N_0}{2}\right)$.

A Eq. (3-27) é reescrita como

$$\gamma_t(s, s') = p(r_t^{(0)} | x_t^{(0)})p(r_t^{(1)} | x_t^{(1)})P(m_t = m). \quad (3-33)$$

Substituindo (3-33) em (3-18), tem-se

$$P(m_t = m | \mathbf{r}) = N_m \sum_{(s, s') \in \mathcal{S}_m} \alpha_t(s) p(r_t^{(0)} | x_t^{(0)}) p(r_t^{(1)} | x_t^{(1)}) P(m_t = m) \beta_{t+1}(s'). \quad (3-34)$$

Como $p(r_t^{(0)} | x_t^{(0)})$ e $P(m_t = m)$ independem da estrutura do codifica-

dor,

$$\begin{aligned}
 P(m_t = m | \mathbf{r}) &= N_m p(r_t^{(0)} | x_t^{(0)}) P(m_t = m) \sum_{(s,s') \in \mathcal{S}_m} \alpha_t(s) p(r_t^{(1)} | x_t^{(1)}) \beta_{t+1}(s') \\
 &= N_m \cdot P_{\text{int},t}(m) \cdot P_{\text{pri},t}(m) \cdot P_{\text{ext},t}(m)
 \end{aligned} \tag{3-35}$$

onde a notação

$$P_{\text{int},t}(m) = p(r_t^{(0)} | x_t^{(0)}) \tag{3-36}$$

designa uma função densidade de probabilidade denominada de *probabilidade intrínseca*, que carrega informação sobre o vetor recebido $\mathbf{r}^{(0)}$ dado o vetor codificado $\mathbf{c}^{(0)}$. A notação

$$P_{\text{pri},t}(m) = P(m_t = m) \tag{3-37}$$

designa a *probabilidade a priori* dos bits transmitidos, representando a informação sobre esses bits (sem condicionamento no vetor \mathbf{r}). A notação

$$P_{\text{ext},t}(m) = \sum_{(s,s') \in \mathcal{S}_m} \alpha_t(s) p(r_t^{(1)} | x_t^{(1)}) \beta_{t+1}(s') \tag{3-38}$$

designa uma função densidade de probabilidade denominada de *probabilidade extrínseca*. Esta probabilidade corresponde à informação produzida pelo decodificador em relação aos bits transmitidos.

3.2

Decodificação Iterativa

A decodificação turbo, como dito anteriormente, ocorre de maneira iterativa. O decodificador turbo, como pode ser visualizado na Fig. 3.3, é composto por dois decodificadores MAP que empregam o algoritmo MAP na decodificação de cada componente de codificação. O primeiro decodificador utiliza como entradas as seqüências recebidas $(\mathbf{r}^{(0)}, \mathbf{r}^{(1)})$, enquanto que o segundo decodificador utiliza as seqüências $(\Pi[\mathbf{r}^{(0)}], \mathbf{r}^{(2)})$.

Cada iteração l é composta por duas fases, cada fase correspondendo à decodificação de um código componente. Durante a decodificação iterativa, a informação trocada entre os decodificadores é a informação extrínseca que cada decodificador produz em sua saída. A informação extrínseca de um decodificador é utilizada como se fosse uma probabilidade *a priori* pelo

outro, de modo que a cada nova decodificação, a probabilidade *a priori* utilizada, fornece informações mais precisas sobre a seqüência transmitida \mathbf{m} .

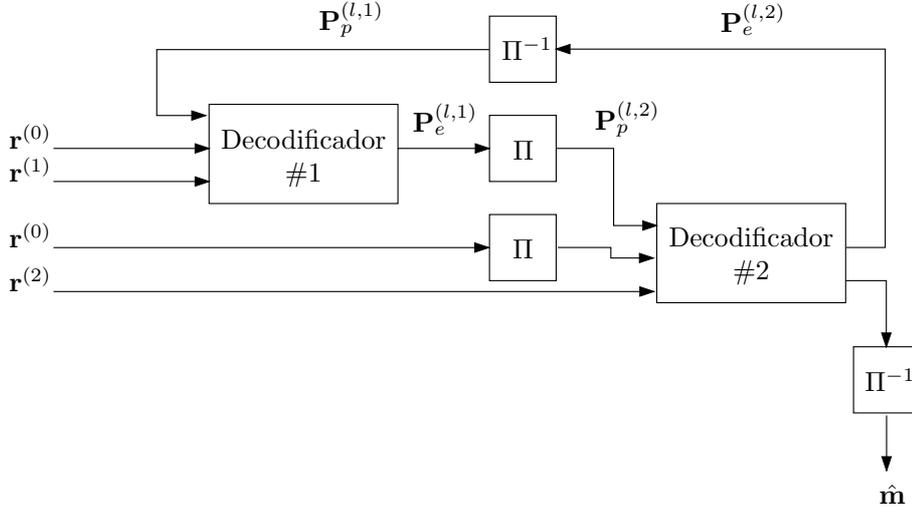


Figura 3.3: Diagrama de blocos de um decodificador turbo.

Considere

$$\mathbf{P}_e^{(l,j)} = \begin{pmatrix} P_{\text{ext},0}^{(l,j)}(m_0 = 0) \cdots P_{\text{ext},t}^{(l,j)}(m_t = 0) \cdots P_{\text{ext},N-1}^{(l,j)}(m_{N-1} = 0) \\ P_{\text{ext},0}^{(l,j)}(m_0 = 1) \cdots P_{\text{ext},t}^{(l,j)}(m_t = 1) \cdots P_{\text{ext},N-1}^{(l,j)}(m_{N-1} = 1) \end{pmatrix} \quad (3-39)$$

como a matriz com as informações extrínsecas sobre os bits transmitidos na iteração l , ($l = 1, \dots, L$), obtido na saída do decodificador j , ($j = 1, 2$), e

$$\mathbf{P}_p^{(l,j)} = \begin{pmatrix} P_{\text{pri},0}^{(l,j)}(m_0 = 0) \cdots P_{\text{pri},N-1}^{(l,j)}(m_{N-1} = 0) \\ P_{\text{pri},0}^{(l,j)}(m_0 = 1) \cdots P_{\text{pri},N-1}^{(l,j)}(m_{N-1} = 1) \end{pmatrix} \quad (3-40)$$

como a matriz das probabilidades *a priori* dos bits transmitidos, utilizada pelo decodificador j na iteração l .

Ao iniciar o processo iterativo, $l = 0$, assume-se que os bits transmitidos são equiprováveis, ou seja, para o decodificador 1

$$P_{\text{pri},t}^{(0,1)}(m_t = 0) = P_{\text{pri},t}^{(0,1)}(m_t = 1) = P(m_t = m) = 1/2, \quad t = 0, \dots, N - 1. \quad (3-41)$$

O algoritmo de decodificação turbo proposto em [4] é descrito abaixo:

Algoritmo 3.1 (Decodificação Turbo)

- 1) Faça $P_{\text{pri},t}^{(0,1)}(m_t = m) = P(m_t = m)$, para $m = 0, 1$ e $t = 0, \dots, N - 1$;
- 2) Para $l = 1, \dots, L$:

- a) Usando os elementos de $\mathbf{P}_p^{(l-1,1)}$ como probabilidades *a priori* $P(m_t = m)$, $\forall t$, calcule:
- α , β e γ de acordo com Eq. (3-20), Eq. (3-22) e Eq. (3-27);
 - $P_{\text{ext},t}^{(l,1)}$ de acordo com Eq. (3-38);
- b) Faça³ $\mathbf{P}_p^{(l,2)} = \Pi[\mathbf{P}_e^{(l,1)}]$;
- c) Usando os elementos de $\mathbf{P}_p^{(l,2)}$ como probabilidades *a priori* $P(m_t = m)$, $\forall t$, calcule:
- α , β e γ de acordo com Eq. (3-20), Eq. (3-22) e Eq. (3-27);
- d) Se $l \neq L$:
- Calcule $P_{\text{ext},t}^{(l,2)}$, $\forall t$, utilizando Eq. (3-38);
 - Faça $\mathbf{P}_p^{(l+1,1)} = \Pi^{-1}[\mathbf{P}_e^{(l,2)}]$;
- e) Se $l = L$:
- Usando os elementos de $\mathbf{P}_p^{(l,2)}$ como probabilidades *a priori* $P(m_t = m)$, calcule para $t = 0, \dots, N - 1$, $P(m_t = m|\mathbf{r})$ de acordo com Eq. (3-35);
- 3) Decida para $t = 0, \dots, N - 1$:
- Se $P(\tilde{m}_t = 0|\mathbf{r}) > P(\tilde{m}_t = 1|\mathbf{r})$, $\hat{m}_t = 0$;
 - Senão $\hat{m}_t = 1$;
- 4) Faça $\hat{\mathbf{m}} = \Pi^{-1}[\hat{\tilde{\mathbf{m}}}]$;
- 5) Fim. ■

Neste capítulo a decodificação turbo foi desenvolvida através da abordagem convencional. No próximo capítulo, esta é desenvolvida segundo a teoria de grafos fatores proposta em [25].

³As notações $\Pi[\cdot]$ e $\Pi^{-1}[\cdot]$ foram utilizadas para indicar a operação de entrelaçamento e desentrelaçamento, respectivamente.