

Decodificação Turbo Integrada a Codificação de Rede na Camada Física

Maria Cláudia Castro^{1,2}, Bartolomeu F. Uchôa-Filho¹, Tiago T. V. Vinhoza³, Mário Noronha-Neto² e João Barros³

Resumo—Consideramos um sistema com dois usuários que desejam trocar mensagens entre si através de uma rede *Two-Way Relay*. Os usuários codificam suas mensagens através do mesmo código turbo e o *relay* emprega codificação de rede na camada física (PNC). Este trabalho visa à recuperação do XOR das mensagens enviadas a partir do sinal recebido no *relay*, que consiste na sobreposição dos sinais codificados, desvanecidos e acrescidos de ruído. Propomos um esquema de decodificação turbo no *relay* que opera numa treliça produto e decodifica o XOR das mensagens transmitidas. O esquema proposto incorpora uma melhoria que pode ser vista como uma forma de se considerar estimativas suaves do XOR das mensagens, adiando, o maior tempo possível, uma decisão abrupta. Os resultados revelam um ganho de codificação da ordem de 1.5 dB em um modelo de canal com desvanecimento Rayleigh.

Palavras-Chave—Codificação de rede na camada física, Codificação turbo, Canal Rayleigh.

Abstract—We consider a two-way relay network, in which two users want to exchange messages through a relay. The users encode their messages using the same turbo code and the relay employ physical-layer network coding (PNC). Our goal is to recover the XOR of the sent messages based on the received signal at the relay, which is the superposition of the coded signals distorted by fading and corrupted by additive noise. We propose a turbo decoding scheme that operates on a product trellis. The proposed scheme introduces an improvement to which we can trace the parallel of working with soft estimates of the XOR for as long as possible. Results show that the proposed decoding scheme outperforms its simplified version with coding gains of up to 1.5 dB in a Rayleigh fading channel model.

Keywords—Physical-layer network coding, Turbo codes, Rayleigh fading channel.

I. INTRODUÇÃO

A técnica de Codificação de Rede tem atraído cada vez mais a atenção da comunidade científica, particularmente para os ambientes sem fio. A ideia principal foi primeiramente proposta por Ahlswede et al. [1], que mostraram que, se for permitido aos nós intermediários de uma rede combinar os pacotes recebidos, antes de encaminhá-los, o fluxo de informação máxima poderá ser alcançado. A codificação de rede foi proposta pela primeira vez para operar nas camadas superiores da pilha protocolar e, portanto, preocupava-se com o problema de codificação de dados sobre canais com uma certa capacidade, ignorando a natureza física subjacente dos canais de comunicações. Mais recentemente, os princípios da codificação da rede têm sido aplicados na camada física,

explorando a natural superposição de ondas eletromagnéticas que ocorre nas comunicações sem fio. Esta superposição é geralmente considerada como um obstáculo quando a recuperação dos sinais individuais é necessária. No entanto, usando uma estratégia de codificação desenvolvida por Zhang et al. [2], denominada Codificação de Rede na Camada Física (PNC-Physical-Layer Network Coding), essa mistura de ondas é uma operação de codificação de rede que ocorre naturalmente e deve ser aproveitada para permitir uma comunicação mais eficiente.

A configuração de rede normalmente adotada como objeto de estudo em PNC é a rede de retransmissão de duas vias (TWRN- Two-Way Relay Network), estudado por Shannon [3]. Neste modelo, dois usuários comunicam-se através de um nó intermediário (ou *relay*) para trocar mensagens, não havendo ligação direta entre eles. Com PNC, os dois usuários primeiramente transmitem seus fluxos de dados binários independentes e simultaneamente ao *relay*, que recebe os sinais sobrepostos. O *relay*, em seguida, extrai do sinal recebido a soma módulo-2 (XOR) dos dois fluxos de dados, sem a necessidade de obtê-los individualmente de forma explícita. O resultado da operação XOR é então transmitido pelo *relay*, permitindo que cada usuário, após a subtração da parcela do sinal que ele conhece, possa recuperar o fluxo de dados do outro usuário. Todo o processo requer apenas dois *times slots*.

Apesar de usar a interferência de uma forma benéfica, ainda é preciso lidar com o ruído do canal. A fim de garantir uma transmissão confiável, o uso de codificação de canal é uma técnica importante. Nesse contexto, uma questão interessante que se coloca é como a codificação de canal pode ser eficientemente integrada ao sistema PNC.

Segundo Zhang e Liew [4], há duas maneiras para aplicar codificação de canal em PNC. Uma delas é o PNC codificado fim-a-fim, no qual o nó intermediário não realiza qualquer codificação ou decodificação de canal. A outra forma é chamada PNC codificado *link-a-link*, em que não apenas os dois usuários, mas também o *relay* executa codificação/decodificação de canal. A operação crucial é o assim chamado processo de codificação de rede e decodificação de canal (CNC- Channel-Decoding-Network-Coding) [4], e acontece no *relay*, sendo caracterizada pela recuperação da mensagem XOR a partir da sobreposição dos sinais codificados e corrompidos. Neste trabalho, adotamos a segunda abordagem.

Os códigos de canal propostos para integração ao PNC em [4] são os chamados códigos *repete e acumula* (RA - *Repeat-Accumulate*), que podem ser vistos simultaneamente como uma classe de códigos *turbo-like* e uma classe de códigos

¹Universidade Federal de Santa Catarina, Florianópolis, Brasil, ²Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Florianópolis, Brasil, ³Instituto de Telecomunicações, Faculdade de Engenharia da Universidade do Porto, Portugal. Email: {claudiacastro,noronha}@ifsc.edu.br, {uchoa,tiago.vinhoza}@ieee.org,jbarros@fe.up.pt.

LDPC (Low-Density Parity-Check). Essas duas classes de códigos correspondem às mais importantes em termos de desempenho em relação ao limite estabelecido por Shannon. Os resultados em [4], porém, são restritos ao ruído aditivo Gaussiano branco (AWGN - Additive White Gaussian Noise), ou seja, os ganhos de canal para todos os *links* foram assumidos constantes em todos os momentos. No presente trabalho, estendemos este cenário ao caso de desvanecimento Rayleigh e focamos o trabalho em código turbo e na integração do PNC no algoritmo de decodificação BCJR.

Recentemente, em [5], [6], os autores usaram códigos turbo integrados ao PNC considerando ambos os canais AWGN e com desvanecimento Rayleigh. É relatado que, em seu processo de decodificação turbo, as duas mensagens de dados (associadas aos dois usuários) são recuperadas individualmente e, em seguida a operação XOR é realizada. Para reduzir a complexidade de decodificação turbo, eles se concentraram na construção de uma treliça com um número reduzido de estados, com base na qual o decodificador turbo é supostamente capaz de recuperar a informação XOR diretamente a partir dos sinais sobrepostos. Uma proposta de códigos LDPC integrados ao PNC sobre canais sujeitos a desvanecimento foi apresentada em [7]. Um algoritmo soma-produto generalizado operando no corpo de Galois \mathbb{F}_4 foi desenvolvido, e simulações comprovaram um bom desempenho. Contrastando com [5], [6], neste artigo propomos a utilização conjunta de PNC e um decodificador turbo com base em uma treliça completa, onde o *relay* calcula o XOR das mensagens enviadas pelas fontes diretamente a partir dos sinais sobrepostos. Inspirados na ideia de [7], propomos neste artigo a decodificação que denotamos por XOR- \mathbb{F}_4 , onde interpretamos de forma distinta a informação que sai dos quatro ramos da treliça produto numa transição de estado. Mais adiante, veremos que na decodificação denotada por XOR- \mathbb{F}_2 , analisamos da mesma forma a informação de um par de ramos da treliça produto, o que pode ser interpretado como a aplicação de uma decisão suave na estimativa do XOR.

Este artigo é organizado da seguinte maneira. Na Seção II, descrevemos o modelo completo do sistema, incluindo os modelos estatísticos dos canais e os codificadores. Na Seção III, o decodificador turbo proposto para os bits XOR é apresentado. Destacamos a diferença entre as decodificações XOR- \mathbb{F}_2 e - \mathbb{F}_4 . Na Seção IV, apresentamos os resultados de simulações. Finalmente, na Seção V, apresentamos as nossas conclusões.

II. MODELO DO SISTEMA

Ao longo do artigo, adotamos a seguinte notação. Os vetores/matrices são representados por letras minúsculas em negrito, com exceção de \mathbf{I}_k , que representa a matriz identidade de dimensão $k \times k$, \oplus representa a operação XOR, e \circ representa o produto Hadamard (*component-wise*). Utilizamos $\tilde{\mathbf{x}}$ para denotar uma versão entrelaçada do vetor \mathbf{x} . Consideramos uma TWRN, onde dois nós fonte, S_1 e S_2 , desejam se comunicar através de um nó *relay*, R . Não existe *link* direto entre S_1 e S_2 . Quando PNC é empregada no nó intermediário, a troca de mensagens entre S_1 e S_2 é realizada em dois *times slots* distintos, denominados de fase de acesso múltiplo (MA

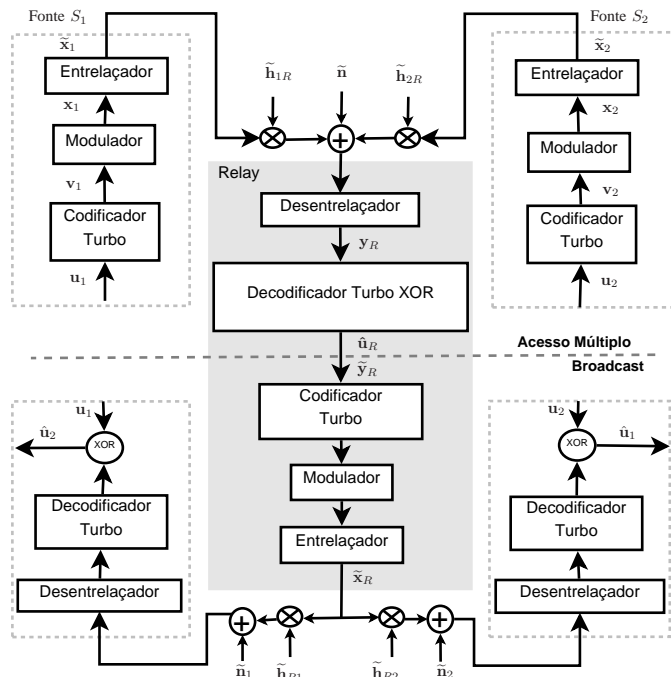


Fig. 1. Modelo do Sistema.

- Multiple Access) e fase de difusão (BC- Broadcast). Na fase MA, S_1 e S_2 transmitem seus sinais codificados por um codificador de canal para o nó *relay* e, devido à restrição *half-duplex*, as fontes não podem ouvir qualquer informação. O nó *relay* recebe os sinais de S_1 e S_2 sobrepostos e corrompidos, e estima a operação XOR das mensagens originais. Deve ser enfatizado que nenhum conhecimento individual das mensagens é necessário para este processo. Depois, na fase de BC, o nó *relay* codifica a mensagem XOR estimada e, aproveitando a natureza de difusão do meio sem fio, transmite o sinal codificado de volta para S_1 e S_2 . O modelo do sistema é mostrado na Figura 1.

Assumimos, neste trabalho, que todos os nós empregam codificação turbo [8] para codificar suas mensagens. O codificador turbo é constituído por dois códigos convolucionais recursivos e sistemáticos (RSC - Recursive Systematic Convolutional) idênticos, operando em paralelo com um entrelaçador entre eles. Para simplificar a exposição, assumimos o código turbo com uma taxa igual a $1/n$. Para $i = 1, 2$, seja $\mathbf{u}_i = [u_i(1), \dots, u_i(N)]$ o vetor de mensagem de tamanho N contendo os bits de informação da fonte S_i e $\mathbf{v}_i = [v_i(1), \dots, v_i(N)]$, a sequência codificada correspondente de tamanho nN bits. Cada bloco codificado $\mathbf{v}_i(t) = [v_i^1(t), \dots, v_i^n(t)]$ é um vetor de n bits e corresponde à saída de um codificador turbo considerando a entrada binária $u_i(t)$ no tempo t . Assumimos um codificador sistemático, de tal forma que $v_i^1(t) = u_i(t)$, para todo t . O bloco codificado é então modulado via BPSK para produzir \mathbf{x}_i , através da operação $x_i^j(t) = 1 - 2v_i^j(t)$ para $j = 1, \dots, n$. Os canais entre os nós fonte e intermediário são modelados como canais AWGN com desvanecimento Rayleigh correlacionados no tempo, o que pode levar a erros em rajada (*burst*). Para evitar este problema, empregamos um entrelaçador externo antes de

transmitir o sinal modulado. Cada nó fonte então transmite \tilde{x}_i .

O canal de desvanecimento de S_i para R é caracterizado pelo vetor de coeficientes de canal, $\tilde{\mathbf{h}}_{iR} = [\tilde{h}_{iR}(1), \dots, \tilde{h}_{iR}(N)]$, onde $\tilde{h}_{iR}(t) = [\tilde{h}_{iR}^1(t), \dots, \tilde{h}_{iR}^n(t)]$. O sinal recebido pelo nó *relay* é dado por $\tilde{\mathbf{y}}_R = \tilde{\mathbf{z}}_R + \tilde{\mathbf{n}}$, em que $\tilde{\mathbf{z}}_R = [\tilde{z}_R(1), \dots, \tilde{z}_R(N)]$ contém a superposição dos sinais desvanecidos: $\tilde{z}_R(t) = \tilde{x}_1(t) \circ \tilde{\mathbf{h}}_{1R}(t) + \tilde{x}_2(t) \circ \tilde{\mathbf{h}}_{2R}(t)$, e $\tilde{\mathbf{n}} = [\tilde{n}(1), \dots, \tilde{n}(N)]$ é o vetor ruído Gaussiano de valor complexo, com média zero e matriz de covariância $\sigma^2 \mathbf{I}_{nN}$. Antes do processo de decodificação, $\tilde{\mathbf{y}}_R$ é desentrelaçado. O sinal recebido na entrada do decodificador do *relay* é dado por

$$\mathbf{y}_R = \mathbf{z}_R + \mathbf{n}, \quad (1)$$

em que

$$\mathbf{z}_R = \mathbf{x}_1 \circ \mathbf{h}_{1R} + \mathbf{x}_2 \circ \mathbf{h}_{2R}. \quad (2)$$

Assumimos que o *relay* tem informação completa sobre o estado do canal, ou seja, conhece os vetores \mathbf{h}_{1R} e \mathbf{h}_{2R} .

A operação conjunta de decodificação de canal e codificação de rede na camada física, realizada no *relay* na fase MA, e representada pelo operador $\text{CNC}(\cdot)$, foi descrita em [4]. Este operador produz uma estimativa de $\mathbf{u}_R = [u_R(1), \dots, u_R(N)]$, em que $u_R(t) = u_1(t) \oplus u_2(t)$, baseado no sinal recebido \mathbf{y}_R , ou seja, $\hat{\mathbf{u}}_R = \text{CNC}(\mathbf{y}_R)$. A estimação direta de \mathbf{u}_R através de \mathbf{y}_R obtida pelos decodificadores turbo será descrita na Seção III.

Finalmente, no estágio BC, os bits XOR estimados $\hat{\mathbf{u}}_R$ são novamente codificados, com o mesmo código utilizado nas fontes, e remodulados pelo *relay* gerando uma sequência de símbolos \mathbf{x}_R . Essa sequência deve então ser entrelaçada e enviada às fontes através dos canais com desvanecimento Rayleigh $\tilde{\mathbf{h}}_{R1}$ e $\tilde{\mathbf{h}}_{R2}$. Voltamos a assumir que o decodificador S_i conhece o estado do canal $\tilde{\mathbf{h}}_{Ri}$ que conecta R a S_i . Essa informação é usada para obter a estimação de $\hat{\mathbf{u}}_R$ com base no sinal transmitido pelo nó *relay*, \mathbf{x}_R . Como $S_{1(2)}$ conhece sua própria informação, os dados de $S_{2(1)}$ podem ser facilmente obtidos por uma simples operação XOR.

Uma vez que o processo de decodificação realizado pelas fontes na fase BC é padrão, na próxima seção, descreveremos apenas a decodificação realizada pelo nó *relay*, por ser esse o processo mais crítico e inovador.

III. DECODIFICAÇÃO TURBO-XOR

O processo de decodificação iterativa baseada no uso conjunto de PNC e código turbo é aqui detalhado. O trabalho é focado no desenvolvimento de dois decodificadores para o nó *relay*, que receberão os sinais sobrepostos e desvanecidos, \mathbf{y}_R . Eles devem ter a capacidade de minimizar a probabilidade de erro de bits de $u_R(t) = u_1(t) \oplus u_2(t)$, no tempo t , via decodificação turbo iterativa. Utilizamos o algoritmo BCJR [9] para produzir, de forma independente, estimativas melhoradas das probabilidades *a priori* de $u_R(t)$ com base nos sinais sobrepostos e corrompidos, \mathbf{y}_R . Estas estimativas são expressas através da LLR *a posteriori*

$$\Lambda(u_R(t) | \mathbf{y}_R) = \ln \left(\frac{P(u_R(t) = 1 | \mathbf{y}_R)}{P(u_R(t) = 0 | \mathbf{y}_R)} \right). \quad (3)$$

O critério de decisão final sobre o bit $u_R(t)$ é o habitual:

$$\hat{u}_R(t) = \begin{cases} 0, & \text{se } \Lambda(u_R(t) | \mathbf{y}_R) \leq 0 \\ 1, & \text{caso contrário.} \end{cases}$$

No entanto, é necessário modificar o algoritmo de decodificação BCJR para que este possa operar em uma nova treliça, aqui denominada por *treliça produto*, projetada para o decodificador turbo-XOR. Esta é construída tendo como base as treliças T_1 e T_2 , que representam as treliças dos códigos utilizados em S_1 e S_2 , respectivamente. A Figura 2 ilustra esta construção.

Considere a treliça T_i associada ao código utilizado em S_i . Seja $s_i(t)$ o vetor de estado do codificador turbo da fonte S_i no tempo t e faça a transição de estado do codificador sobre a recepção do símbolo fonte $u_i(t) = 0$ (resp. $u_i(t) = 1$) ser representada por uma linha sólida (resp. tracejada). O codificador convolucional que compõe o codificador turbo possui memória igual a ν_i . Além disso, assuma que, para S_1 , o estado inicial no tempo t é dado por k , ou seja, $s_1(t-1) = k$; a transição quando $u_1(t) = 0$ leva ao estado l originando o bloco codificado $\mathbf{v}'_1(t)$ e a transição quando $u_1(t) = 1$ leva ao estado m originando o bloco codificado $\mathbf{v}''_1(t)$. Da mesma forma, para S_2 , nós temos que o estado inicial no tempo t é q , ou seja, $s_2(t-1) = q$; a transição quando $u_2(t) = 0$ leva ao estado r originando a palavra código $\mathbf{v}'_2(t)$ e a transição quando $u_2(t) = 1$ leva ao estado s e palavra código $\mathbf{v}''_2(t)$. Aqui, $k, l, m \in \{1, \dots, 2^{\nu_1}\}$ e $q, r, s \in \{1, \dots, 2^{\nu_2}\}$.

As saídas, $\mathbf{v}'_i(t)$ e $\mathbf{v}''_i(t)$, dos codificadores podem então ser definidas como uma função do estado do codificador e do símbolo de entrada utilizando o mapeamento: $\Gamma[s_i(t-1), u_i(t)]$.

Considere agora a treliça produto T_p . Seja $s_p(t)$ o vetor de estados do decodificador turbo para o *relay* no tempo t . Os estados de T_p são formados pela concatenação dos estados das treliças dos códigos correspondentes, ou seja, $s_p(t) = s_1(t) \parallel s_2(t)$, em que \parallel denota concatenação. Por exemplo, se o estado de T_1 e T_2 são k e q respectivamente, o estado de T_p é $k \parallel q$ (ou simplesmente, kq). Do estado $s_p(t-1) = kq$, são possíveis quatro transições, $s_p(t)$, para os estados lr, ls, mr, ms , os quais são combinações de todos os possíveis estados de $s_1(t)$ e $s_2(t)$. Portanto, $lr, ls, mr, ms \in \{1, \dots, 2^{\nu_1 + \nu_2}\}$.

Na treliça produto, as transições entre dois estados são representados por uma linha que liga os estados. Criamos a variável $u_{\mathbb{F}_4}(t)$ definida como $u_{\mathbb{F}_4}(t) \triangleq 2u_1(t) + u_2(t) \in \{0, 1, 2, 3\}$ para representar esta linha, como mostrado na Figura 2. A saída da treliça produto é também definida como uma função do novo conjunto de símbolos de entrada utilizando o mapeamento $\Gamma'[s_p(t-1), u_{\mathbb{F}_4}(t)] = \Gamma[s_1(t-1), u_1(t)] \circ \mathbf{h}_{1R}(t) + \Gamma[s_2(t-1), u_2(t)] \circ \mathbf{h}_{2R}(t)$.

Podemos verificar que na treliça produto as transições entre estados são classificadas de quatro maneiras distintas. Faça $l' \triangleq s_p(t-1)$ e $l \triangleq s_p(t)$. Então, para $i = 0, 1, 2, 3$, definimos que $\mathcal{R}_i \triangleq \{(l', l) \mid (l', l) \text{ é associado com } u_{\mathbb{F}_4}(t) = i\}$, onde (l', l) denota a transição do estado l' para o estado l .

Os decodificadores turbo-XOR aqui propostos, consideram em seus processos iterativos, que as regiões \mathcal{R}_i para $i =$

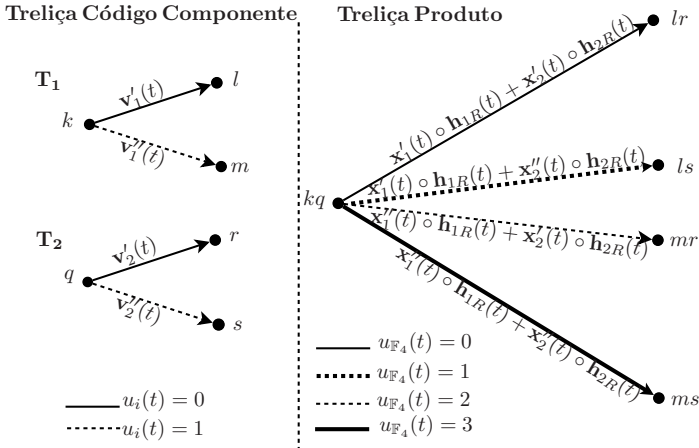


Fig. 2. Estrutura da treliça produto.

0, 1, 2, 3 carregam duas ou quatro diferentes informações. As subseções que seguem apresentam mais detalhes.

A. Decodificação XOR- \mathbb{F}_2

Observando o lado direito da Figura 2, temos dois ramos sólidos, $u_{\mathbb{F}_4}(t) = 0$ e $u_{\mathbb{F}_4}(t) = 3$, responsáveis por $u_R(t) = 0$ e dois ramos tracejados, $u_{\mathbb{F}_4}(t) = 1$ e $u_{\mathbb{F}_4}(t) = 2$, para $u_R(t) = 1$. O processo de decodificação XOR- \mathbb{F}_2 considera que os ramos em $\mathcal{R}_0 \cup \mathcal{R}_3$, ou $\mathcal{R}_1 \cup \mathcal{R}_2$, carregam a mesma informação. Os valores de $P(u_R(t) = 0 | \mathbf{y}_R)$ e $P(u_R(t) = 1 | \mathbf{y}_R)$ no tempo t são então calculados como

$$P(u_R(t) = 0 | \mathbf{y}_R) = \sum_{(l', l) \in \mathcal{R}_0 \cup \mathcal{R}_3} P(l', l | \mathbf{y}_R)$$

e

$$P(u_R(t) = 1 | \mathbf{y}_R) = \sum_{(l', l) \in \mathcal{R}_1 \cup \mathcal{R}_2} P(l', l | \mathbf{y}_R),$$

onde $P(l', l, \mathbf{y}_R) = \alpha_{t-1}(l') \cdot \gamma_t(\mathbf{y}_R(t), l', l) \cdot \beta_t(l)$ para todo $t = 1, \dots, N$. Os valores de α e β são obtidos de forma padrão. Contudo, o cálculo de γ_t dependerá da saída atual do canal bem como das probabilidades de transição da treliça produto. Devido a presença do entrelaçador externo, o canal visto pelo relay, se comporta como um canal sem memória. O resultado disso faz com que seu valor seja dado por

$$\gamma_t(\mathbf{y}_R(t), l', l) = P(u_R(t)) \exp\left(-\frac{\sum_{k=1}^n d^2(y_R^k(t), z_R^k(t))}{2\sigma^2}\right)$$

onde $z_R^k(t)$ é o rótulo da transição de estado correspondente (l', l) na treliça produto, $d(a, b)$ é a distância Euclidiana entre os pontos a e b, no plano complexo, e σ^2 é a variância do ruído. A probabilidade *a priori* de $u_R(t)$ é inicialmente setada como $P(u_R(t)) = 1/2$ e é obtida e melhorada na saída do decodificador turbo. A expressão da LLR *a posteriori* de $u_R(t)$ pode então ser reescrita como

$$\Lambda(u_R(t) | \mathbf{y}_R) = \ln \frac{\sum_{(l', l) \in \mathcal{R}_1 \cup \mathcal{R}_2} P(l', l, \mathbf{y}_R)}{\sum_{(l', l) \in \mathcal{R}_0 \cup \mathcal{R}_3} P(l', l, \mathbf{y}_R)}.$$

A informação extrínseca do decodificador turbo XOR- \mathbb{F}_2 pode ser representada como

$$\Lambda_e(u_R(t)) = \Lambda(u_R(t) | \mathbf{y}_R) - \Lambda_{u_R} - \Lambda_S$$

em que $\Lambda(u_R(t) | \mathbf{y}_R)$ é a LLR *a posteriori* calculada no processo de decodificação atual, Λ_{u_R} é a informação *a priori* de $u_R(t)$ calculada no processo de decodificação passado e Λ_S representa a parcela da LLR *a posteriori* relacionada apenas com os bits sistemáticos, $z_R^1(t)$, que por sua vez é dada por

$$\begin{aligned} \Lambda_S &= \ln \left(\frac{p(u_R(t) = 1 | y_R^1(t))}{p(u_R(t) = 0 | y_R^1(t))} \right) \\ &= \ln \left(\frac{p(y_R^1(t) | u_R(t) = 1)P(u_R(t) = 1)}{p(y_R^1(t) | u_R(t) = 0)P(u_R(t) = 0)} \right), \end{aligned}$$

onde, como indicado acima no texto, $P(u_R(t))$ é uniformemente distribuída sobre \mathcal{R}_i antes da primeira iteração. A função densidade de probabilidade da saída ser y_R^1 dado que a entrada $u_R(t)$ ocorreu é dada por

$$p(y_R^1(t) | u_R(t)) = \frac{E_b}{\sigma\sqrt{2\pi}} \exp\left(-\frac{|y_R^1(t) - z_R^1(t)|^2}{2\sigma^2}\right),$$

em que $|\cdot|$ é a operação módulo, E_b é a energia de bit, e σ^2 é a variância do ruído.

B. Decodificação XOR- \mathbb{F}_4

O processo de decodificação XOR- \mathbb{F}_4 considera que os quatro conjuntos de transição associados aos símbolos $u_{\mathbb{F}_4}(t)$ carregam diferentes informações. Assim consegue-se explorar ao máximo a redundância do código proposto. Apenas ao final do processo de decodificação iterativa, se obtém a estimativa do XOR. Os valores das probabilidades $P(u_{\mathbb{F}_4}(t) = i | \mathbf{y}_R)$, para $i = 0, 1, 2$ e 3 , no tempo t , são então calculados como

$$P(u_{\mathbb{F}_4}(t) = i | \mathbf{y}_R) = \sum_{(l', l) \in \mathcal{R}_i} P(l', l | \mathbf{y}_R),$$

e a j -ésima LLR *a posteriori* de $u_{\mathbb{F}_4}(t)$ para $j = 1, 2$ e 3 é dada por

$$\Lambda_j(u_{\mathbb{F}_4}(t) | \mathbf{y}_R) = \ln \frac{\sum_{(l', l) \in \mathcal{R}_0} P(l', l, \mathbf{y}_R)}{\sum_{(l', l) \in \mathcal{R}_j} P(l', l, \mathbf{y}_R)}.$$

O cálculo de γ_t se dá de forma similar ao apresentado para o processo de decodificação XOR- \mathbb{F}_2 . Antes da primeira iteração, os valores *a priori* de $u_{\mathbb{F}_4}(t)$ são uniformemente distribuídos em $\{0, 1, 2, 3\}$. O cálculo das informações extrínsecas também se dará da mesma forma que no processo XOR- \mathbb{F}_2 . Porém, lembramos que agora teremos que calcular três parcelas. Para $j = 1, 2$ e 3 , tem-se

$$\Lambda_{j,e}(u_{\mathbb{F}_4}(t)) = \Lambda_j(u_{\mathbb{F}_4}(t) | \mathbf{y}_R) - \Lambda_j - \Lambda_{j,S}.$$

Ao término do processo iterativo, calcula-se a LLR *a posteriori* final da seguinte forma

$$\Lambda(u_R(t) | \mathbf{y}_R) = \ln \frac{\sum_{\mathcal{R}_1} P(l', l, \mathbf{y}_R) + \sum_{\mathcal{R}_2} P(l', l, \mathbf{y}_R)}{\sum_{\mathcal{R}_0} P(l', l, \mathbf{y}_R) + \sum_{\mathcal{R}_3} P(l', l, \mathbf{y}_R)}.$$

IV. RESULTADOS DAS SIMULAÇÕES

O desempenho dos algoritmos de decodificação XOR propostos são avaliados através de simulações de computador. As fontes S_1 e S_2 empregam o mesmo codificador turbo que é composto pela concatenação paralela de dois codificadores recursivos e sistemáticos com o comprimento de memória $\nu_i = 2$ e descritos pela matriz geradora $\mathbf{G}(D) = [1 \ (1 + D^2)/(1 + D + D^2)]$. Este código tem taxa igual a $1/3$. Para aumentar a taxa, removemos alguns dos bits de paridade através da técnica de puncionamento utilizando a matriz $\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^T$, em que 0 (zero) indica que o bit foi puncionado e portanto não transmitido. De acordo com este padrão de puncionamento, os bits de paridade dos dois codificadores RSC são alternadamente excluídos. A taxa conseguida por este código puncionado é, portanto, $1/2$. Foram utilizados dois entrelaçadores distintos. Na simulação, o tamanho do vetor mensagem \mathbf{u}_i é definido como 4000 bits. O entrelaçador interno que conecta os codificadores convolucionais que constituem o código turbo é um entrelaçador aleatório que gera a permutação do bloco de 4000 bits. O entrelaçador externo é também aleatório e opera com blocos de 8000 bits. O papel do entrelaçador externo é o de espalhar os símbolos codificados ao longo do tempo, para evitar os efeitos do desvanecimento profundo. Os canais, das fontes para o *relay* e do *relay* para as fontes, são canais com desvanecimento independentes. As sequências dos coeficientes de canal são obtidos através do modelo Jakes [10]. Este procedimento corresponde à geração de sequências independentes de variáveis aleatórias Gaussianas complexas de potência unitária correlacionadas. Os resultados são mostrados em termos da frequência Doppler normalizada ($f_d T$), onde f_d é a frequência Doppler e T é a duração do símbolo. Nas simulações, uma $f_d T = 0.05$ foi assumida.

As simulações foram realizadas no final da fase MA, no nó *relay*, uma vez que o processo de decodificação realizado pelas fontes na fase BC é padrão. O número de iterações dos decodificadores turbo propostos, em todas as simulações realizadas neste trabalho, foi fixado em 6.

Não mostramos aqui a saturação inerente ao processo de decodificação turbo, que depende do tamanho do pacote utilizado. O problema aparece pela não terminação da treliça do segundo decodificador, ocasionada pela permutação dos bits no processo de codificação turbo. Uma correta escolha do entrelaçador [11] faz com que os dois decodificadores consigam retornar suas treliças ao estado zero. Em nossas simulações, porém, emulamos essa possibilidade através da transmissão de sequências *toda-zero*. Essa análise merece ser melhor investigada.

A Figura 3 mostra o desempenho do decodificador quando mudamos de XOR- \mathbb{F}_2 para XOR- \mathbb{F}_4 . O processo de decodificação XOR- \mathbb{F}_4 melhora o desempenho em quase 1.5 dB.

Observamos que a maioria dos trabalhos que se utilizam da técnica PNC assumem canais Gaussianos cujo coeficiente de desvanecimento é sempre unitário. Aqui simulamos canais reais sem fio. A referência [6] apresenta uma proposta semelhante à nossa, porém com um desempenho bastante inferior mesmo quando comparado ao desempenho do decodificador

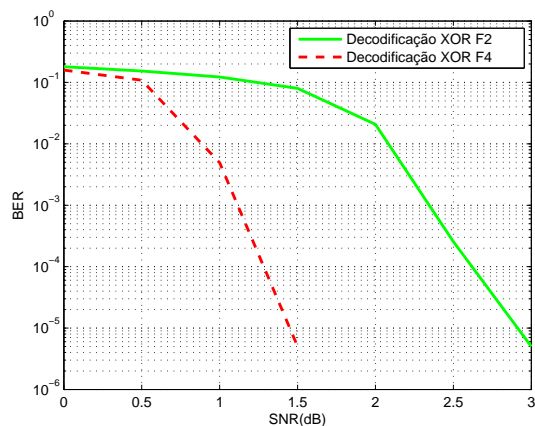


Fig. 3. Comparação entre os processos de decodificação XOR- \mathbb{F}_2 e \mathbb{F}_4 .

turbo XOR- \mathbb{F}_2 . Essa melhora é devida ao cálculo direto da LLR de $u_R(t) = u_1(t) \oplus u_2(t)$, onde não decidimos sobre $u_1(t)$ e $u_2(t)$ individualmente para depois calcular $u_R(t)$, como é feito em [6].

V. CONCLUSÕES

Propusemos a utilização conjunta de PNC e decodificação turbo com base em uma treliça completa e uma computação modificada da métrica para obter a informação XOR diretamente dos sinais sobrepostos. As simulação utilizando o canal Rayleigh mostraram que o desempenho de erro do sistema foi substancialmente melhorado através dessa simples modificação do algoritmo de decodificação BCJR.

REFERÊNCIAS

- [1] R. Ahlswede, N. Cai, and S. Li, "Network information flow," *Information Theory, IEEE*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] S. Zhang, S. C. Liew, and P. P. Lam, "Hot Topic : Physical-Layer Network Coding," in *Proc. 12th MobiCom*, pp. 358–365, 2006.
- [3] C. E. Shannon, "Two-way communication channels," in *Proc. 4th Berkeley Symp. Math. Stat. and Prob.*, vol. 1, 1961, pp. 611–644.
- [4] S. Zhang and S. C. Liew, "Channel coding and decoding in a relay system operated with physical-layer network coding," *Selected Areas in Communications, IEEE*, vol. 27, no. 5, pp. 788–796, 2009.
- [5] S. Lu, Y. Li, and J. Cheng, "Low-complexity turbo decoding scheme for two-way relay network," in *Wireless Communications and Signal Processing (WCSP), 2010 International Conference on*. IEEE, 2010, pp. 1–5.
- [6] D. To and J. Choi, "Reduced-state decoding in two-way relay networks with physical-layer network coding," in *Information Theory Workshop (ITW), 2010 IEEE*. IEEE, 2010, pp. 1–5.
- [7] D. Wubben and Y. Lang, "Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems," in *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*. IEEE, 2010, pp. 1–5.
- [8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC 93. Geneva. Technical program, conference record, IEEE international conference on*, vol. 2, no. 1. IEEE, 1993, pp. 1064–1070.
- [9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284–287, 1974.
- [10] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.
- [11] J. Hokfelt, O. Edfors, and T. Maseng, "A survey on trellis termination alternatives for turbo codes," in *Vehicular Technology Conference, 1999 IEEE 49th*, vol. 3. IEEE, 1999, pp. 2225–2229.