Convergence Analysis for Channel-coded Physical Layer Network Coding in Gaussian Two-way Relay Channels

Tao Huang*, Tao Yang†, Jinhong Yuan*† and Ingmar Land‡

*School of Electrical Engineering and Telecommunications, University of New South Wales, Australia
†Wireless Networking and Technology Lab, CSIRO ICT Centre, Australia
‡Institute of Telecommunication Research, University of South Australia, Australia

Abstract—This paper considers an irregular repeat-accumulate (IRA) coded physical layer network coding scheme in a Gaussian two-way relay channel. We address the convergence behavior of the iterative receiver of the channel coded PNC at the relay. The proposed method is based on a non-trivial extension of the extrinsic information transfer (EXIT) chart analysis to the PNC scheme with superimposed ternary signal. We develop a new method to model the a priori information for the component decoders in the PNC scheme. This proposed method enables the convergence analysis and thus facilitates the design of channel-coded PNC schemes.

I. INTRODUCTION

Efficient communications over two-way relay channels (TWRCs) have attracted intensive research efforts after the discovery of network coding [1]. Several network coding schemes have been proposed for the relay operation. In particular, an amplify-and-forward based scheme, namely analog network coding [2]–[4], suffers from noise amplification and unnecessary power consumption [5]. A decode-and-forward (DF) based scheme, where the relay completely decodes both users’ messages to compute the network coded (NC) message, suffers from multiplexing loss, as full decoding is demanding and unnecessary in a TWRC [6], [7].

The recent proposed physical layer network coding (PNC) technique [5], [8], [9] outperforms the DF-based scheme. The key idea of the PNC scheme is that the relay directly computes the NC message from its received signal, without complete decoding of both individual messages of the two users. It has been shown in [6], [10] from the information theoretic perspective that, the PNC scheme can achieve within 1/2 bit of the capacity of the Gaussian TWRCs and is asymptotically optimal at a high signal-to-noise ratio (SNR). In [11], the authors have shown that the PNC scheme achieves a higher maximum sum-rate, and a lower sum-bit error rate than the conventional transmission scheme for a number of practical scenarios. The work in [12], [13] investigated the optimization of the modulation constellation with PNC in a TWRC. The work in [14]–[16] proposed several approaches to mitigate the effects brought by the asynchronous transmission in PNC schemes. Several works have addressed how to employ practical binary error control codes in the PNC scheme. In the pioneering work [9], a regular repeat accumulate (RA) code has been utilized, and the NC message is directly computed using a modified belief propagation (BP) algorithm on an equivalent Tanner graph (ETG). In [17], convolutional codes are employed. The NC message is computed with a modified Viterbi algorithm on a reduced trellis to decrease the decoding complexity.

Fig. 1. System model: a PNC scheme in a TWRC, uplink and downlink.

While these works address the utilization of practical codes in the PNC schemes, we are yet to understand how to perform convergence analysis and the code of the PNC scheme. This motivates the work in this paper.

In this paper, we use irregular RA (IRA) codes within channel-coded PNC and study the convergence behavior of the iterative PNC decoder. We generalize the update functions of the variable and check nodes of the IRA code in its ETG. We propose a method to model the extrinsic information transfer (EXIT) of the PNC scheme in the IRA decoder. Based on this model, we analyze the convergence behavior of the PNC scheme. We show that our method can accurately characterize the convergence behavior of the IRA coded PNC scheme.

This work has been supported by the Australian Research Council under the ARC Discovery Grant DP110104995 and DP0986098.
II. SYSTEM MODEL

The system model of the PNC scheme for a Gaussian TWRC is depicted in Fig. 1. Two single-antenna users, denoted by \(A\) and \(B\), exchange information via a single-antenna relay \(R\). The users and the relay operate in a half-duplex mode and there is no direct link between the users. The transmission protocol employs two consecutive time-slots for each round of information exchange. In the first time-slot (uplink), the users transmit simultaneously and the relay remains silent. In the second time-slot (downlink), the relay broadcasts to the two silent users. At each node, the received signal is corrupted by additive white Gaussian noise (AWGN).

Let us first consider the uplink phase. Let \(b_A \in \{0, 1\}^k\) and \(b_B \in \{0, 1\}^k\) denote the length-\(k\) binary message sequences of user \(A\) and \(B\). The message sequence of user \(m, m \in \{A, B\}\), is encoded with a binary channel code, the resultant codeword \(c_m \in \{0, 1\}^n\), where \(n\) represents the length of a codeword. In this paper, we consider the case where the two users employ the same code rate \(R = k/n\). The codewords of the two users are modulated via binary phase shift keying (BPSK), resulting in signal sequences \(x_m = 2c_m - 1 \in \{-1, +1\}^n\). The two users’ signal sequences are simultaneously transmitted to the relay.

We assume that the system has perfect synchronization and power control. Following [5], we assume that the symbols of the two users are received at the relay with the same energy \(E\). The signal sequence received by the relay is then given by

\[
y_R = \sqrt{E}(x_A + x_B) + n_R,\tag{1}
\]

where \(n_R\) is the AWGN vector with variance \(\sigma^2\). The per-user SNR is defined as \(\text{SNR} = E/\sigma^2\).

Following the ideas of network coding [1], the relay needs to compute the \(NC\) message sequence \(b_N = b_A \oplus b_B\) from received signal \(y_R\), where \(\oplus\) denotes modulo-2 addition. The estimated \(NC\) message sequence is denoted by \(\hat{b}_N\).

In the downlink phase, the relay re-encodes the recovered message sequence \(\hat{b}_N\) into a binary codeword \(c_N\). Then, the relay broadcasts the BPSK modulated codeword \(x_R = 2c_N - 1\), to the two users. The signals received by the two users are given by \(y_m = \sqrt{E_R}x_R + n_m\), where \(E_R\) denotes the energy per symbol of the signal received by the users, and \(n_m\) denotes the AWGN sequence at user \(m, m \in \{A, B\}\).

Upon receiving \(y_B\), user \(B\) first decodes the \(NC\) message \(b_N\). If the \(NC\) message \(b_N\) is correctly recovered by both the relay and user \(B\), user \(B\) can correctly recover \(b_A\) as \(b_A = b_B \oplus b_N\), with the knowledge of its own information sequence \(b_B\). The decoding operation at user \(A\) is similar.

In the described two-way relay system, the key issue is to compute the \(NC\) message \(b_N\) at the relay, without decoding of both individual messages of the two users. In this paper, we will only focus on this computation operation, as in [9].

III. COMPUTATION AT RELAY IN THE PNC SCHEME

A. An Equivalent Tanner Graph for an IRA code in PNC

Consider the system described in Section II. In the uplink, an IRA code is utilized to encode the users’ message sequences \(b_A\) and \(b_B\). We assume an IRA code with both irregular variable-node degrees and irregular check-node degrees [18]. Fig. 2 depicts an \textit{equivalent Tanner graph} (ETG) of a PNC scheme with an IRA code. The ETG is constructed by superimposing the individual Tanner graphs of the two users [9].

Define \(b_s = b_A + b_B \in \{0, 1, 2\}^k\), representing a \textit{superimposed message sequence}, where \(+\) denotes addition over real numbers. Define \(c_s = c_A + c_B \in \{0, 1, 2\}^n\), representing a \textit{superimposed codeword}. In the equivalent encoding process associated with the ETG, the \(b_s\) serves as the input to generate the \(c_s\). In the computation process, the relay takes the noisy observation of the superimposed codeword \(c_s\), i.e., the channel observation \(y_s\), to compute \(b_s\). Then, the NC message sequence \(b_N\) is obtained by taking the \(b_s\) modulo-2, i.e., \(b_N(t) = 0\) for \(b_s(t) = 0\), 2, and \(b_N(t) = 1\) otherwise [9].

The structure of the ETG resembles that of the single-user IRA code. However, there are two major differences: 1) The input and output associated with the ETG have three levels, i.e., \(b_s \in \{0, 1, 2\}^k\) and \(c_s \in \{0, 1, 2\}^n\). Thus, every message passed in the ETG represents three probabilities, in contrast to two probabilities in a single-user IRA decoder. 2) The ETG is featuring an \textit{equivalent check node} (CND) equation, denoted by \(f_{\text{CND}}(\cdot)\) and an \textit{equivalent variable node} (VND) equation \(f_{\text{VND}}(\cdot)\), differing from those in the single-user case.

B. Messages for the Ternary Symbols

In the ETG, the nodes exchange messages that describe the ternary superimposed message symbols \(b_s(t) \in \{0, 1, 2\}\). We first introduce messages that directly represent the probability distribution of \(b_s(t)\), i.e., \(p(b_s(t) = 0), p(b_s(t) = 1), p(b_s(t) = 2)\). Later on, we introduce an alternative representation with two log-likelihood ratios (LLRs).

Consider a node in the ETG which has \(M\) edges. The three \textit{a priori} probabilities input to the \(l\)th edge of this node, \(l = 1, \cdots, M\), are written as the vector \(P(l) = [p_0^{(l)}, p_1^{(l)}, p_2^{(l)}]\). The set of these vectors for all edges is denoted by \(P \triangleq \{P(1), \cdots, P(M)\}\). In the iterative computation process, the node takes the \(a\) priori probabilities \(P\) to calculate the output \textit{extrinsic} probabilities, according to its update rule. For the \(l\)th edge, \(l = 1, \cdots, M\), the three output extrinsic probabilities are denoted by \(Q(l) = [q_0^{(l)}, q_1^{(l)}, q_2^{(l)}]\), and the corresponding
set for all edges is denoted by \( Q \triangleq \{ Q(1), \ldots, Q(M) \} \). The update rule can then be written as \( Q = f(P) \).

Initially, the relay calculates the three intrinsic probabilities for the \( j \)th coded bit based on the channel observation \( y_r \). Using \( y'_R = y + 2\sqrt{E} \), the intrinsic probabilities are computed as

\[
\begin{align*}
\pi^\text{CH}(j) = & \left\{ \begin{array}{ll}
\gamma \exp \left( -\frac{(y'_R(j) - i(2\sqrt{E}))^2}{2\sigma^2} \right) & \text{if } i = 0, 2; \\
2\gamma \exp \left( -\frac{(y'_R(j) + i(2\sqrt{E}))^2}{2\sigma^2} \right) & \text{if } i = 1,
\end{array} \right. \\
\end{align*}
\]

where \( \gamma \) is a normalization factor to ensure that \( \pi^\text{CH}(j) + \pi^\text{CH}(j) + \pi^\text{CH}(j) = 1 \). Similar to above, these intrinsic probabilities are collected into the vector \( P^\text{CH}(j) = [\pi^\text{CH}(j), \pi^\text{CH}(j), \pi^\text{CH}(j)] \). The a priori probabilities are initialized as \( P(l) = [1/4, 1/2, 1/4] \) [9]. In the process of computing \( b_s \), the probability vectors are iteratively exchanged between the component nodes in the ETG.

We now define the equivalent messages using LLR. For the \( l \)th edge of a node in the ETG, we define the two LLRs

\[
\Lambda^{(l)}_C \triangleq \log \left( \frac{P^{(l)}_0 + P^{(l)}_2}{P^{(l)}_1} \right) \quad \text{and} \quad \Omega^{(l)}_C \triangleq \log \left( \frac{P^{(l)}_0}{P^{(l)}_2} \right),
\]

(2)

where \( \Lambda \) and \( \Omega \) are equivalent to the probability message \( P \). We refer to \( \Lambda^{(l)}_C \) as the primary LLR and \( \Omega^{(l)}_C \) as the secondary LLR. Similarly, the primary and secondary LLRs associated with the output extrinsic probabilities are defined as

\[
\Lambda^{(l)}_Q \triangleq \log \left( \frac{q^{(l)}_0 + q^{(l)}_2}{q^{(l)}_1} \right) \quad \text{and} \quad \Omega^{(l)}_Q \triangleq \log \left( \frac{q^{(l)}_0}{q^{(l)}_2} \right).
\]

(3)

Next, we investigate the update rules for CND and VND in terms of the LLRs \( \Lambda \) and \( \Omega \).

C. Update Rules

Let us first consider a CND with degree \( d_c = 2 \). Assume that we are using the a priori information from the first and second edge to update the extrinsic information of the third edge. The update rule for this CND can be expressed as

\[
\Lambda^{(3)}_C = \log \left( \frac{1 + \exp \left( \Lambda^{(1)}_C \right) \exp \left( \Lambda^{(2)}_C \right)}{\exp \left( \Lambda^{(1)}_C \right) + \exp \left( \Lambda^{(2)}_C \right)} \right),
\]

(4)

\[
\Omega^{(3)}_C = \log \left( \frac{1 + \exp \left( \Omega^{(1)}_C \right) \exp \left( \Omega^{(2)}_C \right) + K_{\text{CND}}}{\exp \left( \Omega^{(1)}_C \right) + \exp \left( \Omega^{(2)}_C \right) + K_{\text{CND}}} \right),
\]

(5)

where

\[
K_{\text{CND}} = \frac{1 + \exp \left( \Omega^{(1)}_C \right)}{2} \left[ \frac{1 + \exp \left( \Omega^{(2)}_C \right)}{\exp \left( \Lambda^{(1)}_C \right) \exp \left( \Lambda^{(2)}_C \right)} \right].
\]

In general, for a CND with degree \( d_c > 2 \), the update function \( f^{CND}_{\text{CND}}(\cdot) \) can be carried out by successively utilizing the degree 2 CND update rule, given by

\[
[\Lambda, \Omega]^{(1)}_C = f^{CND}_{\text{CND}}(\Lambda, \Omega)^{(1)}_P, \quad [\Lambda, \Omega]^{(2)}_C = f^{CND}_{\text{CND}}(\Lambda, \Omega)^{(2)}_P, \quad \vdots
\]

\[
[\Lambda, \Omega]^{(M)}_C = f^{CND}_{\text{CND}}(\Lambda, \Omega)^{(M-3)}_P, \quad [\Lambda, \Omega]^{(M-1)}_C = f^{CND}_{\text{CND}}(\Lambda, \Omega)^{(M-1)}_P.
\]

(6)

We refer to the above approach as a successive update (SU).

The update rule for a VND with degree \( d_v \) can be written as

\[
\Lambda^{(l)}_Q = \left( d_v - 2 \right) \log 2 + \sum_{l'=1, l' \neq l} \Lambda^{(l')}_{P} + K_{\text{VND}},
\]

(7)

where

\[
K_{\text{VND}} = \log \left( \frac{1 + \prod_{l'=1, l' \neq l} \exp \left( \Omega^{(l')}_{P} \right)}{\prod_{l'=1, l' \neq l} \left( 1 + \exp \left( \Omega^{(l')}_{P} \right) \right)} \right).
\]

IV. CONVERGENCE BEHAVIOR ANALYSIS

A. Modeling of a priori information

To carry out convergence behavior analysis, we partition the ETG of the PNC scheme into two parts: an inner component code consisting of the combined CNDs and ACC, and an outer component code consisting of the VNDs. The idea of the EXIT techniques is to predict the behavior of the iterative process by solely looking at the input/output mutual information of the two individual code components of the PNC scheme.

Numerical results show the following observations: 1) the probability density function (PDF) of the primary LLR \( \Lambda \) approaches a Gaussian-like distribution with its mean equal to half of its variance with iterations; 2) the PDF of the secondary LLR \( \Omega \) behaves like a combination of a Gaussian-like distribution and an impulse. In particular, observation 2) makes the analytical treatment of the EXIT functions difficult. In order to tackle this problem, we now present two models for the a priori input, which allow for bound the EXIT curves.

Model I: The a priori primary LLR \( \Lambda \) of the NC information is modelled as

\[
\Lambda = \frac{\sigma^2}{2} (1 - 2b_N) + n_A,
\]

(8)

where \( n_A \) is a Gaussian random variable, and \( \sigma^2 \) is its variance [19]. In a addition, we assume that we know perfectly the a priori secondary LLR \( \Omega \), given by

\[
\Omega = \begin{cases} 
+\infty & \text{if } b_s = 0; \\
0 & \text{if } b_s = 1; \\
-\infty & \text{if } b_s = 2.
\end{cases}
\]

(9)
This model will lead to an upper bound for the EXIT curve, since the actual decoding process does not have perfect a priori secondary LLR $\Omega_P$.

Model II: The a priori primary LLR $\Lambda_P$ is the same as in Model I. However, the a priori secondary LLR $\Omega_P$ is assumed to be zero, i.e., $\Omega_P = 0$. This model will lead to a lower bound for the EXIT curve, as the actual decoding retains certain input a priori secondary LLR $\Omega_P$.

B. Extrinsic Information Transfer Function

We now consider the EXIT functions generated by employing the two models above. To measure the NC information contents of the input a priori knowledge, mutual information between the NC bits and primary LLR will be $I_A = I(b_N; \Lambda_P)$. Similarly, mutual information $I_E = I(b_N; \Lambda_Q)$ is used to quantify the NC information contents of the output extrinsic information. For the inner CND-ACC decode, the EXIT transfer function is denoted as $I_E = T_{\text{Inner}}(I_A, \Omega_P, d_v, E_b/N_0)$. The EXIT transfer function for the outer VND decoder is denoted as $I_E = T_{\text{Outer}}(I_A, \Omega_P, d_v)$. The EXIT curves for the inner decoder with CND $d_c = 1, 2, 3, 4, 5$ for a given SNR are shown in Fig. 3. Model I generates higher extrinsic information than Model II, as Model I contains the contributions from the secondary LLR $\Omega_P$. In addition, we observe that this performance gain due to the availability of the secondary LLR $\Omega_P$ diminishes with increasing CND degree $d_c$. This suggests that, when CND-ACC decoder has a large CND degree $d_c$, Model I and Model II generate similar distributions of the ternary symbols in its output extrinsic information. Thus, we can use Model II to analyze the distribution of the extrinsic output of the ternary symbols for the CND-ACC decoder. Using the CND update rule Eq. (4) and Eq. (5) and successive update approach, we are able to verify that Model II generates all secondary LLR $\Omega_Q = 0$ for the CND-ACC decoder. This is important for understanding the VND decoder behavior and will be discussed next.

The EXIT curves for the outer decoder with VND $d_v = 2, 3, 4$ are shown in Fig. 4. The performance gain between the two models is contributed from the $K_{\text{VND}}$ in (6). As the VND degree $d_v$ increases, the performance gap becomes larger, which is different than the CND-ACC decoder.

It is known from [20] that, for a single-input single-output (SISO) AWGN channel, a capacity achieving IRA code typically has a relatively large average degree $\bar{d}_c$, and thus $\bar{d}_v > 2$. Therefore, from the observation of Fig. 3, we can use either Model I and Model II for the CN-ACC decoder in the code design. Note that the extrinsic information output from the CND-ACC decoder is the a priori information input to the VND decoder. Previous analysis of the CND-ACC decoder suggests that, with large CND degree, its output extrinsic information has most of its secondary LLR $\Omega_Q = 0$. The suggests that the input to the VND decoder with its secondary LLR most likely to be zero. For VND decoder, it is more appropriate to use Model II to value its performance. We will show in next section with two examples of IRA codes, and compare its decoding trajectory with EXIT functions using Model II. In code design, using Model II has an additional advantage over using Model I. Since Model II leads to a lower bound on the EXIT functions, the iterative decoding of an optimized code designed base on Model II will converge.

V. NUMERICAL RESULTS

In Fig. 5, we plot the EXIT functions obtained by using the Model II and the actual decoding trajectory obtained from simulation, where the code rate is $R = 1/3$. This IRA code has an average CND degree $\bar{d}_c = 2.4$ and an average VND degree $\bar{d}_v = 7.2$. We observe that using the developed Model II, the EXIT functions of the component decoders of the IRA code can be accurately characterized. This suggests that the actual EXIT function of the VND decoder in an IRA coded PNC scheme are more close to the lower bound (Model II). In Fig. 6, we plot the EXIT functions obtained by using the Model II
and the actual decoding trajectory obtained from simulation, where the code rate is \( R = 3/4 \). This IRA code has an average CND degree \( \bar{d}_c = 4.2 \) and an average VND degree \( d_v = 5.6 \). Again, the decoding trajectory matches well with the EXIT functions of Model II.

VI. CONCLUSION

In this paper, we developed a method to model the a priori information for the component decoders in the PNC scheme in a Gaussian TWRC. We have shown that the proposed model is able to accurately characterize the convergence behavior of the NC information in an ETG with ternary symbols. The developed convergence analysis method provides a new framework for optimization of capacity approaching codes in PNC schemes.

REFERENCES


