Characterization and Optimization of LDPC codes for the 2-user Gaussian Multiple Access Channel

Aline Roumy[*], David Declercq[†]

June 4, 2007

Abstract

In this paper we address the problem of designing good LDPC codes for the Gaussian multiple access channel (MAC). The framework we choose is to design multiuser LDPC codes with joint Belief Propagation decoding on the joint graph of the 2-user case. Our main result compared to existing work is to express analytically EXIT functions of the multiuser decoder with two different approximations of the Density Evolution. This allows us to propose a very simple linear programming optimization for the complicated problem of LDPC code design with joint multiuser decoding. The stability condition for our case is derived and used in the optimization constraints. The codes that we obtain for the 2-user case are quite good for various rates, especially if we consider the very simple optimization procedure.

Introduction

In this paper we address the problem of designing good LDPC codes for the Gaussian multiple access channel (MAC). The corner points of the capacity region have long been known to be achievable by single user decoding. This idea was also used to achieve any point of the capacity region by means of rate splitting [13]. Here we focus on the design of multiuser codes since the key idea for achieving any point in the capacity region of the Gaussian MAC is random coding and optimal joint decoding [1, 8]. A suboptimal but practical approach consists in using irregular Low-Density Parity-Check codes (LDPC) decoded with Belief Propagation (BP) [9, 2, 5]. In this paper we aim at proposing low complexity LDPC code design methods for the 2-user MAC where joint decoding is performed with belief propagation decoder (BP).

Here as in [2], we tackle the difficult and important problem where all users have the same power constraint and the same rate in order to show that the designed multiuser codes can get close to any

[*] INRIA, Rennes, France; `aline.roumy@irisa.fr`

[†] ETIS, Cergy-Pontoise, France; `declercq@ensea.fr`

point of the boundary in the capacity region of the Gaussian MAC. We propose two optimization approaches based on two different approximations of Density Evolution (DE) in the 2-user MAC factor graph: the first is the Gaussian Approximation (GA) of the messages, and the second is an *erasure channel* (EC) approximation of the messages. These two approximations, together with constraints specific to the multiuser case, lead to very simple LDPC optimization problems, solved by linear programming. The paper is organized as follows: in section 1, we present the MAC factor graph and the notations used for the LDPC optimization. In section 2, we describe our approximations of the mutual information evolution through the central function node, that we call *state check node.* A practical optimization algorithm is presented in section 3 and finally, we report in section 4 the thresholds of the optimized codes computed with density evolution and we plot some finite length performance curves.

1 2-user MAC factor graph and decoding algorithm

In a 2-user Gaussian MAC, we consider 2 independent users $x^{[1]}$ and $x^{[2]}$, being sent to a single receiver. Each user is LDPC encoded by different irregular LDPC codes[1] with codeword length N, and their respective received power will be denoted σ_i^2. The codeword is BPSK modulated and the synchronous discrete model of the transmission at time n is given by, $\forall 0 \leq n \leq N-1$

$$y_n = \sigma_1 x_n^{[1]} + \sigma_2 x_n^{[2]} + w_n = \left[\begin{array}{cc} \sigma_1 & \sigma_2 \end{array} \right] .Z_n + w_n \qquad (1)$$

Throughout the paper, neither the flat fading nor the multipath fading effect is taken into account. More precisely, we will consider the *equal rate/equal power* 2-user MAC channel, that is $R_1 = R_2 = R$ and $\sigma_1^2 = \sigma_2^2 = 1$. The equal receive power channel can be encountered in practice, for example, if power allocation is performed at the transmitter side. In equation (1), $Z_n = [x_n^{[1]}, x_n^{[2]}]^T$ is the *state vector* of the multiuser channel, and w_n is a zero mean additive white Gaussian noise with variance σ^2: its probability density function (p.d.f.) is denoted $\mathcal{N}(0, \sigma^2)$.

In order to jointly decode the two users, we will consider the factor graph [7] of the whole multiuser system, and run several iterations of BP [10]. The factor graph of the 2-user LDPC-MAC is composed of the 2 LDPC graphs,[2] which are connected through function nodes representing the link between the state vector Z_n and the coded symbols of each user $x_n^{[1]}$ and $x_n^{[2]}$. We will call this node the *state check node.* Figure 1 shows the state check node neighborhood and the messages on the edges that are updated during a decoding iteration.

In the following, the nodes of each individual LDPC graph are referred to as *variable nodes* and *check nodes.* Let $m_{ab}^{[k]}$ denotes the message from node a to node b for user k, where (a, b) can either

[1] The LDPC codes could however belong to the same code ensemble.

[2] an LDPC graph denotes the Tanner graph [15] that represents an LDPC code.

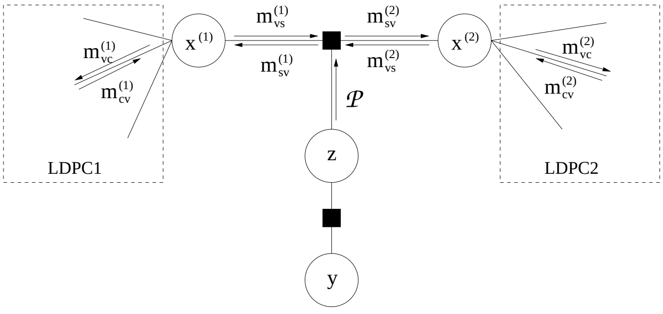

Figure 1: Factor graph of the 2-user synchronous MAC channel: zoom around the state check node neighborhood

be v for variable node, c for check node or s for state check node.

From now on and as indicated on Figure 1, we will drop the time index n in the equations. All messages in the graph are given in log-density ratio form $\log\left(\frac{p(.|x^{[i]}=+1)}{p(.|x^{[i]}=-1)}\right)$, except for the probability message $\mathcal{P}$ coming from the channel observation y. $\mathcal{P}$ is a vector composed of four probability messages given by:

$$\mathcal{P} = \begin{bmatrix} \mathcal{P}_{00} \\ \mathcal{P}_{01} \\ \mathcal{P}_{10} \\ \mathcal{P}_{11} \end{bmatrix} = \begin{bmatrix} p(y|Z = [+1 \ \ +1]^T) \\ p(y|Z = [+1 \ \ -1]^T) \\ p(y|Z = [-1 \ \ +1]^T) \\ p(y|Z = [-1 \ \ -1]^T) \end{bmatrix} \tag{2}$$

Since for the equal power case $p(y|Z = [-1 \ \ +1]^T) = \mathcal{P}_{10} = \mathcal{P}_{01} = p(y|Z = [+1 \ \ -1]^T)$, the likelihood message $\mathcal{P}$ is completely defined by only three values.

At initialization, the log-likelihoods are computed from the channel observations y. The message update rules for all messages in the graph $(m^{[i]}_{cv}, m^{[i]}_{vc}, m^{[i]}_{vs})$ follow from usual LDPC BP decoding [7, 11]. We still need to give the update rule through the state check node to complete the decoding algorithm description. The message $m^{[i]}_{sv}$ at the output of the state check node is computed from $m^{[j]}_{sv}$ for $(i, j) \in \{(1, 2), (2, 1)\}$ and $\mathcal{P}$:

$$\left\{ \begin{array}{lcl} m^{[1]}_{sv} & = & \log \frac{\mathcal{P}_{00} e^{m^{[2]}_{vs}} + \mathcal{P}_{01}}{\mathcal{P}_{10} e^{m^{[2]}_{vs}} + \mathcal{P}_{11}} \\ m^{[2]}_{sv} & = & \log \frac{\mathcal{P}_{00} e^{m^{[1]}_{vs}} + \mathcal{P}_{10}}{\mathcal{P}_{01} e^{m^{[1]}_{vs}} + \mathcal{P}_{11}} \end{array} \right. \tag{3}$$

The channel noise is Gaussian $\mathcal{N}(0, \sigma^2)$, and (3) can be rewritten for the equal power case as

$$m^{[i]}_{sv} = \log \frac{e^{\frac{2y-2}{\sigma^2}} e^{m^{[j]}_{vs}} + 1}{e^{m^{[j]}_{vs}} + e^{\frac{-2y-2}{\sigma^2}}} \tag{4}$$

where the distribution of y is a mixture of Gaussian distributions $y \sim \frac{1}{4}\mathcal{N}(2, \sigma^2) + \frac{1}{2}\mathcal{N}(0, \sigma^2) +$

$\frac{1}{4}\mathcal{N}(-2, \sigma^2)$ since the channel conditionnal distributions are:

$$\begin{cases} y|_{(+1,+1)} \sim \mathcal{N}(2, \sigma^2) \\ y|_{(+1,-1)} \sim \mathcal{N}(0, \sigma^2) \\ y|_{(-1,+1)} \sim \mathcal{N}(0, \sigma^2) \\ y|_{(-1,-1)} \sim \mathcal{N}(-2, \sigma^2) \end{cases} \tag{5}$$

Now that we have stated all the message update rules within the whole graph, we need to indicate in which order the message computation are performed. We shall consider in this work two differents schedulings:

- serial scheduling. A *decoding iteration* for a given user (or "round" [11]) consists in activating all the variable nodes, and thus sending information to the check nodes, activating all the check nodes and all the variable nodes again that now send information to the state check nodes, and finally activating all the state check nodes that send information to the next user. Once this iteration for one user is completed, a new iteration can be performed for the second user. In a serial scheduling, a decoding round for user two is not performed until a decoding round for user one is completed.
- parallel scheduling. In a *parallel scheduling*, the decoding rounds (for the two users) are activated simultaneously (in parallel).

2 Mutual information evolution through the state check node

The DE is a general tool that aims to predict the asymptotical and average behavior of LDPC codes or more general graphs decoded with BP. However, DE is computationally intensive and in order to reduce the computational burden of LDPC codes optimization, faster techniques have been proposed, based on the approximations of DE by a one-dimensional dynamical system (see [16, 14] and reference therein). This is equivalent to considering that the true density of the messages is mapped onto a single parameter, and tracking the evolution of this parameter along the decoding iterations.

It is also known that an accurate single parameter is the mutual information between the variables associated with the variable nodes and their messages [16, 14]. The mutual information evolution describes each computation node in BP-decoding by a *mutual information transfer function*, which is usually referred to as the EXtrinsic mutual Information Transfer (EXIT) function. For parity-check codes with binary variables only (as for LDPCs or Irregular Repeat Accumulate codes), the EXIT charts can be expressed analytically [14], leading to very fast and powerful optimization algorithms.

In this section, we will express analytically the EXIT chart of the state check node update, based on two different approximations. First, we will express a symmetry property for the state check node, then we will present a Gaussian Approximation (GA) of the messages densities, and finally we will consider that the messages are the output of an erasure channel (EC).

Similarly to the definition of the messages (see Section 1), we will denote x_{ab} the mutual information from node a to node b, where (a, b) can either be v for variable node, c for check node or s for state check node.

2.1 Symmetry property

First of all, let us present one of the main differences between the single-user case and the 2-user case. For the single user, memoryless, binary-input and symmetric-output channel, the transmission of the all-one BPSK sequence is assumed in the DE. The generalization of this property for non-symmetric channels is not trivial and some authors have recently addressed this question [3, 18].

In the 2-user case, the channel seen by each user is not symmetric since it depends on the other users decoding. However, the asymmetry of the 2-user MAC channel is very specific and much simpler to deal with than the general case. We proceed as explained below.

Let us denote $\Psi_S(y, m)$ the state check node map of the BP decoder, that is the equation that takes an input message m from one user and the observation y and computes the output message that is sent to the second user. The symmetry condition of a state check node map is defined as:

Definition 1. State check node symmetry condition: the state check node update rule is said to be symmetric if sign inversion invariance holds i.e.

$$\Psi_S(-y, -m) = -\Psi_S(y, m).$$

Note that the update rule defined in (4) is symmetric.

In order to state a symmetry property for the state check node, we further need to define some symmetry conditions for the channel and the messages passed by in the BP decoder.

Definition 2. Symmetry conditions for the channel observation. A 2-user MAC is output-symmetric if its observation y verifies

$$p(y_t | x_t^{[k]}, x_t^{[j]}) = p(-y_t | -x_t^{[k]}, -x_t^{[j]})$$

where y_t is the observation at time index t and $x_t^{[k]}$ is the t^{th} element of the codeword sent by user k. Note that this condition holds for the 2-user Gaussian MAC.

Definition 3. Symmetry conditions for messages. A message is symmetric if

$$p(m_t | x_t) = p(-m_t | -x_t)$$

where m_t is a message at time index t and x_t is the variable that is estimated by message m_t.

Proposition 1. Consider a state check node. Let us assume a symmetric channel observation, the entire average behavior of the state check node can be predicted from its behavior assuming transmission of the all-one BPSK sequence for the output user and a sequence with half symbols fixed at '1' and half symbols at '-1' for the input user.

Proof. See Appendix B. □

2.2 Gaussian approximation of the state check messages (GA)

The first approximation of the DE through the state check node considered in this work assumes that the input message m_{vs} is Gaussian with density $\mathcal{N}(\mu_{vs}, 2\mu_{vs})$, and that the output message m_{sv} is a mixture of two Gaussian densities with means $\mu_{sv}|_{(+1,+1)}$ and $\mu_{sv}|_{(+1,-1)}$, and variances equal to twice the means. The state check node update rule is symmetric and thus we omit the user index in the notations.

Hence by noticing that m_{sv} in (4) can be rewritten as the sum of three functions of Gaussian distributed random variables:

$$m_{sv} = -m_{vs} + \log\left(1 + e^{m_{vs} + \frac{2y-2}{\sigma^2}}\right) - \log\left(1 + e^{-m_{vs} - \frac{2y+2}{\sigma^2}}\right) \tag{6}$$

we get the output means:

$$\begin{cases} \mu_{sv}|_{(+1,+1)} = F_{+1,+1}\left(\mu_{vs}, \sigma^2\right) \\ \mu_{sv}|_{(+1,-1)} = F_{+1,-1}\left(\mu_{vs}, \sigma^2\right) \end{cases}$$

where

$$\begin{cases} F_{+1,+1}\left(\mu, \sigma^2\right) = \dfrac{1}{\sqrt{\pi}} \displaystyle\int_{-\infty}^{+\infty} e^{-z^2} \log\left(\dfrac{1 + e^{-2\sqrt{\mu + \frac{2}{\sigma^2}}z + \mu + \frac{2}{\sigma^2}}}{1 + e^{-2\sqrt{\mu + \frac{2}{\sigma^2}}z - \mu - \frac{6}{\sigma^2}}}\right) dz - \mu \\ F_{+1,-1}\left(\mu, \sigma^2\right) = \dfrac{1}{\sqrt{\pi}} \displaystyle\int_{-\infty}^{+\infty} e^{-z^2} \log\left(\dfrac{1 + e^{-2\sqrt{\mu + \frac{2}{\sigma^2}}z - \mu - \frac{2}{\sigma^2}}}{1 + e^{-2\sqrt{\mu + \frac{2}{\sigma^2}}z + \mu - \frac{2}{\sigma^2}}}\right) dz + \mu \end{cases} \tag{7}$$

The detailed computation of these functions is reported in Appendix A. Note that these expressions need to be accurately implemented with functionnal approximations in order to be used efficiently in an optimization procedure.

As mentioned earlier, it is desirable to follow the evolution of the mutual information as single paramater, so we make use of the usual function that relates the mean and the mutual information: for a message m with conditional pdf $m|x = 1 \sim \mathcal{N}(\mu, 2\mu)$, and $m|x = -1 \sim \mathcal{N}(-\mu, 2\mu)$ the mutual information is $I(x; m) = J(\mu)$ where

$$J(\mu) = 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2\left(1 + e^{-2\sqrt{\mu}z - \mu}\right) dz \tag{8}$$

Note that $J(\mu)$ is the capacity of a binary-antipodal input additive white Gaussian channel (BI-AWGNC) with variance $2/\mu$.

Now that we have expressed the evolution of the mean of the messages when they are assumed Gaussian, we make use of the function $J(\mu)$ (8) in order to give the evolution of the mutual information through the state check node under Gaussian approximation. This corresponds exactely to the EXIT chart [16] of the state check node update:

$$\begin{cases} x_{sv}|_{(+1,+1)} = J\left(F_{+1,+1}\left(J^{-1}(x_{vs}), \sigma^2\right)\right) \\ x_{sv}|_{(+1,-1)} = J\left(F_{+1,-1}\left(J^{-1}(x_{vs}), \sigma^2\right)\right) \end{cases}$$

It follows

$$x_{sv} = \frac{1}{2} x_{sv}|_{(+1,+1)} + \frac{1}{2} x_{sv}|_{(+1,-1)} = \frac{1}{2} J\left(F_{+1,+1}\left(J^{-1}(x_{vs}), \sigma^2\right)\right) + \frac{1}{2} J\left(F_{+1,-1}\left(J^{-1}(x_{vs}), \sigma^2\right)\right) \quad (9)$$

2.3 Erasure channel approximation of the state check messages (EC)

This approximation assumes that the distribution of the messages at the state check node input (m_{vs} see figure 1) is the output of a binary erasure channel (BEC). Thus when the symbol +1 is sent, the LLR distribution consists of two mass points, one at zero and the other at $+\infty$. Let us denote δ_x, a mass point at x. It follows that the LLR distribution when the symbol +1 is sent is:

$$\mathcal{E}_{+}(\epsilon) \triangleq \epsilon\delta_0 + (1-\epsilon)\delta_{\infty}$$

Similarly, when -1 is sent, the LLR distribution is $\mathcal{E}_{-}(\epsilon) \triangleq \epsilon\delta_0 + (1-\epsilon)\delta_{-\infty}$. The mutual information associated with these distributions is the capacity of a BEC:

$$\widetilde{x} = 1 - \epsilon. \quad (10)$$

The distribution of channel observation y is not consistent with the approximation presented here since y is the output of a ternary input additive white Gaussian channel (TIAWGNC) with input distribution $\frac{1}{4}\delta_{-2} + \frac{1}{2}\delta_0 + \frac{1}{4}\delta_2$ (because of the symmetry property, see Section 2.1) and variance σ^2. The capacity of such a channel is:

$$\begin{aligned} &\mathcal{C}_{TIAWGNC}(\mu) \triangleq \frac{3}{2} \\ &-\frac{1}{2\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2\left(1 + \frac{1}{2}e^{2\sqrt{\mu}z-\mu} + \frac{1}{2}e^{-2\sqrt{\mu}z-\mu}\right) dz \\ &-\frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2\left(1 + e^{-2\sqrt{\mu}z-\mu}\right) dz, \end{aligned} \quad (11)$$

with $\mu = 2/\sigma^2$.

In order to use coherent hypothesis in the erasure approximation of the state check node, the real channel is mapped onto an erasure channel with same capacity. The ternary erasure channel (TEC) used for the approximation has input distribution $\frac{1}{4}\delta_{-2} + \frac{1}{2}\delta_0 + \frac{1}{4}\delta_2$ and erasure probability p. The capacity of such a TEC is:

$$\mathcal{C}_{TEC} = \frac{3}{2}(1-p) \quad (12)$$

Therefore the true channel with capacity $\mathcal{C}_{TIAWGNC}$ will be approximated by a TEC with erasure probability $p = 1 - \frac{2}{3}\mathcal{C}_{TIAWGNC}$.

Because of the symmetry property (see Section 2.1), we consider only two cases:

- Under the $(+1, +1)$-hypothesis and by definition of the erasure channel, the observation y is either an erasure with probability (w.p.) p or $y = 2$ w.p. $(1 - p)$. The input message corresponds to the symbol $+1$ and its distribution is $\mathcal{E}_+(\epsilon)$. The output message corresponds to the symbol $+1$ and by applying (3), we obtain the output distribution $m_{sv}|_{(+1,+1)} \sim \mathcal{E}_+(p)$.

- Under the $(+1, -1)$-hypothesis, the observation of the erasure channel y is either an erasure w.p. p or $y = 0$ w.p. $(1 - p)$. The input message corresponds to the symbol -1 and its distribution is $\mathcal{E}_-(\epsilon)$. The output message corresponds to the symbol $+1$ and by applying (3), we obtain the output distribution $m_{sv}|_{(+1,-1)} \sim \mathcal{E}_+\Big(1 - (1-p)(1-\epsilon)\Big)$.

By applying (10), (12) and the assumption $\mathcal{C}_{TIAWGNC} = \mathcal{C}_{TEC}$, the mutual information transfer function through the state check node is thus

$$\begin{cases} x_{sv}|_{(+1,+1)} = \dfrac{2}{3}\,\mathcal{C}_{TIAWGNC} \\ \\ x_{sv}|_{(+1,-1)} = \dfrac{2}{3}\,x_{vs}\,\mathcal{C}_{TIAWGNC} \end{cases}$$

it follows

$$x_{sv} = \frac{1}{2}x_{sv}|_{(+1,+1)} + \frac{1}{2}x_{sv}|_{(+1,-1)} = \frac{1}{3}\,(1 + x_{vs})\,\mathcal{C}_{TIAWGNC} \tag{13}$$

In Figure 2, we compare the two approximations for the state node EXIT function with equations (9) and (13), for three different signal to noise ratios. The solid lines show the GA approximation whereas the broken lines show the EC approximation. We have also indicated with *plus* signs the mutual information obtained with Monte Carlo simulations. Our numerical results show that, the Gaussian a priori GA approximation is more attractive since the mutual information computed under this assumption have the smallest gap to the exact mutual information (Monte Carlo simulation without any approximation).

3 LDPC code optimization

Using the EXIT charts for the LDPC codes ([4, 14]) and for the state check node under the two considered approximations (9), (13), we are now able to give the evolution of the mutual information x along a whole 2-user decoding iteration. The irregularity of the LDPC code is defined as usual by the degree sequences $(\{\lambda_i\}_{i=2}^{d_v}, \{\rho_j\}_{j=2}^{d_c})$, that represent the fraction of edges connected to variable nodes (resp., check nodes) of degree i (resp., j). As in the single user case, we wish to have an

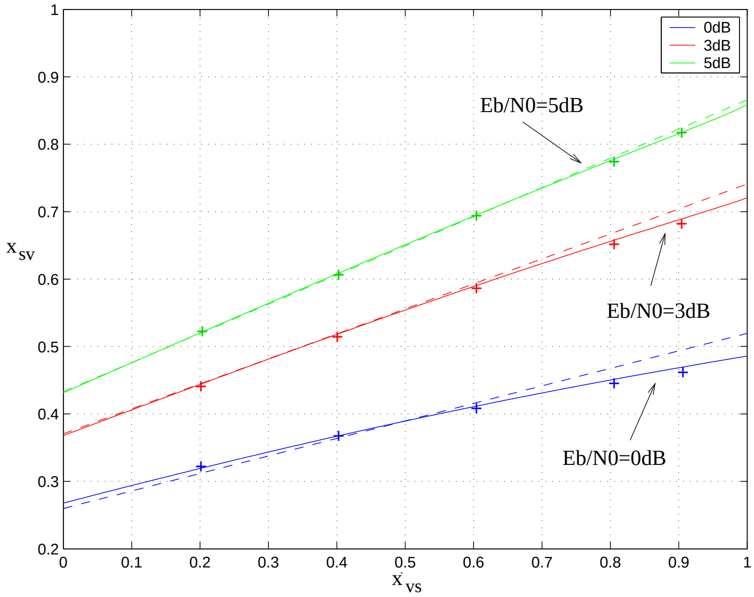

Figure 2: Mutual information evolution at the state check node. Comparison of the approximation methods with the exact mutual information at the state check node. The solid lines represent the GA approximation, the broken lines the EC approximation, and *plus* signs show Monte Carlo simulations.

optimization algorithm that could be solved quickly and efficiently using linear programming. In order to do so, we must make different assumptions that are mandatory to ensure that the evolution of the mutual information is *linear* in the parameters $\{\lambda_i\}$:

$\{\mathcal{H}_0\}$: hypothesis *equal LDPC codes.* Under this hypothesis, we assume that the 2 LDPC codes belong to the same ensemble $(\{\lambda_i\}_{i=2}^{d_v}, \{\rho_j\}_{j=2}^{d_c})$.

$\{\mathcal{H}_1\}$: hypothesis *without interleaver.* Under this hypothesis, each and every state check node is connected to two variable nodes (one in each LDPC code) having *exactly* the same degree.

Proposition 2. Under hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, the evolution of the mutual information x_{vc} at the $l-th$ iteration under the parallel scheduling described in section 1 is linear in the parameters $\{\lambda_i\}$.

Proof. See Appendix C. □

From Proposition 2, we can now write the evolution of the mutual information for the entire graph. More precisely, by using (8), (9), and (13), we finally obtain equations (14) for the Gaussian approximation and (15) for the erasure channel approximation:

$$\begin{aligned} x_{vc}^{(l)} &= \sum_{i=2}^{d_v} \lambda_i J \left\{ J^{-1} \left[\frac{1}{2} J \left(F_{+1,+1} \left\{ i J^{-1} \left[\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) \right], \sigma^2 \right\} \right) \right. \right. \\ &\qquad \left. \left. + \frac{1}{2} J \left(F_{+1,+1} \left\{ i J^{-1} \left[\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) \right], \sigma^2 \right\} \right) \right] + (i-1) J^{-1} \left[\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) \right] \right\} \\ &\triangleq F_{GA} \left(\{\lambda_i\}, x_{vc}^{(l-1)}, \sigma^2 \right) \end{aligned} \tag{14}$$

$$\begin{aligned} x_{vc}^{(l)} &= \sum_{i=2}^{d_v} \lambda_i J \left\{ J^{-1} \left[\frac{\mathcal{C}_{TIAWGNC}}{3} \left(1 + J \left\{ i J^{-1} \left[\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) \right] \right\} \right) \right] + (i-1) J^{-1} \left[\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) \right] \right\} \\ &\triangleq F_{EC} \left(\{\lambda_i\}, x_{vc}^{(l-1)}, \sigma^2 \right) \end{aligned} \tag{15}$$

with

$$\boldsymbol{\rho} \left(x_{vc}^{(l-1)} \right) = 1 - \sum_{j=2}^{d_c} \rho_j J \left((j-1)\, J^{-1} (1 - x_{vc}^{(l-1)}) \right)$$

It is interesting to note that, in the equations (14) and (15), the evolution of the mutual information is indeed linear in the parameters $\{\lambda_i\}$, when $\{\rho_j\}$ are fixed.

As often presented in the literature, we will only optimize the data node parameters $\{\lambda_i\}$, for a fixed (carefully chosen) check node degree distribution $\{\rho_j\}$. The optimization criterion is to maximize R subject to a vanishing Bit Error Rate. The **optimization problem** can be written, for a given σ^2 and a given $\rho(x)$ as:

$$\text{maximize } \sum_{i=2}^{d_v} \lambda_i / i \qquad\qquad \text{subject to} \tag{16}$$

$$\begin{array}{ll} (\mathcal{C}_1)\ \displaystyle\sum_{i=2}^{d_v} \lambda_i = 1 & \text{[mixing constraint]} \\ (\mathcal{C}_2)\ \lambda_i \in [0,1] & \text{[proportion constraint]} \\ (\mathcal{C}_3)\ \lambda_2 < \exp\left(\dfrac{1}{2\sigma^2} \right) / \displaystyle\sum_{j=2}^{d_c} (j-1)\rho_j & \text{[stability constraint]} \\ (\mathcal{C}_4)\ F(\{\lambda_i\}, x, \sigma^2) > x,\ \forall x \in [0,1[& \text{[convergence constraint]} \end{array}$$

where $(\mathcal{C}_3)$ is the condition for the fixed point to be stable (see proposition 3. below) and where $(\mathcal{C}_4)$ corresponds to the convergence to the stable fixed-point $x = 1$, which corresponds to zero error rate constraint.

Solution to the optimization problem: For a given σ^2 and a given $\rho(x)$, the cost function and the constraints $(\mathcal{C}_1)$, $(\mathcal{C}_2)$ and $(\mathcal{C}_3)$ are linear in the parameters $\{\lambda_i\}$. The function used in constraint $(\mathcal{C}_4)$ is either (14) or (15) which are both linear in the parameters $\{\lambda_i\}$. The optimization problem can then be solved for a given $\rho(x)$ by linear programming. We would like to emphasize the fact that the hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ are necessary to have a linear problem, which is the key feature of quick and efficient LDPC optimization.

These remarks allow us to propose an algorithm that solves the optimization problem (16) in the class of functions $\rho(x)$ of the type $\rho(x) = x^n, \forall n > 0$.

- First, we fix a target SNR (or equivalently σ^2).
- Then, for each $n > 0$, $\rho(x) = x^n$ and we perform a linear programming in order to find a set of parameters $\{\lambda_i\}$ that maximizes the rate under the constraints $(\mathcal{C}_1)$ to $(\mathcal{C}_4)$ (16). In order to integrate the $(\mathcal{C}_4)$ constraint in the algorithm, we quantize x. For each quantized value of x, the equation in $(\mathcal{C}_4)$ leads to an additional constraint. Hence, for each n, we get a rate.
- Finally, we choose n, that maximizes the rate (over all n).

In practice, the search over all possible n is performed up to a maximal value. This is to insure that the graph remains sparse.

Stability of the solution. Finally, the stability condition of the fixed point for the 2-user MAC channel is given in the following proposition.

Proposition 3. The local stability condition of the DE for the 2-user Gaussian MAC is the same as that of the single user case:

$$\lambda_2 < \exp\left(\frac{1}{2\sigma^2}\right) / \sum_{j=2}^{d_c} (j-1)\rho_j$$

The proof is given in Appendix D.

4 Results

In this section we present results for codes designed according to the two methods presented in section 2, for rates from 0.3 to 0.6, and we compare the methods on the basis of the true thresholds obtained by DE and finite length simulations.

Table 1 shows the performance of LDPC codes optimized with the Gaussian approximation. Table 2 shows the performance of LDPC codes designed according to the Erasure channel approximation. In both tables are indicated the code rate, the check nodes degrees $\rho(x) = \sum_{j=2}^{d_c} \rho_j^{j-1}$, the optimized parameters $\{\lambda_i\}_{i=2}^{d_v}$ and the gap to the 2-user Gaussian MAC Shannon limit.

We can see that the LDPC codes optimized for the 2-user MAC channel are indeed very good and have decoding thresholds very close to the capacity. Our numerical results show that, the Gaussian a priori approximation is more attractive since the codes designed under this assumption have the smallest gap to Shannon limit.

An interesting result is that the codes obtained for $R = 0.3$ and $R = 0.6$ are worse that the ones obtained for $R = 0.5$. Our opinion is that it does not come from the same reason. For small rates ($R = 0.3$), the multiuser problem is *easy* to solve because the system load (sum-rate) is lower

	GA							
Rate	0.3		0.4		0.5		0.6	
$\rho(x)$	x^7		x^8		x^9		x^{10}	
$\lambda(x)$	λ_i	i	λ_i	i	λ_i	i	λ_i	i
	$2.749809e-01$	2	$2.786702e-01$	2	$3.170178e-01$	2	$4.393437e-01$	2
	$2.040936e-01$	3	$2.306721e-01$	3	$2.312804e-01$	3	$1.305465e-01$	3
	$5.708851e-03$	4	$5.059420e-02$	9	$4.241393e-02$	17	$2.508237e-02$	20
	$1.817382e-02$	5	$4.229097e-04$	10	$1.714436e-01$	18	$2.462773e-01$	21
	$1.891399e-02$	6	$1.608676e-01$	12	$2.378443e-01$	100	$1.587501e-01$	100
	$2.682255e-02$	7	$2.787730e-01$	100				
	$7.317063e-02$	8						
	$1.130643e-01$	13						
	$2.650713e-01$	100						
$\delta - S_l$	0.22		0.15		0.19		0.52	

Table 1: Optimized LDPC codes for the 2-user Gaussian channel obtained with the Gaussian Approximation of the state check node. The distance between the (E_b/N_0) threshold δ (evaluated with true DE) and the Shannon limit S_l is given in dBs.

than 1, but the approximations of DE become less and less accurate as the rate decreases. $R = 0.3$ gives worse codes than $R = 0.5$ because of the LDPC part of the multiuser graph. For larger rates ($R = 0.6$), the DE approximations are fairly accurate, but the multiuser problem we address is more difficult, as the system load is larger than 1 (equal to 1.2). $R = 0.6$ gives worse codes than $R = 0.5$ because of the multiuser part of the graph (state check node).

In order to verify the asymptotical results obtained with DE, we have made extensive simulations for a finite length equal to $N = 50000$. The codes have been build with an efficient parity check matrix construction. Since the progressive edge growth algorithm [6] tends to be inefficient at very large code lengths, we used the ACE algorithm proposed in [17] which helps to prevent the apparition of small cycles with degree two bitnodes. The ACE algorithm generally lowers greatly the error floor of very irregular LDPC codes (like the ones in tables 1 and 2).

Figure 3 shows the simulation results for three rates $R \in \{0.3, 0.5, 0.6\}$ and for the two different approximations of the state check node EXIT function presented in this paper: GA and EC. The curves are in accordance with the threshold computations, except the fact that codes optimized with the EC approximation tend to be better than the GA codes for the rate $R = 0.3$. We confirm also the behavior previously discussed in that the codes with $R = 0.5$ are closer to the Shannon

	EC							
Rate	0.3		0.4		0.5		0.6	
$\rho(x)$	x^7		x^8		x^9		x^{10}	
$\lambda(x)$	λ_i	i	λ_i	i	λ_i	i	λ_i	i
	$2.762791e-01$	2	$2.792405e-01$	2	$3.165084e-01$	2	$4.388191e-01$	2
	$2.321906e-01$	3	$2.456371e-01$	3	$2.339989e-01$	3	$1.303074e-01$	3
	$7.870900e-02$	9	$1.020663e-01$	13	$4.285469e-02$	18	$1.649224e-01$	20
	$1.077795e-01$	10	$8.130383e-02$	14	$1.713483e-01$	19	$1.093493e-01$	21
	$3.050418e-01$	100	$2.917522e-01$	100	$2.352897e-01$	100	$1.566018e-01$	100
$\delta - S_l$	0.38		0.26		0.21		0.59	

Table 2: Optimized LDPC codes for the 2-user Gaussian channel obtained with the Erasure Channel Approximation of the state check node. The distance between the (E_b/N_0) threshold δ (evaluated with true DE) and the Shannon limit S_l is given in dBs.

Limit than the codes with $R = 0.3$ and $R = 0.6$.

Conclusion

This paper has tackled the optimization of LDPC codes for the 2-user Gaussian MAC and has shown that it is possible to design good irregular LDPC codes with very simple techniques, the optimization problem being solved by linear programming. We have proposed 2 different analytical approximations of the state check node update, one based on a Gaussian approximation and one very simple based on an Erasure Channel approach. The codes obtained have decoding thresholds as close as $0.15dB$ away from the Shannon limit, and can be used as initial codes for more complex optimization techniques based on true Density Evolution. Future work will deal with the generalization of our approach to more than two users and/or users with different powers.

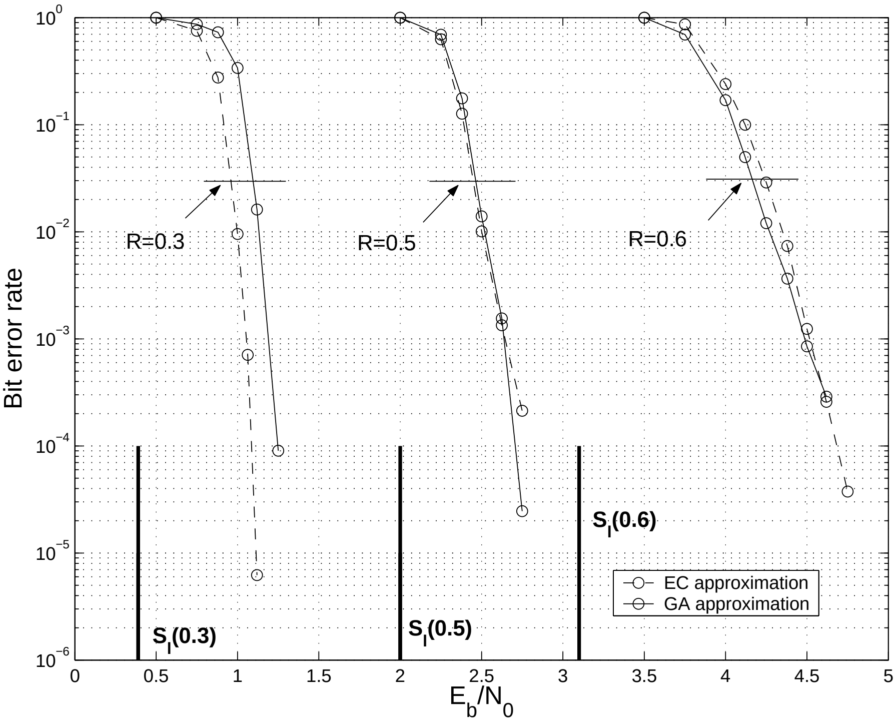

Figure 3: Simulation results for the optimized LDPC codes given in table 1 and 2. The codeword length is $N = 50000$. The maximum number of iterations is set to 200. For comparison, we have indicated the Shannon limit for the three considered rates.

APPENDIX

A Computation of functions $F_{+1,+1}$ and $F_{+1,-1}$

We proceed to compute the state check node update rule for the mean of the messages.

Let us first consider hypothesis $Z = [+1, +1]^T$. Under the Gaussian assumption, the conditional input distributions are:

$$\begin{cases} y|_{(+1,+1)} & \sim & \mathcal{N}(2, \sigma^2) \\ m_{vs}|_{(+1,+1)} & \sim & \mathcal{N}(\mu_{vs}, 2\mu_{vs}) \end{cases}$$

Therefore

$$\begin{cases} m_{vs} + \frac{2y-2}{\sigma^2}|_{(+1,+1)} & \sim & \mathcal{N}\left(\mu_{vs} + \frac{2}{\sigma^2}, 2\mu_{vs} + \frac{4}{\sigma^2}\right) \\ m_{vs} + \frac{2y+2}{\sigma^2}|_{(+1,+1)} & \sim & \mathcal{N}\left(\mu_{vs} + \frac{6}{\sigma^2}, 2\mu_{vs} + \frac{4}{\sigma^2}\right) \end{cases}$$

Since for a Gaussian random variable $x \sim \mathcal{N}(\mu + a, 2\mu + b)$, where a and b are real valued constants,

$$E\left[\log\left(1 + e^{\pm x}\right)\right] = \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log\left(1 + e^{\pm(\sqrt{4\mu+2b}z+\mu+a)}\right) dz$$

and by using (6), we get

$$E\left[m_{sv}|Z=[+1,+1]^T\right]=-\mu_{vs}+\frac{1}{\sqrt{\pi}}\int_{-\infty}^{+\infty}e^{-z^2}\log\left(\frac{1+e^{+2\sqrt{\mu_{vs}+\frac{2}{\sigma^2}}z+\mu_{vs}+\frac{2}{\sigma^2}}}{1+e^{-2\sqrt{\mu_{vs}+\frac{2}{\sigma^2}}z-\mu_{vs}-\frac{6}{\sigma^2}}}\right)dz=F_{+1,+1}\left(\mu_{vs},\sigma^2\right)$$

Similarly we get $F_{+1,-1}\left(\mu_{vs},\sigma^2\right)$.

B Proof of Proposition 1

To prove Proposition 1, we first need to show the following lemmas.

Lemma 1. Consider a state check node. Let us assume a symmetric input message and a symmetric channel observation. The output message is symmetric.

Proof of Lemma 1. We consider a state check node that verifies the symmetry condition (see Definition 1). Without loss of generality we can assume k to be the output user and j the input user.

Let y (z resp.) denote the observation vector when the codewords $x^{[k]}, x^{[j]}$ ($-x^{[k]}, -x^{[j]}$ resp.) are sent. Now note that a symmetric-output 2-user MAC can be modeled as (see Lemma 1 in [11]))

$$y=-z$$

since $p(y_t|x_t^{[k]}, x_t^{[j]}) = p(-y_t|-x_t^{[k]}, -x_t^{[j]})$ and since we are interested in the performance of the BP algorithm i.e. the densities of the messages.

Similarly we denote $m_t^{[j]}, m_t^{[k]}$ ($r_t^{[j]}, r_t^{[k]}$ resp.) the input and output messages of the state check node at position t when the codewords $x^{[k]}, x^{[j]}$ ($-x^{[k]}, -x^{[j]}$ resp.) are sent.

Let us assume a symmetric input message i.e. $p(m_t^{[j]}|x_t^{[j]}) = p(-m_t^{[j]}|-x_t^{[j]})$. Here again we can model this input message as

$$m_t^{[j]}(y) = -r_t^{[j]}.$$

The state check node update rule is denoted $\Psi_S(y_t, m_t^{[j]})$.

The output message verifies

$$\begin{aligned}
m_t^{[k]} &= \Psi_S(y_t, m_t^{[j]}) \\
&= \Psi_S(-z_t, -r_t^{[j]}) \\
&= -\Psi_S(z_t, r_t^{[j]}) \\
&= -r_t^{[k]}(z)
\end{aligned}$$

where the second equation is due to the symmetry conditions of the channel and the input message and the third equation follows from the symmetry condition of the state check node map.

This can be rewritten as

$$p(m_t^{[k]}|x_t^{[k]}, x_t^{[j]}) = p(-m_t^{[k]}| - x_t^{[k]}, -x_t^{[j]}) \tag{17}$$

and therefore

$$p(m_t^{[k]}|x_t^{[k]}) = p(-m_t^{[k]}| - x_t^{[j]}) \tag{18}$$

by marginalizing the probability with respect to $x_t^{[j]}$ and by using (17).

Equation (18) implies that with symmetric observation and symmetric input message, the message at the state check node output is also symmetric. The symmetry is conserved through the state check node which completes the proof of Lemma 1. □

Lemma 2. Consider a state check node. Let us assume a symmetric channel observation. At any iteration, the input and output messages of the state check node are symmetrics.

Proof of Lemma 2. Lemma 1 shows that the state check node conserves the symmetry condition, Lemma 1 in [11] shows the conservation of the symmetry condition of the messages through the variable and check node. At initialization, the channel observation is symmetric therefore a proof by induction shows the conservation of the symmetry property at any iteration with a BP decoder. □

Proof of Proposition 1. A consequence of Lemma 1, is that the number of cases that need to be considered to determine the entire average behavior of the state check node can be divided by a factor 2. We can assume that the all-one sequence is sent for the output user. However all the sequences of the input user need to be considered and therefore on the average we can assume an input sequence with half symbols fixed at '1' and half symbols at '-1'. □

C Proof of Proposition 2

Lemma 3. Under the parallel scheduling assumption described in section 1 and by using hypothesis $\mathcal{H}_0$ (see section 3), the entire behavior of the BP decoder can be predicted with one decoding iteration (i.e. half of a round).

Proof of Lemma 3. Under the parallel scheduling assumption described in section 1, two decoding iterations (one for each user) are completed simultaneously. Hence by using hypothesis $\mathcal{H}_0$ (same code family for both users), the two decoding iterations are equivalent in the sense that they provide messages with the same distribution. This can be easily shown by induction. It follows that a whole round is entirely determined by only one decoding iteration (i.e. half of a round). □

Therefore in the following we omit the user index.

Proof of Proposition 2. We now proceed to compute the evolution of the mutual information through all nodes of the graph. By assuming that the distributions at any iteration are Gaussian, we obtain similarly to method 1 in [14] the mutual information evolutions as

$$\left\{\begin{array}{lcl} x_{vc}^{(l)} & = & \sum_{i=2}^{d_v} \lambda_i J\Big(J^{-1}\big(x_{sv}^{(l-1)}\big) + (i-1)J^{-1}\big(x_{cv}^{(l-1)}\big)\Big) \\ x_{cv}^{(l)} & = & 1 - \sum_{j=2}^{d_c} \rho_j J\Big((j-1)J^{-1}\big(1 - x_{vc}^{(l-1)}\big)\Big) \\ x_{vs}^{(l)} & = & \sum_{i=2}^{d_v} \tilde{\lambda}_i J\Big(iJ^{-1}\big(x_{cv}^{(l)}\big)\Big) \\ x_{sv}^{(l)} & = & f\big(x_{vs}^{(l)}, \sigma^2\big) \end{array}\right. \tag{19}$$

where $\tilde{\lambda}_i$ denotes the fraction of variable nodes of degree i ($\tilde{\lambda}_i = (\lambda_i/i)/(\sum_j \lambda_j/j)$) and where

$$f(x_{sv}, \sigma^2) = \frac{1}{2} x_{sv}|_{(+1,+1)} + \frac{1}{2} x_{sv}|_{(+1,-1)}$$

with x_{sv} defined either in (9) or (13), depending on the approach used.

First notice that this system is not linear in the parameters $\{\lambda_i\}$. But by using hypothesis $\mathcal{H}_1$, the input message m_{sv} of a variable node of degree i results from a variable node with the same degree. It follows that the third equation in (19) reduces to

$$x_{vs}^{(l)} = J\Big(iJ^{-1}\big(x_{cv}^{(l)}\big)\Big)$$

Finally the global recursion in the form (14-15) is obtained by combining all four equations and the global recursion is linear in the paremeters $\{\lambda_i\}$. □

D Proof of Proposition 3

Similarly to the definition of the message (see Section 1) and of the mutual information (see Section 2), we will denote $P_{ab}^{(l)}$ the distribution of the messages from node a to node b in iteration l, where (a, b) can either be v for variable node, c for check node or s for state check node.

We follow in the footsteps of [12] and analyze the local stability of the zero error rate fixed point by using a small perturbation approach. Let us denote Δ_0 the dirac at 0 i.e. the distribution with 0.5-BER and $\Delta_{+\infty}$ the distribution with zero-BER when the symbol '+1' is sent.

From Lemma 3 (see Appendix C) we know that only half of a complete round needs to be performed in order to get the entire behavior of the BP decoder. All distributions of the DE are conditional densities of the messages given that the symbol sent is +1. From the symmetry property of the variable and check nodes, the transformation of the distributions can be performed under the assumption that the all-one sequence is sent. However for the state check node, different cases will be considered as detailed below.

We consider the DE recursion with state variable of the dynamical system P_{vc}. In order to study the local stability of the fixed-point Δ_∞, we initialize the DE recursion at the point:

$$P_{vc}^{(0)} = (1-2\epsilon)\Delta_\infty + 2\epsilon\Delta_0$$

for some small $\epsilon > 0$, and we apply one iteration of the DE recursion. Following [12] (and also in [14]), the distribution $P_{cv}^{(0)}$ can be computed which leads to $P_{vs}^{(0)}$ as

$$P_{vs}^{(0)} = \Delta_\infty + O(\epsilon^2). \tag{20}$$

For the sake of brevity, we omit the now well known step-by-step derivation and focus on the transformation at the state check node. Note that (20) holds with and without the hypothesis $\mathcal{H}_1$ (without interleaver) since it follows from the fact that a i-fold convolution of the distribution $P_{cv}^{(0)}$ is performed with $i \geq 2$ in both cases.

From the symmetry property (Proposition 1) of the state check node, the entire behavior at a state check node can be predicted under the two hypotheses called $(+1,+1)$ and $(+1,-1)$ i.e. when the output symbol is $+1$ and when the input symbol is either $+1$ or -1 with probability 1/2 each. In the following, we seek for the output distribution $P_{sv}^{(0)}$, for a given input distribution $P_{vs}^{(0)}$ (conditional distribution given that the input symbol is $+1$) and a given channel distribution.

Hypothesis $(+1,+1)$ w.p. $\frac{1}{2}$. From (20) and (5) we get

$$\begin{cases} m_{vs}^{(0)} & \sim & P_{vs}^{(0)} = \Delta_\infty + O(\epsilon^2) \\ y & \sim & \mathcal{N}(2,\sigma^2) \end{cases}$$

Hence, by applying (4) we have

$$m_{sv}^{(0)} = \frac{2+2y}{\sigma^2} \sim \mathcal{N}\left(\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\right) \tag{21}$$

Hypothesis $(+1,-1)$ w.p. $\frac{1}{2}$. From (20) and from the symmetry property of the input message at the state check node, we have

$$m_{vs}^{(0)} \sim P_{vs}^{(0)}(-z) = \Delta_{-\infty} + O(\epsilon^2)$$

and from (5) we get

$$\begin{cases} m_{vs}^{(0)} & \sim & \Delta_{-\infty} + O(\epsilon^2) \\ y & \sim & \mathcal{N}(0,\sigma^2) \end{cases}$$

Hence, by applying (4) we have

$$m_{sv}^{(0)} = \frac{-2y-2}{\sigma^2} \sim \mathcal{N}\left(\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\right) \tag{22}$$

Combining (21) and (22), we obtain

$$P_{sv}^{(0)} = \mathcal{N}\left(\frac{2}{\sigma^2}, \frac{4}{\sigma^2}\right).$$

It follows that at convergence, the channel seen by one user is $P_{sv}^{(0)}$ which is exactly the LLR distribution of a BIAWGNC with noise variance σ^2. It follows that at convergence the DE recursion is equivalent to the single user case and the stability condition is therefore [12]

$$\lambda_2 < \exp\left(\frac{1}{2\sigma^2}\right) / \sum_{j=2}^{d_c}(j-1)\rho_j$$

□

References

[1] R. Ahlswede, “Multi-way communication chanels, *Proc. Int. Symp. Information Theory, Aremenian S.S.R. Prague*, pp. 23-52, 1971.

[2] A. Amraoui, S. Dusad, R. Urbanke, “Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative decoding”, in *(ISIT 2002)*, July 2002, p. 334.

[3] A. Bennatan and D. Burshtein, “On the application of LDPC codes to Arbitrary Discrete Memoryless Channels,” *IEEE Trans. on Info. Theory*, vol. 50, pp. 417-438, March 2004.

[4] S.-Y. Chung, T. J. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian Approximation,” *IEEE Trans. on Inf. Th*, vol. 47, no. 2, February 2001, pp. 657-670.

[5] A. DeBaynast, D. Declercq, “Gallager Codes for Multiple Access”, in *(ISIT 2002)*, July 2002.

[6] X. Hu, E. Eleftheriou and D.M. Arnold, ”Progressive Edge Growth Tanner Graphs”, *IEEE Global Telecommun. Conf.*, pp 995-1001, November 2001, San Antonio (USA).

[7] F.R. Kschischang, B.J. Frey, H.A. Loeliger, “Factor graphs and the sum-product algorithm”, *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 498-519, Feb. 2001.

[8] H. Liao, *Multiple Access channels*, PhD thesis, University of Hawai, 1972.

[9] R. Palanki, A. Khandekar, and R. McEliece, “Graph-based codes for synchronous Multiple Access Channels,” Proceedings of *the 39th Annual Allerton Conference on Communication, Control, and Computing*, Oct 2001.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA, 1988.

[11] T.J. Richardson, R.L. Urbanke, “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 599-618, Feb. 2001.

[12] T.J. Richardson, M.A. Shokrollahi, R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 619-637, Feb. 2001.

[13] B. Rimoldi and R. Urbanke, “A rate splitting approach to the Gaussian multiple-access channel,” *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 364–375, March 1996.

[14] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, “Design methods for irregular repeat accumulate codes,” *IEEE Trans. on Inform. Theory*, vol. 50, no. 8, pp. 1711–1727, Aug. 2004.

[15] R. M. Tanner, “A recursive approach to low complexity codes, IEEE Trans. Inform. Theory, vol. IT-27, pp. 533547, Sept. 1981.

[16] S. ten Brink, “Designing iterative decoding schemes with the extrinsic information transfer chart,” *AEÜ Int. J. Electronic. Commun.*, vol. 54, no. 6, pp. 389-398, Dec. 2000.

[17] T. Tian, C. Jones, J.D. Villasenor and R.D. Wesel, ”Construction of Irregular LDPC Codes with low Error Floors”, *IEEE International Conference on Commun.*, pp. 3125-3129, May 2003.

[18] C.C. Wang, S.R. Kulkarni and H.V. Poor, “Density Evolution for Asymmetric Memoryless Channels,” to appear in *IEEE Trans. on Info. Theory*, 2006.