Performance Analysis of Structured LDPC over the Polya-Urn Channel with Finite Memory

Vijay Nagarajan and Olgica Milenkovic
Electrical and Computer Engineering Department, University of Colorado
Boulder, Colorado 80309

E-mail: {nagarajn, milenkov}@colorado.edu

***Abstract*-- In this paper we investigate the performance of structured LDPC codes over a simple channel with memory. The channel under consideration is based on Polya's urn model [1] and can be used as a representative of several classes of practical channels with burst errors. Random codes have large storage requirements, and this makes the complete error-control scheme more difficult to implement. We will compute the bit error rate (BER) curves for iterative decoding with channel estimation, for both random codes and codes based on parity-check matrices of the form of block-circulants with permutation blocks and show that the latter codes have performance of the quality of random codes, with significantly smaller storage complexity. Furthermore, following the approach in [4], we will determine the complete convergence region within the parameter space of the channel and develop a new technique for generating side-information which also allows for complete characterization of the convergence region of the Gilbert-Elliott channel analyzed in [4].**

1. Introduction

Low-Density Parity-Check (LDPC) codes introduced by Gallager [5] are known to have excellent error-correcting characteristics under iterative decoding [6,7]. The performance of these codes over memoryless channels has been studied extensively, leading to the design of capacity approaching codes using tools such as density evolution [12,13] or EXIT chart analysis [2]. Many practical and ubiquitous channels, such as wireless channels, have memory. The problem of code design/analysis for channels with memory has recently become the focus of many research papers. Several algorithms for joint channel state estimation and iterative decoding have been proposed in [4,15]. In [4], a comprehensive analysis of LDPC codes for application over the Gilbert-Elliott (henceforth, GE) channel was conducted. The convergence region was characterized in a three-dimensional space of parameters, and the complete density evolution algorithm was outlined.

The GE channel is one of the simplest models used for channels with memory, and it is specified by four parameters. On the other hand, the finite memory version of the communication channel described in terms of the Polya's urn model for contagion [1] (henceforth, the PC channel) depends only on three parameters and can be used to represent a variety of channels. For the special case of memory of length one, the model reduces to a simple Markov chain with two states. For reasons of clarity, we will restrict our attention to this case only, although all results presented in the next several sections can be extended to apply to the more general case. In the paper, we will first describe the message-passing algorithm and density evolution for the PC channel. Based on these results, we will evaluate the performance of random codes and a special class of structured LDPC codes over this channel, for the case of iterative decoding with channel state estimation. Channel-state estimation is performed by using the standard BCJR algorithm [3]. Finally, we will present a complete characterization of the zero-error-probability (convergence) region in the underlying parameter space, based on a new side-information technique referred to as state XOR-ing. State XOR-ing can also be used to complete the characterization of the convergence region for the GE channel, described in [4].

The outline of the paper is as follows: Section 2 gives a brief description of the PC model. Section 3 contains a short overview of the construction techniques for LDPC codes under consideration. The iterative channel estimation/decoding algorithm is described in Section 4. Section 5 presents a complete characterization of the convergence region of the PC channel. Section 6 reports the simulation results, while Section 7 highlights the issues in modeling bursty channels as urn representations and a method to solve the same. The concluding remarks are presented in Section 8.

2. Finite Memory Polya-Contagion Channel Model

The PC channel is a binary, additive channel with memory for which the occurrence of an error increases the probability of future errors [1]. The output sample at time i, Y_i, is the modulo-two sum of the input X_i and the noise sequence Z_i , where $\{X_i\}$ and $\{Z_i\}$ are independent sequences. The noise process $\{Z^{(i)}\}_{i=1}^{\infty}$ is formed based on the following scheme: an urn initially contains T balls, where R balls are red and S are black. At the j^{th} draw, j=1,2,… a ball is randomly selected from the urn and replaced with 1+Δ balls of the same color. After an initial set of M draws, Δ balls of the color picked at time j are retrieved from the urn. This process is applied to each subsequent drawing. The noise samples $\{Z^{(i)}\}_{i=1}^{\infty}$ are the outcomes of consecutive draws from the urn, and Z_i is set to one if the i^{th} ball was red and to zero if the i^{th} ball was black. The finite-memory PC channel described above is a stationary, ergodic channel with memory M, whose capacity increases with M. Only three parameters are needed to describe the channel, namely (σ, δ, M), where $\rho=R/T < ½$, $\sigma = 1- \rho = S/T$ and $\delta = \Delta/T$.

The PC channel represents an interesting alternative to many other classes of channels with memory, and it can be used as a simplified approximation for a wireless channel. The small dimension of the parameter space of the PC channel makes detailed analysis of density evolution and the description of the convergence region particularly simple. In what follows, we will restrict our interest to the simplest case M=1. As already pointed

out, the analytic methods described in subsequent sections can be extended and generalized for arbitrary values of *M* but is omitted due to space limitations.

Let $a = (\sigma + \delta)/(1+ \delta)$ and $b = (1- \sigma + \delta)/(1+ \delta)$, and define a two-state Markov chain with states S_0 and S_1 and transition probabilities 1-*a* and 1-*b*, respectively. State S_0 corresponds to a noise sample of value zero, and state S_1 corresponds to a noise sample of value one. The channel transition matrix is given by:

$$\Pi = \begin{bmatrix} a & 1-a \\ 1-b & b \end{bmatrix} \quad (1)$$

Note that this channel model differs from the GE channel analyzed in [4], since the latter covers only the case when the inversion probability *p* is such that $0<p<1$.

3. Design of Structured LDPC Codes

The high hardware complexity of error-control schemes exploiting LDPC codes is a direct consequence of the fact that for the prevalently used random codes a large amount of information is necessary to specify positions of the nonzero elements in the parity-check matrix. Structured LDPC codes, a concept opposed to the prevalent practice of using random code constructions, were first investigated by Kou, Lin and Fossorier [8] and Rosenthal and Vontobel [16]. The parity-check matrix of these codes is completely determined by a small set of parameters resulting in very low complexity implementations.

For the purpose of performance analysis of structured codes over channels with memory, we will use a coding schemes developed by the authors and described in a companion paper [10]. For this construction, *H* is defined as a $(m\times N)\times(s\times N)$ block-circulant matrix with permutation blocks, of the form

$$H = \begin{bmatrix} P^{i_1} & P^{i_2} & P^{i_3} & \dots & P^{i_s} \\ P^{i_s} & P^{i_1} & P^{i_2} & \dots & P^{i_{s-1}} \\ \dots & \dots & \dots & \dots & \dots \\ P^{i_{s-m+2}} & P^{i_{s-m+3}} & P^{i_{s-m+4}} & \dots & P^{i_{s-m+1}} \end{bmatrix} \quad (2)$$

where $i_1,\dots,i_s$ are non-negative integers and

$$P = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (3)$$

is a $N\times N$ standard permutation matrix. It was shown in [14] that if the integers $i_1,\dots,i_s$ form a so called *cycle-invariant difference set* (CIDS), then the structured LDPC code specified by Equation (2) has girth at least six and minimum distance at least six. Simulation results show that the codes based CIDS have excellent performance over the Additive White Gaussian Noise (AWGN) channel and under iterative decoding, even when the number of iterations is as small as 15. The hardware requirements for this class of structured LDPC codes are reduced to only storing the integer exponents of the permutation matrices (for more details regarding code construction, the reader is referred to [10,14]).

Example 1: Let *S* be a set of integers and let *M* be an integer greater than one. *S* is called a difference set modulo *M*, if the pair-wise differences of elements from *S* modulo *M* generate every non-zero integer less than *M* not more than once. For example, for *S*={1,2,4} and *M*=7, one has:

1-2=-1 mod 7=6, 1-4=-3 mod 7=4, 2-1=1 mod 7, 2-4=-2=5 mod 7, 4-1=3 mod 7, and 4-2=2 mod 7

Note that in the above example every non-zero element appears *exactly* once (i.e. it is a complete difference set). A CIDS is a difference set that has the additional property that the differences of the elements of the set and a certain number of its cyclic shifts, modulo *m*, are also difference sets. For example *{4,1,2}-{1,2,4}={3,6,5}*, and *{3,6,5}* is also a difference set.

We will next describe a construction of CIDS based on the theory of finite fields.

Theorem 1: Let ω be a primitive element of the finite field $GF(q^4)$. Define a set *S* of integers by

$$S = \{a : 0 \le a < q^4 - 1, \omega^a + \omega \in GF(q)\} \quad (4)$$

Then the set *S* forms a *q*-fold CIDS mod $q^4 - 1$.

Sketch of Proof: The proof is a modification/generalization of the construction for difference sets due to Bose.

Take an arbitrary ordering of the set *S* and define $c_{a_i} = \omega^{a_i} + \omega$. Consider the following two polynomials:

$$\begin{aligned} p_1(x) &= (x-c_i)(x-c_j)-(x-c_l)(x-c_f), \\ p_2(x) &= (x-c_{i-t})(x-c_{f-t})(x-c_l)(x-c_j)-(x-c_{l-t})(x-c_{j-t})(x-c_i)(x-c_f) \end{aligned} \quad (5)$$

where $1 \le t \le q-1$ and the subscripts in $p_2(x)$ are taken modulo |*S*|=*q*.

Assume there exist indices *i,j,l,f* such that $i \ne l \wedge j \ne f, i \ne j \wedge l \ne f$ and $a_i - a_l = a_j - a_f \mod (q^4 - 1)$. The polynomial $p_1(x)$ has coefficients in *GF*(*q*), degree at most one and ω as its root. Since ω is a primitive element of $GF(q^4)$, no non-zero polynomial over *GF*(*q*) of degree smaller than four can have ω as a root. Hence, it follows that $p_1(x) \equiv 0$. Therefore, $\{c_i, c_j\} = \{c_l, c_f\}$ which implies $\{a_i, a_j\} = \{a_l, a_f\}$ and consequently {*i,j*}={*l,f*}. This contradicts the starting assumption. Hence, S is a difference set modulo $q^4 - 1$.

Assume next that that the integers *i,j,l,f* are as described before and that

$$(a_{i-t} - a_i) - (a_{j-t} - a_j) = (a_{l-t} - a_l) - (a_{f-t} - a_f) \mod (q^4 - 1), \quad (6)$$

where the subscripts are taken modulo |*S*|=*q*. If $c_{a_i} = \omega^{a_i} + \omega$, then the polynomial $p_2(x)$ has coefficients in *GF*(*q*), degree at most three and ω as one of its root. Since ω is a primitive element of $GF(q^4)$ it follows that $p_2(x) \equiv 0$. Hence,

$$\{c_{i-t}, c_{f-t}, c_l, c_j\} = \{c_{l-t}, c_{j-t}, c_i, c_f\}$$

and consequently

$$\{a_{i-t}, a_{f-t}, a_l, a_j\} = \{a_{l-t}, a_{j-t}, a_i, a_f\},$$

or equivalently

$$\{i-t, f-t, l, j\} = \{l-t, j-t, i, f\}.$$

One can distinguish several possible cases for which the two sets above are equivalent, each contradicting the starting assumptions. Therefore, *S* is a *q*-fold cycle invariant difference set. ■

Theorem 2: Let i_l, $1 \le l \le s$, be elements of a (m+1)-fold CIDS constructed according to Theorem 1. Then the LDPC code specified by H of the form given by Equation (5) has girth at least ten.

Proof: It is straightforward to see that there cannot be any cycles of length four in any code constructed using cycle-invariant difference sets. In order to have a cycle of length six in the code, there must exist six elements of the CIDS, say $\{i_1, i_2, ..., i_6\}$ such that without loss of generality

$$i_1 - i_2 = (i_3 - i_4) + (i_5 - i_6) \mod q^4 - 1.$$

This is equivalent to the requirement that

$$\omega^{i_1 - i_2} = \omega^{i_3 - i_4} \cdot \omega^{i_5 - i_6}, \text{ or } \omega^{i_1 + i_4 + i_6} = \omega^{i_2 + i_3 + i_5}.$$

Based on Theorem 2, this implies the existence of six different integers $b_1, b_2, ..., b_6 \in GF(q)$, such that the previous equation can be written as

$$(b_1 - \omega) \cdot (b_4 - \omega) \cdot (b_6 - \omega) = (b_2 - \omega) \cdot (b_3 - \omega) \cdot (b_5 - \omega).$$

The previous equation implies that ω is a root of an equation of degree at most two with coefficients over $GF(q)$, contradicting the starting assumption that ω is a primitive element of the field $GF(q^4)$.

One can show in a similar manner that there are no cycles of length eight in the codes, since in this case one would have to have eight different elements of the cycle invariant set such that the sum of the differences of two pairs is equal to the sum of the differences of the other two pairs. This would result in an equation of degree at most three of which ω is a root. This, again, would contradict the assumption that ω is a primitive element of the field $GF(q^4)$ ■

Example 2: ***a)*** If the construction in Theorem 1 is performed over GF(q^2), then the resulting set is in most cases is a CIDS as well. One such example is the following set:
S= { 2, 17 ,27, 39, 47, 58, 79, 85, 102, 136, 145, 149, 150, 152, 178, 231}, resulting in a code of length 4608. The dimension of the code is determined by the choice of the parameter *m* described above.
b) Another example is the following CIDS set obtained from Theorem 1 with *q*=7 is *S= {431, 561, 1201, 1312, 1406, 1579, 1883},* resulting in a code of length 16800.

4. Analysis of PC-LDPC Density Evolution and Joint Decoding /Channel Estimation

In this section, we will describe the message passing algorithm and density evolution for the PC-LDPC decoder/estimator. In order to perform joint decoding and channel estimation, it is necessary to use the extrinsic information from the decoder in order to improve the quality of the estimate of the channel state and vise-versa. The PC-LDPC decoder/estimator is shown in Figure 1.

From Figure 2, one can observe that the extrinsic information from the decoder, χ is a log-likelihood ratio (LLR) and that

$$P\{X_i = Y_i \mid \chi\} = q(\chi) = \frac{1}{2}\left(1 + (-1)^{Y_i} \tanh\left(\frac{\chi}{2}\right)\right). \quad (7)$$

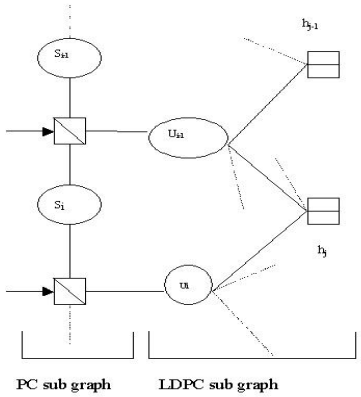

Figure 1: PC-LDPC Decoder

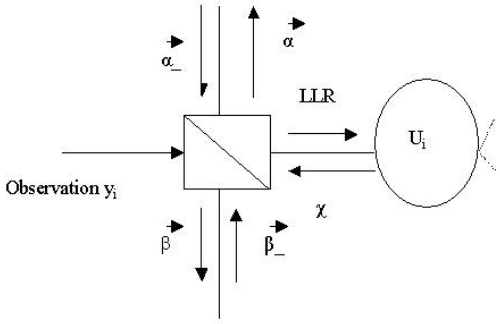

Figure 2: BCJR algorithm for the channel state estimation incorporating the extrinsic information from the decoder

Then, using the BCJR algorithm[3], the forward message vector $\vec{\alpha} = (\alpha(1), \alpha(2))$ can be computed recursively as:

$$\alpha(1) = P\{S(i) = S_0; Y_1^i; \chi\} = \alpha^-(1) a q(\chi) + \alpha^-(2)(1-b) q(\chi) \quad (8)$$

$$\alpha(2) = P\{S(i) = S_1; Y_1^i; \chi\} = \alpha^-(1)(1-a)(1-q(\chi)) + \alpha^-(2) b (1-q(\chi))$$

where $S(i)$ is the state at time i, Y_1^i denotes the observed sequence $\{Y_1, Y_2 \ldots, Y_i\}$, while $\vec{\alpha^-} = (\alpha^-(1), \alpha^-(2))$ represents the message received from state $S(i\text{-}1)$.

Similarly, the backward message vector $\vec{\beta} = (\beta(1), \beta(2))$ is given by:

$$\beta(1) = P\{Y_i^n \mid S(i) = S_0, \chi\} = \beta^-(1) a q(\chi) + \beta^-(2)(1-a)(1-q(\chi)) \quad (9)$$

$$\beta(2) = P\{Y_i^n \mid S(i) = S_1, \chi\} = \beta^-(1)(1-b) q(\chi) + \beta^-(2) b (1-q(\chi))$$

In Equation (9), $\vec{\beta^-} = (\beta^-(1), \beta^-(2))$ represents the message received from state $i+1$, and n represents the total number of symbol variable nodes. The channel information passed to the decoder can be derived as:

$$LLR = (-1)^{Y_i} \log\left(\frac{\beta(1)\left[\alpha^-(1) a + \alpha^-(2)(1-b)\right]}{\beta(2)\left[\alpha^-(1)(1-a) + \alpha^-(2) b\right]}\right) \quad (10)$$

To derive the density of the channel state variables, one can adapt the procedure in [4]. Due to space limitations, this procedure will not be described in more detail.

5. Characterization of the Convergence Region of the PC Channel

In this section, we will characterize the convergence region for iterative decoding within the parameter space of the PC channel. Since the concept of physically degraded channels can be applied only to the case of memoryless channels, the authors of [4] derived a set of new methods to analyze convergence regions for the GE channel. We will use two of their techniques, the so-called Side-Information Lemma and the Segmentation argument to derive partial characterization of the convergence region for the PC channel. We will then describe a new technique, referred to as state XOR-ing, which allows for complete characterization of the convergence region. Furthermore, this technique can be applied to the GE channel to provide for the missing specification of the convergence region in the fourth dimension [4].

In this section, we will follow the same notations as used in [4]. For the sake of completeness, we will briefly repeat the most important definitions: P_e will represent the probability of error and $P_e[l]$ represent the aposteriori probability of symbol error after the l^{th} iteration of PC-LDPC decoding. The error at a specific point in the parameter space is denoted as $P_e[l](a,b)$. The expression $lim_{l\to\infty} P_e[l]=0$ will be described by $P_e[l]\to 0$ and similarly, $P_e[l] \not\to 0$ will be used to state that $lim_{l\to\infty} P_e[l]>0$.

5.1 Segmentation for the PC channel

The first characterization of the parameter space comes from the segmentation technique introduced and developed in [4] for the GE channel. Consider a set $S=\{s^{(i)}\}_{i=1}^{\infty}$ of state sequences, where each sequence $s^{(i)}$ is generated independently by a PC state process with identical state transition probability parameters (a,b). From S, we construct a new state sequence s^* as follows. Let $\{d^{(i)}\}_{i=1}^{\infty}$ be a sequence of increasing integers such that $d_1 = 1$,and for all i, $d_{i+1} = d_i + u_i$, where u_i is {0,1}-Bernoulli random variable with $\Pr\{u_i = 1\} = p$. Given $\{d^{(i)}\}_{i=1}^{\infty}$, the elements of the sequence s^* are defined as $s_i^* = s_i^{(di)}$.

Lemma 1: *If s^* is defined as above, then the channel arising from s^* is a PC channel with parameters*

$a^*=1-(1-p)(1-a)-p(1-a)/(2-a-b)$,
$b^*=1-(1-p)(1-b)-p(1-b)/(2-a-b)$.

The lemma is a simple consequence of the segmentation technique and the fact that the stationary probabilities of state S_0 and S_1 can be easily found from the transition matrix Π as:

$$\pi_0 = \frac{1-b}{2-a-b}\ ,\quad \pi_1 = \frac{1-a}{2-a-b} \tag{11}$$

The complete proof is omitted.

Solving for p in Lemma 1, it can easily be shown that all channels obtained from a PC by segmentation can be represented by points on the line through (a^*, b^*) and (1,1).

Theorem 3: *Let (a^*,b^*) represent a valid point in the PC parameter space with $P_e[l]\to 0$. Then all points that lie on the line passing through (a^*,b^*) and (1,1) with $a>a^*$ and $b>b^*$ have $P_e[l]\to 0$.*

Sketch of Proof: The proof follows from the observation that a line through (1-a^*,1-b^*) and (0,0) in the (1-a,1-b) plane corresponds to a line through (a,b) and (1,1) in the (a^*,b^*) plane. ■

Corollory 1*:* Let (a^*,b^*) represent a valid point in the PC parameter space with $P_e[l] \not\to 0$. Then all points that lie on the line passing through (a^*, b^*) and (1,1) with $a>a^*$ and $b>b^*$ have $P_e[l] \not\to 0$. Thus, any point of convergence (a^*,b^*) induces a line of convergence in the (a,b) plane.

5.2 State XORing

In this subsection, we introduce a new technique, termed state XORing, in order to further characterize the convergence region of the PC space. For the case of the PC channel, this technique is equivalent to concatenation with a Binary Symmetric Channel (BSC). It will be shown in Appendix 1 that this technique can be applied to the GE channel to complete the characterization of its convergence region.

Consider a BSC channel with parameter p. This channel has an equivalent characterization as an i.i.d source which generates Bernoulli samples $U_i \in\{0,1\}$with $\Pr(U_i =1) = p$, and where $U_i=1$ represents message inversion and $U_i=0$ represents no inversion. We will show next that the concatenation of a PC channel with a BSC channel with parameter p results in another PC channel. In other words, XORing or modulo 2 addition of the state sequence with an i.i.d sequence of 0's and 1's with parameter p results in another PC channel.

Let S represent the state sequence of the original PC channel. Denote the new noise sequence constructed from state XOR-ing by $W_i =(S(i) + U_i) \bmod 2$.

By using the stationary probabilities of Equation (9), it is straightforward, but tedious to show that the parameters (a^*, b^*) of the new channel are of the form:

$$a^* = \frac{(1-b)(1-p)[a(1-p)+(1-a)p]+(1-a)p[(1-b)(1-p)+bp]}{(1-b)(1-p)+(1-a)p} \tag{12}$$

and

$$b^* = \frac{(1-b)p[ap+(1-a)(1-p)]+(1-a)(1-p)[(1-b)p+b(1-p)]}{(1-b)p+(1-a)(1-p)} \tag{13}$$

Solving for (a,b) gives:

$$a = \frac{(a^*-b^*-2)p^2+(2-2b^*)p+(a^*b^*-a^*)}{(a^*-b^*-2)2p^2+(4-b^*-3a^*)p+(b^*-1)}$$
$$b = \frac{(a^*-b^*-2)p^2+(2-2a^*)p+(a^*b^*-b^*)}{(a^*-b^*-2)2p^2+(4-a^*-3b^*)p+(b^*-1)} \tag{14}$$

Observe that unlike all the equations derived for the GE channel [4], the equations for a and b are not linear in a^* and b^*.

Clearly, side information in the form of the BSC noise sequence gives full information about the original channel and helps restore it. Hence, the above non-linear equations can be

used in conjunction with the Side Information Lemma [4] to prove the following theorem.

***Theorem 4* (State XORing or BSC concatenation)** *Let (a^*,b^*) be a valid point in the PC parameter space for which $P_e[l] \rightarrow 0$. Then all points (a,b) that satisfy Equation (14) for some p such that $0 \le p \le 1$ and*

- *$(a+b)<1$ & $b>a$ or*
- *$(a+b)>1$ & $b<a$*

have the property $P_e[l] \rightarrow 0$.

Sketch of Proof: For the PC channel, one has $\delta>0$ and a,b both depend on δ. Imposing this inequality within Equation (14), one obtains the two described conditions. Thus, if one can generate a channel with parameters (a,b) from a channel with parameters (a^*,b^*) for some BSC channel with transition probability p, then Equation (14) has to hold and vise-versa. Again, using the Side information lemma [4] one can show that if (a^*, b^*) is a point for which $P_e[l] \rightarrow 0$, then all points generated by Equation (14) and satisfying the stated conditions will also have the property of convergence to zero-error probability. ■

Conversely, if $P_e[l] \not\rightarrow 0$ for some (a^*,b^*), then any point (a,b) formed by concatenation of a PC channel (a^*, b^*) with a BSC cannot have a better $P_e[l]$. This establishes the following corollary.

Corollary 2: Let (a^*,b^*) be a valid point in the PC parameter space for which $P_e[l] \not\rightarrow 0$. Then all points (a,b) that satisfy Equation (12) for some p such that either

i) $(a+b)<1$ & $b>a$ or

ii) $(a+b)>1$ & $b<a$

is satisfied, have the property $P_e[l] \not\rightarrow 0$

Thus, state XOR-ing provides curves of convergence for every pair (a^*, b^*). This allows us to completely characterize the convergence region in the parameter space of the PC channel.

Consider a convergent point (a^*,b^*). This point induces a curve of convergent points based on the State XORing theorem. But for each such point on the curve, we will be able to draw a line of convergent points to (1,1) based on the Segmentation theorem. Note that due to the non-linearity of the equations used in state XORing, we will not get collinear one-dimensional regions based on the two theorems. Analogous to this region of convergence, using the two corollaries, we can induce a region of non-convergence for every non-convergent point in the *(a,b)* space. This two dimensional region is shown in Figure 3.

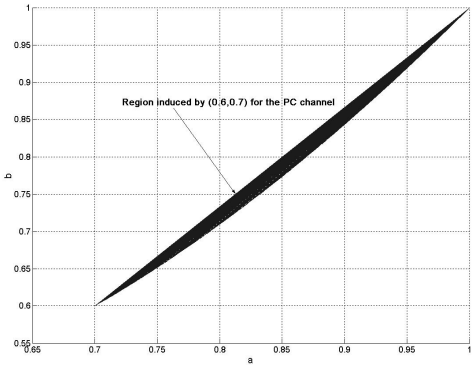

Figure 3: 2-dimensional convergence region induced in the (a,b) plane of the PC parameter space by the point (0.7,0.6)

*Remark 1:*Another technique for describing the convergence region, called state scrambling was introduced in [4]. This technique produces useful equations for the GE channel, but results in a trivial result for the PC channel.

6. Simulation Results

In this section, we present simulation results describing the performance of structured codes outlined in Section 3 over the PC channel. We use the cycle-invariant set S for the set of exponents in H given by Example 2a. The rate of the code for m=4 is R=0.75. Figure 4 shows the performance of these codes for a set of channel parameters and for joint iterative decoding and channel state estimation. The value of δ was fixed to 0.2 and σ was varied between 0.95 and 1 to generate pairs of *(a,b).* We transmitted 1000 code words of length 4608 each and to assure small complexity we performed only 5 iterations of message passing at the decoder to find the error rate (increasing the number of iterations to 20 gave negligible improvements). Figure 4 also compares the performance of the structured codes used against a random regular (3,12) code of length 4608 based on the construction given by Mackay et al [9] for the case with channel estimation. These results indicate that the structured codes based on cycle-invariant difference sets perform almost equivalently to the random constructions for the case of channels with memory, while having minimum storage requirements (reduced to memorizing 16 integers, as compared to the positions of 13824 ones). Figure 4 also shows the performance of the code described in Example 2b with and without estimator. It is seen that the estimator-decoder performs better for higher sigma values. There are several plausible explanations for this finding, such as the length of the code and the speed of convergence. It is possible to make the estimator-decoder perform better than the iterative decoder alone over a given range of σ values by adopting a suitable scheduling algorithm.

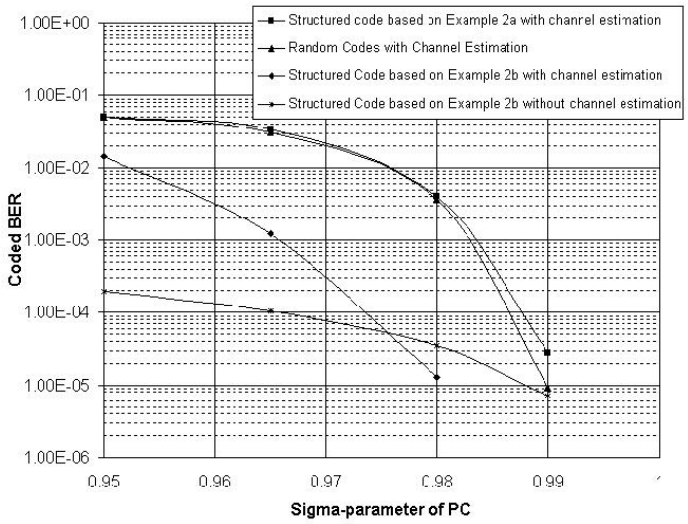

Figure 4: Performance of CIDS-based structured code in the PC channel with channel estimation as compared to random codes

Figure 5 shows the performance of the structured code with a channel state estimation vis-à-vis the case without channel state estimation for various values of the other channel parameter δ. It

is to be noted that the stationary probabilities of the channel states is a function of σ alone. On the other hand, δ characterizes the nature of the burst. Larger the value of δ, larger is the length of error bursts. So, for the same σ, the role of the channel estimator is more significant for larger δ. The simulation results in Figure 5 ratify the same point.

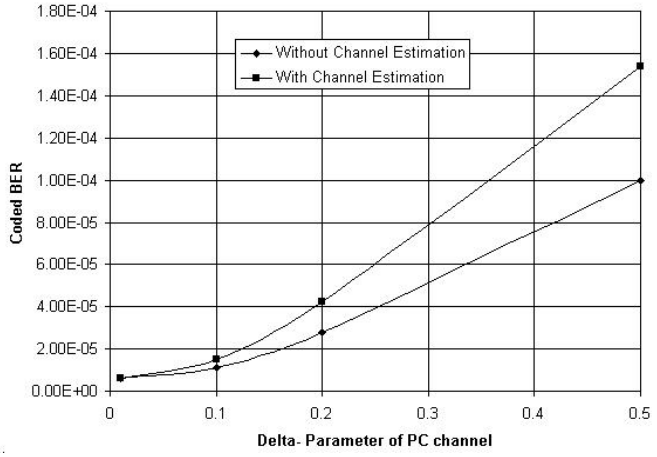

Figure 4: Error performance of the structured code from Eqn.2 with and without channel estimation Vs delta. Sigma is constant at 0.99

7. Generalized Urn Model Channels : Issues and Solutions

The idea of using urn models to describe a communication channel with memory can be extended to more general cases and is an interesting problem in its own right. One model currently under investigation considers error burst run lengths as numbered urns. The balls have two colors, indicating presence or absence of an error. They are also numbered to indicate the order in which the runs appear. The probability of choosing a ball of a given color will depend on the current urn occupancy. For such an urn model it is difficult to answer questions concerning urn occupancies. Most of the underlying problems in analysis arise because the occupancies are not independent. On the other hand, the dependence of the variables representing the occupancies enters only through the requirement that the number of distributed balls is fixed. This suggests "randomizing" the problem by redefining it in such a way as to make the distribution process stochastic, and the random variables specifying the occupancies independent. Thus a suitable choice of the Urn model channel can address and give exact solutions to several issues pertaining to channels with memory. For this purpose, the authors propose to use generalizations of probabilistic transforms such as the Stirling transform, binomial and negative binomial transforms, Bell transform, geometric transform.etc. developed by the authors in [11]. Thus a suitable choice for the parameter space coupled with these analytical tools can serve as a good and tractable model for wireless channels.

8. Conclusions

In this paper, we compared the performance of structured LDPC codes based on cycle-invariant difference sets with the performance of random codes for a simple channel with memory described in terms of the Polya-Urn model. We derived the equations for belief propagation for the combination of a channel state estimator and iterative decoder for the PC channel. We showed that these structured codes have almost the same performance as the random codes, but with significantly smaller storage complexity. We also provided a complete characterization of the convergence region of the PC channel and completed the characterization of the convergence region for a GE channel, by developing a new analytic technique termed state XOR-ing. We are currently looking into extending this work for more general Urn model channels and analyzing them using probabilistic transforms

References

[1] F.Alajaji and T.Fuja, "A Communication Channel modeled on Contagion," *IEEE Trans. Inform. Theory*, Vol. 40, No. 6., pp.2035-2041, Nov. 1994.

[2] A. Ashikmin, G. Kramer, and S. ten Brink, ``Extrinsic Information Transfer Functions: Model and Erasure Channel Properties," submitted to *IEEE Trans. Inform. Theory*, 2003.

[3] L.R.Bahl, J.Cocke, F.Jelinek, J.Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, Vol. 20, No.2, pp. 284-287, March 1974.

[4] A.W.Eckford, F.R.Kschischang and S.Pasupathy, "Analysis of Low-Density Parity Check Decoding in the Gilbert-Elliott Channel," *submitted to IEEE Trans. Inform. Theory*, May 2003.

[5] R. Gallager, *Low Density Parity Check Codes*, MIT Press, 1963.

[6] J. Garcia-Frias, "Decoding of Low-Density Parity-Check Codes over Finite-State Binary Markov Channels,'' *Proceedings of the ISIT 2001*, Washington DC, p.72, 2001.

[7] F.R.Kschischang, B.J.Frey, and H.-A. Loeliger, "Factor Graphs and the Sum-Product Algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no.2,pp. 498-519, Feb. 2001

[8]Y. Kou, S. Lin, and M.P.C. Fossorier, "Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results", *IEEE Trans. Inform. Theory*, Vol. 47, No. 7, pp. 2711-2736, Nov. 2001.

[9] D. MacKay, and R. Neal, "Good Codes Based on Very Sparse Matrices,'' Cryptography and Coding, 5th IMA Conference, Lecture Notes in Computer Science, C. Boyd, Ed., Vol. 1025, pp. 110-111, 1995, http://www.cs.toronto.edu/~mackay/CodesRegular.html.

[10] O. Milenkovic, K. Prakash, and B. Vasic, "Regular and Irregular Low Density Parity Check Codes for Iterative Decoding based on Cycle-Invariant Difference Sets," *to appear in the 41st Allerton Conference, Oct 2003.*

[11] O. Milenkovic, K.Compton, "Probabilstic Transforms for combinatorial Urn models", *accepted for publication in Combinatorics, Probability and Computing.*

[12] T.J. Richardson, M.A. Shokrollahi and R.L.Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Trans. Inform. Theory*, Vol. 47. No.2, pp. 619-637, Feb 2001.

[13] T.J. Richardson and R.L.Urbanke, "The capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform Theory*, Vol. 47, No.2, pp.599-618, Feb 2001.

[14] B.Vasic, amd O. Milenkovic, "Combinatorial Constructions of Low-Density Parity-Check Codes for Iterative Decoding," submitted to *IEEE Trans. Inform. Theory*.

[15] A. Worthen, and W. Stark, "Unified Design of Iterative Receivers Using Factor Graphs,'' *IEEE Trans. Inform. Theory*, Vol. 47, No. 2, pp. 843-849, Feb. 2001.

[16] J. Rosenthal, and P. O. Vontobel, "Construction of LDPC Codes Using Ramanujan Graphs and Ideas from Margulis," *Proceedings of the ISIT 2001,* Washington, p. 4, 2001.