Improved Joint Turbo Decoding and Physical-Layer Network Coding

Maria Cláudia Castro*†, Bartolomeu F. Uchôa-Filho*, Tiago T. V. Vinhoza‡, Mário Noronha-Neto†, João Barros‡
*Universidade Federal de Santa Catarina, Florianópolis, Brazil
†Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Florianópolis, Brazil
‡Instituto 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

***Abstract*—We present an improved decoding algorithm for joint turbo decoding and physical-layer network coding. Instead of decoding the individual (binary) messages separately at the relay, the proposed algorithm, from the superimposed faded signals, yields an XOR estimate of the sent messages. Moreover, we introduce a softening of the XOR values to improve the overall performance. Simulation results show that this simple idea yields gains up to 4.5 dB in a Rayleigh fading channel model when compared to a similar scheme.**

I. Introduction

The known benefits of network coding offer a strong motivation for finding adequate coding schemes, particularly for wireless channels. The key idea of network coding was first proposed by Ahlswede *et al.* [1] who showed that by allowing intermediate nodes to combine packets before forwarding them, maximum information flow can be achieved in a network. Network coding was first proposed to operate at the upper layers of the protocol stack, and thus was mainly concerned with the problem of encoding data over links with a certain capacity, ignoring the underlying physical nature of the communications channels. More recently, the principles of network coding have been applied at the physical layer, by exploiting the natural superposition of electromagnetic waves that occurs in wireless communications. This superposition is generally regarded as an obstacle to reliable communication, where the recovery of individual signals is required. However, in the presence of side information, the addition of signals can have a positive effect, enhancing communication efficiency.

Specifically, a coding strategy developed by Zhang *et al.* [2], known as Physical-Layer Network Coding (PNC), considers the "wave mixing" as a natural way of network coding operation that can make communication more efficient.

The network configuration usually adopted as the object of study in PNC is the two-way relay network (TWRN), in which two users that can only communicate through a relay want to exchange messages. With PNC, the two users first transmit independent binary data streams simultaneously and the relay receives the superimposed signals. The relay then extracts, out of the received signal, the mod-2 sum (XOR) of the two data streams, without having to obtain the two individual data streams explicitly. The XORed data is then broadcast by the relay, allowing each user to recover the other user's data stream. The whole process requires only two time slots, the minimum possible for this set-up.

Despite using interference in a beneficial manner, one still has to deal with channel noise. One way to apply channel coding in PNC is called link-by-link coded PNC, in which not only the two users but also the relay perform channel coding and decoding. The two-time slot schedule is the same as the one described in the previous paragraph. The difference is that now channel-coded (in contrast with raw) data are transmitted by both users and by the relay. The crucial operation, called Channel-Decoding-Network-Coding (CNC) process in [3], takes place in the relay and is characterized by the recovery of the channel-uncoded but network-coded (XORed) information from the superimposed channel-coded signals. We herein adopt this approach.

The channel codes that were integrated to PNC in [3] are the so-called repeat-accumulate (RA) codes, which can be seen simultaneously as a class of "turbo-like" codes and a class of low-density parity-check (LDPC) codes. While these two classes of codes correspond to the two most important capacity-achieving codes, the results in [3] were restricted to the additive white Gaussian noise (AWGN) channel, i.e., the channel gains for all links were assumed constant at all times. In our work, we extend this scenario and assume the Rayleigh fading channel model. We focus our work on turbo codes and the integration of the PNC in the BCJR decoding process.

Recently, in [4], [5], the authors used turbo codes integrated to PNC over both AWGN and Rayleigh fading channels. It is reported that, in their turbo decoder, the two data messages (associated with the two users) at the relay are first recovered individually, and then the XOR is formed. To reduce the decoding complexity of the turbo decoder, they focused on building a trellis with reduced number of states, based upon which the turbo decoder is reportedly able to recover the XORed information directly from the superimposed signals. However, results indicate that the error performance is significantly affected. In [6], the authors proposed a generalized sum-product algorithm for LDPC codes integrated to PNC over fading channels, which improves over previous results.

Contrasting with [4], [5], we propose the joint use of PNC with a turbo decoder based on a complete trellis where the relay computes the XOR of the messages sent by the sources

This work was partly supported by FCT project PEst-OE/EEI/LA0008/2011.

directly from the superimposed signals. For a better performance of the turbo decoder, we explore the idea originally applied to LDPC codes in [6] in our proposed turbo decoding. It can be interpreted as a softening operation applied to the binary XOR values. It should be mentioned that this cannot be implemented in the reduced trellis adopted in [4], [5]. Our simulation results show that this approach achieves a good performance, even in the presence of Rayleigh fading channels. Considerable gains are obtained in comparison with [4], [5], albeit at the expense of a more complex encoder/decoder. This additional complexity, however, is still manageable.

Our fading channel model also differs from the one in [4], [5]. While the fading channel coefficients therein are i.i.d., herein we assume a time-correlated model, following the well-known Jakes model [7]. While the effect of channel correlation on the error performance of coded PNC is quite negative, we introduce an interleaving operation, performed in each user unit prior to transmission, in order to break the channel memory. Simulation results show that a substantially improved error performance can be obtained by this operation.

This paper is organized as follows. Section II describes the system model. In Section III, we give the construction of the product trellis and discuss the proposed soft decoding algorithm. Simulation results and comparisons with related work are provided in Section IV. Finally, Section V concludes the paper.

II. System Model

Notation: In what follows, vectors/matrices are represented by lowercase boldface letters, except $\mathbf{I}_k$ which represents the $k \times k$ identity matrix, $\oplus$ represents the XOR operation, and $\circ$ represents the Hadamard (component-wise) product. We use $\tilde{\mathbf{x}}$ to denote an interleaved version of vector $\mathbf{x}$.

We consider a TWRN, where two source nodes, S_1 and S_2, wish to communicate with each other through an intermediate (relay) node R. There is no direct link between S_1 and S_2. When PNC is employed at the relay node, the exchange of data between S_1 and S_2 is made in two distinct time-slots, referred to as the multiple access (MA) stage and the broadcast (BC) stage. In the MA stage, both S_1 and S_2 transmit their signals to the relay node. Due to half-duplex constraint the sources cannot listen to any information. The relay node receives the superimposed faded signals from S_1 and S_2 and estimates the XOR of the original messages. It should be emphasized that no knowledge of each individual messages is required for this process. Afterwards, in the BC phase, the relay node reencodes the estimated XORed message and, taking advantage of the broadcast nature of the wireless medium, transmits the coded signal back to both S_1 and S_2. The system model is depicted in Figure 1.

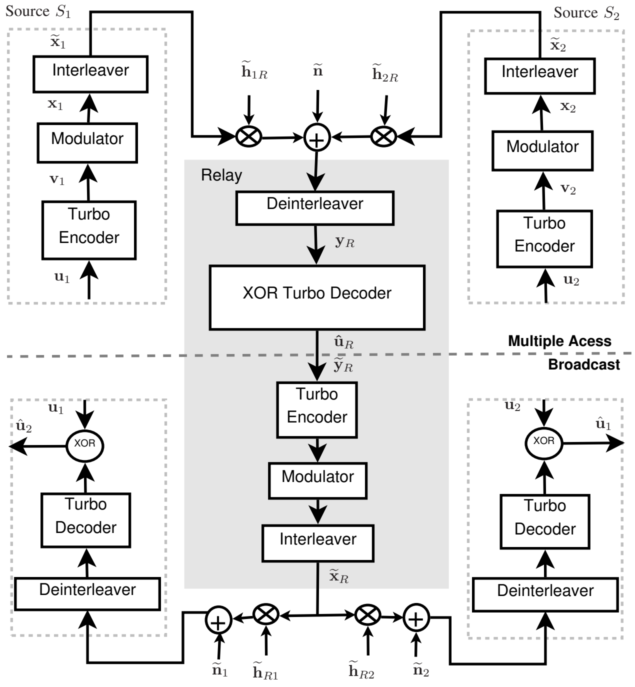

Figure 1. Proposed coding scheme for the two-way relay network.

We assume that all nodes employ turbo codes [8] to encode their messages. The encoder consists of two identical recursive systematic convolutional (RSC) encoders operating in parallel with an interleaver between them. For simplicity of exposition, we assume a turbo code rate equal to $1/n$. For $i = 1, 2$, let $\mathbf{u}_i = [u_i(1), \ldots, u_i(N)]$ denote the message vector of length N containing information bits from source S_i, and let $\mathbf{v}_i = [\mathbf{v}_i(1), \ldots, \mathbf{v}_i(N)]$ be the corresponding coded sequence of length nN bits. Each coded block $\mathbf{v}_i(t) = \left[v_i^1(t), \ldots, v_i^n(t)\right]$ is a vector of length n bits and corresponds to the turbo encoder output for the input bit $u_i(t)$ at time t. We assume a systematic encoder, so that $v_i^1(t) = u_i(t)$ for all t. The coded block $\mathbf{v}_i$ is then BPSK modulated to produce $\mathbf{x}_i$, that is, $x_i^j(t) = 1 - 2v_i^j(t)$ for $j = 1, \ldots, n$. We assume that the channels between the source nodes and the relay node are time-correlated Rayleigh fading channels, which may lead to burst error-events. To remedy this problem, we employ an outer interleaver before transmitting the modulated signal. The outer interleaver for the two source nodes must be the same in order to receive the two codewords aligned in time at the relay. Each source node then transmits $\tilde{\mathbf{x}}_i$.

The flat fading channel from S_i to R is characterized by the channel coefficients vector, $\tilde{\mathbf{h}}_{iR} = \left[\tilde{\mathbf{h}}_{iR}(1), \ldots, \tilde{\mathbf{h}}_{iR}(N)\right]$, where $\tilde{\mathbf{h}}_{iR}(t) = \left[\tilde{h}_{iR}^1(t), \ldots, \tilde{h}_{iR}^n(t)\right]$. The received signal at the relay node is given by $\tilde{\mathbf{y}}_R = \tilde{\mathbf{z}}_R + \tilde{\mathbf{n}}$, where $\tilde{\mathbf{z}}_R = [\tilde{\mathbf{z}}_R(1), \ldots, \tilde{\mathbf{z}}_R(N)]$ contains the superimposed faded signals: $\tilde{\mathbf{z}}_R(t) = \tilde{\mathbf{x}}_1(t) \circ \tilde{\mathbf{h}}_{1R}(t) + \tilde{\mathbf{x}}_2(t) \circ \tilde{\mathbf{h}}_{2R}(t)$, and $\tilde{\mathbf{n}} = [\tilde{\mathbf{n}}(1), \ldots, \tilde{\mathbf{n}}(N)]$ is a complex-valued Gaussian noise vector, with zero mean and covariance matrix $\sigma^2 \mathbf{I}_{nN}$. Before proceeding to the decoding process, $\tilde{\mathbf{y}}_R$ is deinterleaved. The received signal at the input of the relay decoder is then given by

$$\mathbf{y}_R = \mathbf{z}_R + \mathbf{n}, \tag{1}$$

where

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

We assume that the relay has full channel state information as well as the interleaving pattern, hence it knows the entire vectors $\mathbf{h}_{1R}$ and $\mathbf{h}_{2R}$.

The joint channel decoding and physical-layer network encoding performed at the relay in the MA stage is represented by the operator CNC$(\cdot)$, as described in [3]. This operator outputs an estimate of $\mathbf{u}_R = [u_R(1), \ldots, u_R(N)]$, where $u_R(t) = u_1(t) \oplus u_2(t)$, based on the received signal $\mathbf{y}_R$, *i.e.*, $\widehat{\mathbf{u}}_R = \text{CNC}(\mathbf{y}_R)$. The direct estimation of $\mathbf{u}_R$ through $\mathbf{y}_R$ is obtained by the improved turbo decoder that is described in Section III.

Finally, in the BC stage, the estimated XORed bits $\widehat{\mathbf{u}}_R$ are reencoded, with the same code used in the sources, and remodulated by the relay generating a sequence of symbols $\mathbf{x}_R$. This sequence is interleaved and broadcast to the sources again through Rayleigh fading channels $\tilde{\mathbf{h}}_{R1}$ and $\tilde{\mathbf{h}}_{R2}$. We again assume that the decoder in S_i has full channel state information about the channel $\tilde{\mathbf{h}}_{Ri}$ connecting R to S_i. This information is used to obtain the maximum a posteriori estimate of the $\widehat{\mathbf{u}}_R$ on the basis of the signal transmitted by the relay node, $\mathbf{x}_R$. As $S_{1(2)}$ knows their own information, the data from $S_{2(1)}$ can be easily decoded since each component can be obtained by simple (with and without softening) XOR operation.

Since the turbo decoding performed by the sources in the BC stage is standard, we will in the next section describe the turbo decoding performed by the relay only.

III. Soft XOR Turbo Decoder

Upon receiving the superimposed faded signal, $\mathbf{y}_R$, the relay tries to recover the XORed sequence $u_R(t)$ using an iterative decoder which we describe next. It makes use of two decoders serially concatenated via an interleaver. We use the BCJR algorithm [9] to independently produce improved estimates of the a posteriori probabilities of $u_R(t)$ based on $\mathbf{y}_R$. These estimates are expressed by the a posteriori log-likelihood ratio (LLR)

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

When a predetermined number of iterations is reached, a hard decision will be made in the second decoder based on the following rule:

$$\widehat{u}_R(t) = \begin{cases} 0, & \text{if } \Lambda(u_R(t) \mid \mathbf{y}_R) \leq 0 \\ 1, & \text{otherwise.} \end{cases} \tag{4}$$

However, we modify the BCJR decoding algorithm to account for the PNC operation, where an estimate of the XORed message is sought.

The new trellis construction is based on the code trellises T_1 and T_2, for the codes used in S_1 and S_2, respectively. This new trellis, which we call *product trellis*, will be used to successfully decode the received superimposed faded signals. Figure 2 illustrates the product trellis construction.

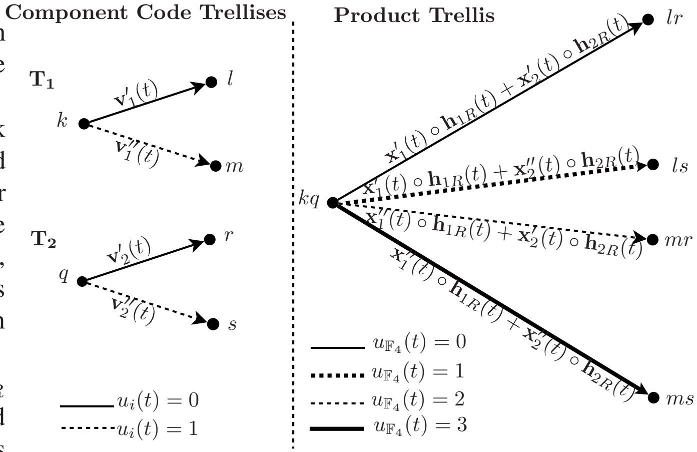
Component Code Trellises
Product Trellis
$\mathbf{T_1}$
k
$\mathbf{v}'_1(t)$
l
$\mathbf{v}''_1(t)$
m
$\mathbf{T_2}$
q
$\mathbf{v}'_2(t)$
r
$\mathbf{v}''_2(t)$
s
$u_i(t) = 0$
$u_i(t) = 1$
kq
$\mathbf{x}'_1(t) \circ \mathbf{h}_{1R}(t) + \mathbf{x}'_2(t) \circ \mathbf{h}_{2R}(t)$
lr
$\mathbf{x}'_1(t) \circ \mathbf{h}_{1R}(t) + \mathbf{x}''_2(t) \circ \mathbf{h}_{2R}(t)$
ls
$\mathbf{x}''_1(t) \circ \mathbf{h}_{1R}(t) + \mathbf{x}'_2(t) \circ \mathbf{h}_{2R}(t)$
mr
$\mathbf{x}''_1(t) \circ \mathbf{h}_{1R}(t) + \mathbf{x}''_2(t) \circ \mathbf{h}_{2R}(t)$
ms
$u_{\mathbb{F}_4}(t) = 0$
$u_{\mathbb{F}_4}(t) = 1$
$u_{\mathbb{F}_4}(t) = 2$
$u_{\mathbb{F}_4}(t) = 3$

Figure 2. Product trellis construction.

Consider the code trellis T_i for the code used in S_i. Let $s_i(t)$ denote the vector state of the turbo encoder of sender S_i at time t and let the state transition of the encoder upon the reception of the source symbol $u_i(t) = 0$ (resp. $u_i(t) = 1$) be represented by a solid (resp. dashed) line. The constituent convolutional encoders of the turbo encoder have memory ν_i. Furthermore, assume that, for S_1, the initial state at time t is k, i.e. $s_1(t-1) = k$; a transition when $u_1(t) = 0$ leads to state l outputting the codeword $\mathbf{v}'_1(t)$, and a transition when $u_1(t) = 1$ leads to state m outputting the codeword $\mathbf{v}''_1(t)$. For S_2 we assume that the initial state at time t is q, i.e. $s_2(t-1) = q$; a transition when $u_2(t) = 0$ leads to state r outputting the codeword $\mathbf{v}'_2(t)$, and a transition when $u_2(t) = 1$ leads to state s outputting the codeword $\mathbf{v}''_2(t)$. Here, $k, l, m \in \{1, \ldots, 2^{\nu_1}\}$ and $q, r, s \in \{1, \ldots, 2^{\nu_2}\}$. The outputs of the encoder $\mathbf{v}'_i(t)$ and $\mathbf{v}''_i(t)$ can then be defined as a function of the encoder state and the input symbol using the mapping $\Gamma[s_i(t-1), u_i(t)]$.

Consider now the product trellis T_p. Let $s_p(t)$ denote the vector state of the turbo decoder at the relay at time t. The states of T_p are formed by the concatenation of the states of the corresponding code trellises, i.e. $s_p(t) = s_1(t) \parallel s_2(t)$, where $\parallel$ denotes concatenation. For instance, if the states of T_1 and T_2 are k and q, respectively, then the state of T_p is $k \parallel q$ (or simply, kq). From state $s_p(t-1) = kq$, transitions are possible to states lr, ls, mr, ms, which are combinations of all the possible states $s_1(t)$ and $s_2(t)$. Accordingly, $lr, ls, mr, ms \in \{1, \ldots, 2^{\nu_1}\} \times \{1, \ldots, 2^{\nu_2}\}$, of size $2^{\nu_1+\nu_2}$.

In the product trellis, the transitions between two states are represented by a straight line connecting the states according to the new variable $u_{\mathbb{F}_4}(t)$ defined as $u_{\mathbb{F}_4}(t) \triangleq 2u_1(t) + u_2(t) \in \{0, 1, 2, 3\}$, as shown in Figure 2. The output of the product trellis is also defined as a function of the new set of input symbols using the mapping $\Gamma'[s_p(t-1), u_{F4}(t)] = \Gamma[s_1(t-1), u_1(t)] \circ \mathbf{h}_{1R}(t) + \Gamma[s_2(t-1), u_2(t)] \circ \mathbf{h}_{2R}(t)$.

We can see that in the product trellis the transitions between states fall into four classes. Let $l' \triangleq s_p(t-1)$ and $l \triangleq s_p(t)$. Then, for $i = 0, 1, 2, 3$, define $\mathcal{R}_i \triangleq$

$\{(l', l) \mid (l', l)$ is associated with $u_{\mathbb{F}_4}(t) = i\}$, where (l', l) denotes the transition from state l' to state l.

The soft XOR turbo decoder considers, in the iterative decoding process, that the symbols $u_{\mathbb{F}_4}(t) = i$, for $i = 0, 1, 2, 3$, associated with the four transition sets just defined, carry different information. Accordingly, the turbo decoder improves, at each iteration, the estimates of the LLRs of the symbol $u_{\mathbb{F}_4}(t)$. At the end of the iterative decoding process, a decision regarding the XORed message is obtained by appropriately combining the LLRs. This is similar to "softening" the turbo decoder as compared to the ones proposed in [4], [5], where the XORed message information is taken into account before computing the LLR metrics.

The probabilities $P(u_{\mathbb{F}_4}(t) = i \mid \mathbf{y}_R)$, for $i = 0, 1, 2, 3$, at time t, can be computed as

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

where $P(l', l, \mathbf{y}_R) = \alpha_{t-1}(l') \cdot \gamma_t(\mathbf{y}_R(t), l', l) \cdot \beta_t(l)$ for $t = 1, \ldots, N$. The values of α and β are obtained as in the standard BCJR algorithm. However, the calculation of γ_t will depend on the current channel output as well as on the transition probabilities of the product trellis. Due to the presence of the outer interleaver, the channel seen at the relay behaves like a memoryless channel. As a result of this, γ_t, for $(l', l) \in \mathcal{R}_i$, where $i \in \{0, 1, 2, 3\}$, can be obtained as

$$\gamma_t(\mathbf{y}_R(t), l', l) = P(u_{\mathbb{F}_4}(t)) \exp\left(-\frac{\sum_{k=1}^{n} d^2(y_R^k(t),\ z_R^k(t))}{2\sigma^2}\right),$$

where $z_R^k(t)$ is the label in the corresponding state transition (l', l) in the product trellis, $d(a, b)$ is the Euclidean distance between the points a and b in the complex plane, and σ^2 is the noise variance.

The a priori probability of $u_{\mathbb{F}_4}(t)$ is initially set to $P(u_{\mathbb{F}_4}(t) = i) = 1/4$, for all $i \in \{0, 1, 2, 3\}$, and is obtained and upgraded at the output of the turbo decoder. For $j = 1, 2$ and 3, the jth a posteriori LLR of $u_{\mathbb{F}_4}(t)$ is given as

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

The extrinsic information, for $j = 1, 2$ and 3, can be represented as

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

where Λ_j denotes the current a priori information of $u_{\mathbb{F}_4}(t)$ obtained from the extrinsic information of the previous iteration, and $\Lambda_{j,S}$ represents the a posteriori LLR responsible only for the portion related to the systematics bits $z_R^1(t)$, which in turn is given by

$$\begin{aligned}\Lambda_{j,S} &= \ln\left(\frac{P(u_{\mathbb{F}_4}(t) = 0 \mid y_R^1(t))}{P(u_{\mathbb{F}_4}(t) = j \mid y_R^1(t))}\right) \\ &= \ln\left(\frac{p(y_R^1(t) \mid u_{\mathbb{F}_4}(t) = 0)P(u_{\mathbb{F}_4}(t) = 0)}{p(y_R^1(t) \mid u_{\mathbb{F}_4}(t) = j)P(u_{\mathbb{F}_4}(t) = j)}\right),\end{aligned}$$

where, as stated above in the text, $u_{\mathbb{F}_4}(t)$ is uniformly distributed over $\{0, 1, 2, 3\}$ before the first iteration. The probability density function of the output $y_R^1(t)$ conditioned on the input $u_{\mathbb{F}_4}(t)$ is given by

$$p(y_R^1(t) \mid u_{\mathbb{F}_4}(t)) = \frac{E_b}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\mid y_R^1(t) - z_R^1(t) \mid^2}{2\sigma^2}\right),$$

where $|\cdot|$ denotes the modulus and E_b is the bit energy.

At the end of the iterative process, we calculate the a posteriori LLR

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

and make a decision based on (4).

IV. Simulation Results

The performance of the proposed decoding algorithm is assessed through computer simulations. The sources S_1 and S_2 employ the same turbo encoder, formed by the parallel concatenation of two recursive systematic encoders with memory length $\nu_i = 2$ and described by the generator matrix $\mathbf{G}(D) = [1\ (1 + D^2)/(1 + D + D^2)]$. This code has rate 1/3. To reduce the amount of redundancy, we remove some of the parity bits through the puncturing matrix $\mathbf{P} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}^T$, where 0 (zero) indicates a punctured bit and 1 (one) a transmitted bit. The rate of the punctured code is 1/2.

We employed two distinct interleavers. In the simulation, the message vector $\mathbf{u}_i$ length was set to 4000 bits. The inner interleaver that connects the constituent convolutional encoders of the turbo encoder is a random interleaver of size 4000 bits. The outer interleaver is also random and works with blocks of 8000 bits. The outer interleaver role is to spread the coded symbols across time to avoid the effects of deep fading.

The sources-to-relay and relay-to-sources channels are independent fading channels. The sequence of channel coefficients is obtained according to the Jakes model [7], and is a time-correlated, unit power complex Gaussian random sequence. The results are shown in terms of the normalized Doppler frequency $(f_d T)$, where f_d is the Doppler frequency and T is the symbol duration. In the simulations, we assumed $f_d T = 0.05$.

The number of iterations of the proposed turbo decoder in all simulations performed in this work was fixed to 6. We should mention that no simulation details including number of iterations of the turbo decoder are provided in [5]. So, in order to compare our results with theirs, we have duplicated their performance curve rather than actually simulating their proposed scheme.

The simulation results are presented in Figures 3 and 4, where the bit error rate (BER) of the estimate of $\mathbf{u}_j$ produced by the decoder in source S_i in the end of the BC stage versus SNR (i.e., the end-to-end BER performance) is given. In Figure 3, we compare the performance of the proposed decoders (both with and without softening) when the outer interleaver is not employed with the decoder proposed in [5]. Our decoder without softening considers, in the iterative decoding process, that the set of the branches $\mathcal{R}_0$ and $\mathcal{R}_3$ (or $\mathcal{R}_1$ and $\mathcal{R}_2$) carries the same information and that induces a 1.5 dB loss in performance. In Figure 4, the same performance comparisons are made with the presence of the outer interleaver. We can see that time-correlation has a negative impact on the performance, and that the adopted interleaver is very effective in removing this drawback. Nevertheless, even without the outer interleaver, our scheme still outperforms the one proposed in [5].

Although not shown, an error floor exists which depends on the message vector size used in the simulation. This occurs due to non-termination of the trellis in the second decoder. This problem has been pointed out in [10], where one possible solution is to choose a specific interleaver design. This choice causes the two decoders to return to the all-zero state. In our simulations, we emulate this possibility through the transmission of the all-zero sequence instead of the actual termination of the trellises. This analysis will be further investigated in the future.

We observe that most of the literature on physical-layer network coding assume only Gaussian channels, i.e. the fading coefficients are always equal to one. In this work, we consider Rayleigh fading channels that are also correlated in time. The work in [5] presents a similar setup, however it has a much lower performance as shown in Figures 3 and 4.

We believe that the improvement of up to 4.5 dB for a BER of 10^{-4} is due to the direct calculation of the LLR of $\mathbf{u}_R$ instead of individually deciding on $\mathbf{u}_1$ and $\mathbf{u}_2$ to then compute $\mathbf{u}_R$, as is done in [5]. The soft XOR turbo decoder can fully exploit the redundancy of the turbo code integrated to PNC, which explains its excellent performance.

We also mention that the estimate $\widehat{\mathbf{u}}_R$ obtained by the decoder in the two source nodes after the BC stage has two possible sources of errors. The first one is due to the possibly erroneous estimation of $\mathbf{u}_R$ from $\mathbf{y}_R$ that takes place at the relay during the MA stage, and the second one is due to fading and noise effects on the signals broadcast by the relay in the BC stage. Therefore, the BER in the relay is slightly less than the BER in the source nodes.

V. Conclusions

We proposed the joint use of PNC with a turbo decoder based on a complete trellis and a modified metric computation to obtain the XORed information directly from the superimposed signals in the two-way relay network. Simulation results over a Rayleigh fading channel scenario showed that a substantially improved error performance can be obtained by this simple modification on the BCJR decoding algorithm.

The same concept developed in this paper for BPSK modulation can be extended to other signal constellations and lattices, which is currently under investigation by the authors.

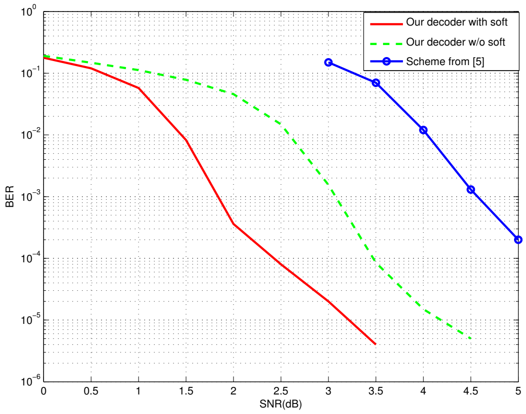

Figure 3. Performance comparison of the XOR decoders (with and without softening) when the outer interleaver is not employed.

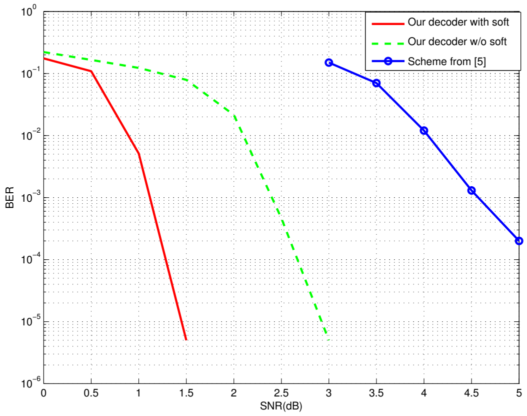

Figure 4. Performance comparison of the XOR decoders (with and without softening) when the outer interleaver is employed.

References

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, 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] S. Zhang and S. C. Liew, “Channel coding and decoding in a relay system operated with physical-layer network coding,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 5, pp. 788–796, 2009.

[4] S. Lu, Y. Li, and J. Cheng, “Low-complexity turbo decoding scheme for two-way relay network,” in *Proc. International Conference on Wireless Communications and Signal Processing*, 2010, pp. 1–5.

[5] D. To and J. Choi, “Reduced-state decoding in two-way relay networks with physical-layer network coding,” in *Proc. of Information Theory Workshop (ITW)*, 2010, pp. 1–5.

[6] D. Wubben and Y. Lang, “Generalized sum-product algorithm for joint channel decoding and physical-layer network coding in two-way relay systems,” in *Proc. IEEE GLOBECOM*, 2010, pp. 1–5.

[7] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall PTR, 1996.

[8] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo-codes.” in *Proc. IEEE International Conference on Communications*, vol. 2, no. 1, 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.),” *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[10] J. Hokfelt, O. Edfors, and T. Maseng, “A survey on trellis termination alternatives for turbo codes,” in *Proc. IEEE Vehicular Technology Conference*, vol. 3, 1999, pp. 2225–2229.