Product Accumulate Codes on Fading Channels

Krishna R. Narayanan, Jing Li and Costas Georghiades
Dept of Electrical Engineering
Texas A&M University, College Station, TX 77843

Abstract

Product accumulate codes are a special case of differentially coded low density parity check (LDPC) codes. This work analyzes the noncoherent performance of PA codes and general differentially-coded LDPC codes on flat Rayleigh fading channels using extrinsic information transfer charts, and proposes a convergence-constraint method to design good LDPC ensembles matched to differential coding.

1 Introduction

Product accumulate (PA) codes are a serially concatenated codes whose inner code is a differential code and outer code is 2 parallel branches of single parity check codes (or a special type of low density parity check (LDPC) codes with only degree-1 and 2 variable nodes) [1]. This work investigates the performance of noncoherently detected PA codes on fading channels, and extend it to the general case where the outer code can be any LDPC code (i.e. differentially coded LDPC codes). The motivation is two-fold. First, previous work on PA codes has established them as a class of low-complexity, capacity-approaching good codes on additive white Gaussian noise (AWGN) channels [1]. Second, PA codes are inherently differentially coded which permits simple (noncoherent) differential detection.

The channel model we consider is flat Rayleigh fading channels with antipodal signaling. The received signal is given by $r_k = \alpha_k e^{j\theta_k} s_k + n_k$, where s_k, n_k, α_k and θ_k are the transmitted signals, the i.i.d. complex AWGN noise with zero mean and power spectrum density $N_0/2$ in each dimension, the Rayleigh fading amplitude with pdf $p_A(\alpha_k) = 2\alpha_k \exp(-\alpha_k^2)$ for $\alpha_k > 0$, and the channel phase with a uniform distribution over $[0, 2\pi)$, respectively. Further, the fading amplitudes and phases are correlated with auto-correlation $\mathcal{R}_k = \frac{1}{2}\mathcal{J}_0(2k\pi f_d T_s)$, where $f_d T_s$ is the normalized Doppler spread, and $\mathcal{J}_0(\cdot)$ is the 0_{th} order Bessel function of the first kind.

To preserve the simplicity of PA codes, instead of using complex multiple-symbol differential detection, we consider pilot symbol assisted modulation (PASM) and a simple iterative differential detection and decoding (IDDD) receiver which has been shown to perform stably at different Doppler rates [2]. In each decoding iteration, an estimation of the fading amplitude and phase is first conducted using a Wiener filter, followed by the "coherent" decoding of the inner differential code $1/(1 + D)$ and the outer code. Soft decision feedback is also used to assist the channel estimation. Detailed discussion on the receiver strategy can be found in [2].

We use extrinsic information transfer (EXIT) charts [4] to discuss a few interesting issues concerning noncoherent differential coding. First, we show that the popular practice of inserting pilot symbols to periodically terminate the differential trellis could cause additional performance loss and/or high error floors due to a "trellis segmentation" effect. Hence, a better way of inserting pilot symbols should be to separate them from the trellis structure. Second, in studying the convergence property of the iterative process, we show that while the performance/convergence behavior of the outer code of a (high-rate) PA code matches well with that of the differential code, a conventional LDPC code does not. Both analysis and simulations confirm that conventional LDPC codes perform better without a differential code than with one.

To further insight into what (outer) codes match well with differential coding, we propose and discuss a "convergence-constraint" method that uses density evolution on EXIT charts to optimize the degree profiles of LDPC codes for use with inner differential coding. Unlike the conventional "threshold-constraint" method that targets at the best asymptotic threshold, the convergence-constraint method controls the convergence behavior of the iterative interaction between the inner differential code and the outer LDPC code. We show that the proposed method is efficient and that the resulting optimal code is more that 1 dB better than the PA code. We expect the method to be useful in designing good LDPC ensembles matched with other inner decoder, demodulator and receiver.

2 EXIT Chart Analysis

In EXIT charts, the exchange of extrinsic information is visualized as a decoding/detection trajectory, which allows the prediction of the convergence and other performance behavior of the iterative process [3] [4]. We use mutual information between the code bit and the corresponding log-likelihood ratio (LLR) to depict the characteristics and relations of the component decoders . X-axis denotes the mutual information to the inner code (*a prior*) or from the outer code (extrinsic), denoted as $I_{a,i}/I_{e,o}$, and Y-axis the mutual information from the inner code or to the outer code, denoted as $I_{e,i}/I_{a,o}$.

[*Pilot Insertion*:] It is well-known that either insufficient or excessive pilot symbols could cause performance degradation. The former is due to poor channel estimation, and the latter is attributed to the fact that the performance gain obtained in channel tracking is not enough to compensate for the energy/rate loss caused by pilot symbols. However, little attention has been paid to the fact that improperly inserted pilots could cause an intrinsic loss in capacity in a differential code. Fig. 1(A) shows the popular practice of inserting pilots in a differential code. By periodically terminating the trellis, pilots here assume a dual role of channel estimation and $1/(1 \oplus D)$ decoding. Unfortunately, this is in fact not a good strategy since segmenting the trellis into small chunks causes a significant amount of short error events (an "inverse" effect of spectrum thinning), and consequently a loss in capacity. This "segmentation effect" is best illustrated using Fig. 2, where EXIT curves for the differential decoder with 0%, 4%, 10% and 20% pilots terminating the trellis are plotted for two different SNR values. We assume that the four curves in each family have the same energy per transmitted symbol, and that perfect channel information is known to the receiver (irrespect of the number of pilot symbols). Hence, the difference of the curves in each family is only due to the difference in pilot spacing. At the left end of the curves, we see that the curves with more pilot symbols are slightly better. This is because when there is little feedback information from the outer code, pilot symbols are the major contribution to *a priori* information. However, at the right end, when there is sufficient information provided by the outer code, pilot symbols are no longer an important source of *a priori* information. Rather, their negative impact of segmenting the trellis and shortening the (average) error events becomes dominant, causing a considerable performance loss. The performance degradation is more severe when more pilot symbols are inserted and when the code is operating at a lower SNR level. This suggests that the popular practice of terminating the trellis is not a good strategy and that a better way of inserting pilots may be to separate them the trellis as shown in Fig. 1(B). This is confirmed by the simulation result in Fig. 3 where more than 3 dB loss in performance is observed due to trellis segmentation for a rate 1/2, code length 64K PA code (solid lines assume perfect channel information and dashed line uses noncoherent detection). It is interesting to see that if we overlook the impact of the different strategies of pilot insertion, we might get the "surprising" result that noncoherent detection (dashed line) performs noticeably better than coherent detection (rightmost solid line)!

[*Codes Matched to Differential Coding*:] As mentioned before, the outer code of PA codes is a special type of LDPC code. Given PA codes perform well (especially at high rates), one tend to ask how a general LDPC code will perform with differential coding. This is an interesting question, since it directs to the solution of how to perform noncoherent detection with LDPC codes. Before we answer the question, we first note two important facts about EXIT analysis. First, in order for iterative decoding to converge successfully, the outer EXIT curve should be strictly below the inner EXIT curve, leaving an open passage between the curves. Second, the area under the EXIT curve, $\mathcal{A} = \int_0^1 I_e dI_a$, has shown to be closely related to the capacity of the code (we use "capacity" to loosely denote the information rate). When the *a priori* information is coming from the erasure channel and when the decoder is an optimal decoder, the area is exactly the capacity of the code [4]. For other channels, this may not be exact, but is nevertheless a good approximation as verified by empirical results. The implication of the above two facts is that, in order to fully achieve the capacity provided by the inner code, the outer code needs to have an EXIT curve closely matched in shape and in position to that of the inner code. Unfortunately, this not is the case of a conventional LDPC code (outer code) and a differential code (inner code). In Fig. 4, we plot a set of three outer EXIT curves corresponding to a regular LDPC code, an irregular LDPC code and the outer code of a PA code, and a set of two inner EXIT curves corresponding to a differential code (on correlated Rayleigh channel) and the plain Rayleigh fading channel. The regular LDPC code in the plot is (3,12)-regular, and the irregular one is optimized with variable node degree profile $\rho(x) = x^{20}$ and check node degree profile $\gamma(x) = 0.1510x + 0.1978x^2 + 0.2201x^6 + 0.0353^7 + 0.3958x^{29}$, which has a threshold of 0.6726 (about 0.0576 dB away from the AWGN capacity) [5]. We observe that while the outer code of (high-rate) PA codes shows a good match with an inner differential code, a conventional LDPC code (regular or irregular) will either intersect with the differential code (decoder failure) or leave a huge area between them (a waste in code capacity). The observation that LDPC codes match better with a plain channel than with a differential code indicates that, unless specifically designed, LDPC codes should not be used with a differential encoder (or more generally with any recursive inner code/modulation). Put another way, an LDPC code that is optimal in the conventional sense (i.e. BPSK modulated on memoryless channels) is not optimal when combined with an inner recursive code/modulation. However, not using differential coding typically requires more pilot symbols in order to track the channel well. Hence, it is expected that on (fast) fading channels where only limited bandwidth expansion is allowed, conventional LDPC codes do not perform well with noncoherent detection (whether or not a differential code is used). On the other hand, (high-rate) PA codes are able to make use of the intrinsic differential code for noncoherent detection, and are thus a better choice.

As a verification of the above EXIT analysis, Fig. 5 plots the performance curves of noncoherently detected PA codes and LDPC codes on Rayleigh channels. LDPC codes are evaluated either with or without a differential code and their degree profiles are the same as specified in Fig. 4. First, we see that the differentially-coded irregular LDPC code is more than 1.7 dB worse than its BPSK-coded peer at BER of 10^{-4}.

This confirms that (conventional) LDPC codes suffer a performance loss when used with a differential code. Second, while the performance gap between BPSK-coded irregular LDPC codes and PA codes is acceptable (about 0.5 dB) with 4% of pilot symbols, it becomes drastically large when pilot symbols are reduced in half, since 2% of pilot symbols are insufficient for non-differentially coded LDPC codes to track the channel. The observations are in good agreement with the EXIT analysis.

3 Code Design Matched to the Receiver

The above analysis leads to a more interesting problem: what LDPC ensembles are good for differential coding and how to optimize them? Below we proposes a "convergence-constraint" method that uses density evolution on EXIT charts to optimize (outer) LDPC degree profiles matched to an inner receiver. The proposed method focuses on the interaction (or the convergence behavior) between the inner and outer code during the iterative process, and is a useful extension of the conventional method for designing LDPC ensembles with good thresholds (call it "threshold-constraint" method) [6].

In order to design an outer code whose EXIT curve will match closely with the given inner EXIT curve, a nature and simple thinking is to "sample" the inner EXIT curve and design an EXIT curve that matches with these sample points (or the "control points"). Mathematically, if we choose a set of M control points in the EXIT chart, denoted as (v_i, w_i), $i = 1, 2, \cdots, M$, and if we use $\mathcal{T}_o(\cdot)$ to denote the input-output mutual information transfer function of the resulting LDPC code (exact expression of $\mathcal{T}_o$ will be defined later in (4)), the optimization problem can be formulated as

$$\max_{\substack{\sum \lambda_i = 1 \\ \sum \rho_j = 1}} \left\{ R = 1 - \frac{\sum \rho_j/j}{\sum \lambda_i/i} \;\middle|\; \mathcal{T}_o(w_k) \geq v_k,\ k = 1, 2, \cdots, M \right\},$$

where R denotes the code rate, λ_i and ρ_i denote the fraction of edges in the bipartite graph that are connected to variable nodes and check nodes of degree i. Collectively, we use $\lambda(x) = \sum \lambda_i x^{i-1}$ and $\rho(x) = \sum \rho_i x^{i-1}$ to describe the degree profiles from the edge perspective, and similarly, $\lambda'(x) = \sum \lambda'_i x^{i-1}$ and $\rho'(x) = \sum \rho'_i x^{i-1}$ from the node perspective [6], where $\lambda'_i = \frac{\lambda_i/i}{\sum \lambda_j/j}$, and $\rho'_i = \frac{\rho_i/i}{\sum \rho_j/j}$. The following functions are also useful for the discussion

$$\mathcal{I}(x) \triangleq 1 - \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi x}} e^{-\frac{(z-x)^2}{4x}} \log(1 + e^{-z}) dz, \quad (1)$$

$$\phi(x) \triangleq \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int \tanh\frac{z}{2}\, e^{-\frac{(z-x)^2}{4x}} dz, & x > 0, \\ 1, & x = 0. \end{cases} \quad (2)$$

The code design process is a dual constraint optimization process that progressively optimizes $\lambda(x)$ and $\rho(x)$ based on the other. Below we discuss only the optimization of $\lambda(x)$ for a given $\rho(x)$. The optimization of $\rho(x)$ can be derived similarly.

Under the assumption that the messages passed along all edges are i.i.d. and Gaussian distributed, the average messages variable nodes receive from their neighbors are mixed Gaussian distributed. From $(l-1)_{th}$ iteration to l_{th} local iteration (in the LDPC decoder), the mean of the messages associated with the variable node, m_v, evolves as

$$m_v^{(l)} = \sum_i \lambda_i \phi\Big(m_0 + (i-1)\sum_j \rho_j \phi^{-1}(1 - (1 - m_v^{(l-1)})^{j-1}\Big),$$

where m_0 denotes the mean of the initial messages received from the channel (or the inner code). Let us denote

$$h_i(m_0, r) \triangleq \phi\Big(m_0 + (i-1)\sum_j \rho_j \phi^{-1}\big(1 - (1-r)^{j-1}\big)\Big),$$

the evolution of the message mean associated with variable nodes can then be described as $r_l = h(m_0, r_{l-1}) \triangleq \sum_i \lambda_i h_i(m_0, r_{l-1})$. The conventional threshold-constraint density evolution forces the resulting code to converge to the zero-error state for a given m_0 by setting $r > h(m_0, r)$ for all $r \in (0, \phi(m_0)]$ [6]. This has implicitly use a control point $(v, w) = (1, \mathcal{I}(m_0))$, i.e., the resulting EXIT curve will stay strictly below point $(1, \mathcal{I}(m_0))$. In general, a control point (v, w) can choose any value from 0 to 1, and the above condition is relaxed to $r > h(m_0, r)$ for all $r \in (r^*, \phi(m_0)]$, where r^* is the critical value that ensures $\mathcal{T}_o(w) \geq v$. Formally, the problem is stated as: given a check node degree profile $\rho(x)$ and a control point (v, w), where $0 \leq v, w \leq 1$,

$$\max_{\sum_i \lambda_i = 1} \quad \sum_i \lambda_i / i, \quad (3)$$

subject to: (i) $\sum_i \lambda_i = 1$,

$$(ii) \sum_i \lambda_i \big(h_i(m_0, r) - r\big) < 0,\ \forall r \in (r^*, \phi(m_0)],$$

where $m_0 = \mathcal{I}^{-1}(w)$ and r^* satisfies

$$\mathcal{T}_o(w) \triangleq \sum_i \lambda'_i\, \mathcal{I}\Big(i \sum_j \rho_j \phi^{-1}\big(1 - (1 - r^*)^{j-1}\big)\Big) \geq v. \quad (4)$$

For a set of M control points, (v_1, w_1), (v_2, w_2), $\cdots$, (v_M, w_M), we can combine the constraints associated with each individual control point and perform a joint optimization on all of them, which will result in an EXIT curve whose shape and position are closely match to the control points.

Note that the above constraint (ii) is a nonlinear function of λ_i's, and that the computation of r^* from (4) requires the knowledge of $\lambda(x)$, which is yet to be optimized. To get around with this, one possible approach is to consider an approximation of $\lambda(x)$ in (4) to compute r^*. Specifically, we consider only the two lowest degree variable nodes λ_{i_1} and λ_{i_2}, and approximate the degree profile as $\tilde{\lambda}(x) \approx \lambda_{i_1} x^{i_1 - 1} + \lambda_{i_2} x^{i_2 - 1}$.

In a conventional LDPC ensemble, $i_1 = 2$, i.e., degree-1 nodes are not allowed, since the outbound messages from these nodes do not improve in the message-passing decoding. However, when an LDPC code is used together with a differential code (or other inner code and/or modulation with memory), weight-1 nodes in the outer LDPC decoder will get extrinsic information from the inner code as the iteration progresses and their estimates will improve accordingly. In this case, the first and the second nonzero λ_i's are λ_1 and λ_2. An analytical bound on λ_1' is difficult, but empirical results show that $\lambda_1' \leq 1-R$ is a reasonable assumption[1]. This is because, otherwise there are at least two degree-1 variable nodes, say the p_{th} and q_{th} node, connecting to the same check, which creates a very vulnerable link. As shown in Fig. 6, when the four bits denoted by solid circles flip altogether, another valid codeword results and the decoder is unable to detect. In other words, for any finite length construction, the minimum distance of this LDPC ensemble is (at the most) 4, which is not desirable. Using the approximation $\tilde{\lambda}(x) = (1-R) + Rx$ in (4), we are able to compute (a lower bound of) r^* to be used in constraint (ii). Code design is thus solvable using linear programming. Experiments show that the optimized EXIT curve has a shape as desired, but the position is slightly lower, i.e. code rate is slightly pessimistic. This can be compensated by pre-setting the control points slightly higher than we actually want them to be.

[*Optimization Results*:] We observe that the LDPC ensemble optimal for differential coding always contains degree-1 and degree-2 variable nodes. For high rate codes above 0.75, these nodes are dominant, or in some cases the only types of variable nodes; for medium rates around 0.5, there are also a good portion of high-degree variable nodes. Hence, it is fair to say that the degree profile of the outer code of high-rate PA codes is (near-)optimal for differential coding. The optimization result of the target rate 0.5 is shown in Fig. 7. The resulting LDPC ensemble has rate $R = 0.5037$ and degree profile $\lambda(x) = 0.0672 + 0.4599x + 0.0264x^8 + 0.0495x^9 + 0.0720x^{10} + 0.0828x^{11} + 0.0855x^{12} + 0.0807x^{13} + 0.0760x^{14}$ and $\rho(x) = x^5$. We see that it matches closely with the noncoherent receiver operating at 0.25 dB. Accounting for the rate of the outer code, we see that the resulting differentially-coded LDPC ensemble requires $0.25 - 10\log_{10}(0.5037) = 3.2283$ dB (asymptotically) in order for the iterative differential detection and decoding to converge successfully. Compared to a rate 0.50 PA code which requires $1.26 - 10\log_{10}(0.5) = 4.2703$ dB, the optimized LDPC ensemble is about 1.04 dB better asymptotically. Simulation results show a good agreement with the analytical result (Fig. 8), and the 64K long code performs about 0.75 dB away from the analytical threshold at BER of 10^{-4}.

[1] The exact code rate is dependent on the optimization result, but we know of the target code rate which is in the vicinity of the final code rate.

4 Conclusion

The major conclusions and contributions of this paper are as follows: First, we show that the popular practice of inserting pilots to periodically terminate the trellis incurs an intrinsic loss in code capacity and is likely to cause severe BER performance loss to the overall code performance. A better way of inserting pilot symbols is suggested which is to separate pilots from trellis. Second, we investigate the performance of conventional LDPC codes using noncoherent detection. We show that conventional LDPC codes suffer a performance loss when used with an inner differential code, yet without the differential code, more pilot symbols are needed to track the channel. Hence, it is fair to say that noncoherently detected LDPC codes do not perform as desirably as the coherent case. Finally, we propose a convergence-constraint method to design good LDPC ensembles matched with differential coding (and in general any receiver). We observe that the LDPC ensemble optimal for differential coding always contains degree-1 and 2 variable nodes, and that for high code rates, these nodes are dominant. The resulting optimal LDPC code shows a 1.04 dB gain over the existing PA code. It is worth mentioning that optimal differentially-coded LDPC codes are in fact (optimal) irregular repeat accumulate (IRA) codes [7], but the proposed optimization procedure has a far-reaching implication and application since it has explicitly taken into account the property and the imperfectness of the receiver.

References

[1] J. Li, K. R. Narayanan and C. N. Georghiades, “Product accumulate codes: a class of capacity-approaching, low complexity codes,” submitted to *IEEE Trans. Inform. Theory*, 2001.

[2] J. Li, K. R. Narayanan and C. N. Georghiades, “On the Performance and Analysis of Product Accumulate Codes over Rayleigh Fading Channels with Coherent and Noncoherent Detection,” in prepartion, 2002.

[3] S. ten Brink, “Convergence behavior of iteratively decoded parallel concatenated codes,” *IEEE Trans. on Commun.* , vol. 49, No. 10, pp. 1727-1737, Oct. 2001.

[4] A. Ashikhmin, G. Kramer, and S. ten Brink, “Extrinsic information transfer functions: a model and two properties,” *Proc. Conf. Inform. Sciences and Systems*, Princeton University, March 2002.

[5] `http://lthcwww.epfl.ch/research/`

[6] S-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 657-670, Feb. 2001.

[7] H. Jin, A. Khandekar and R. J. McEliece, “Irregular repeat-accumulate codes,” *Proc. 2nd Intl. Symp. on Turbo Codes and Related Topics*, pp. 1-5, France, 2000.

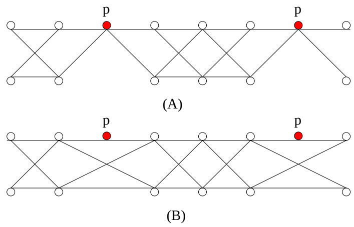

Figure 1: Different strategies of pilot insertion in a differential code.

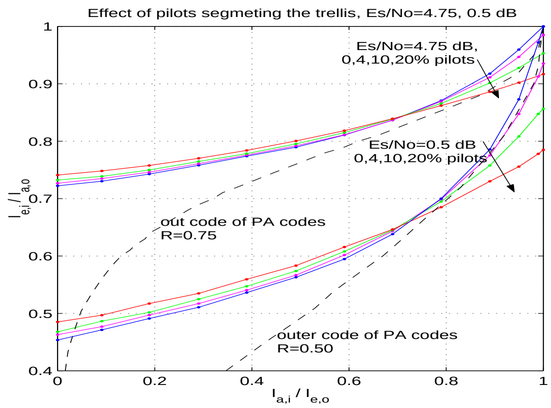

Figure 2: The effect of pilot symbols segmenting the trellis.

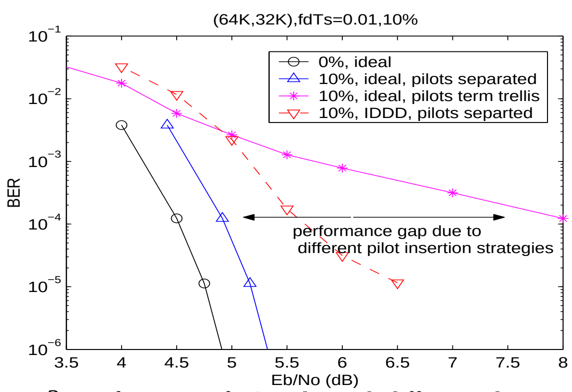

Figure 3: Performance of PA codes with different pilot insertion strategies.

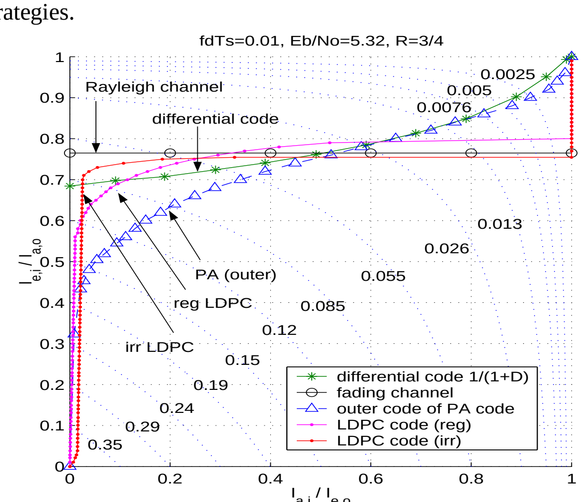

Figure 4: EXIT curves of LDPC codes and PA codes.

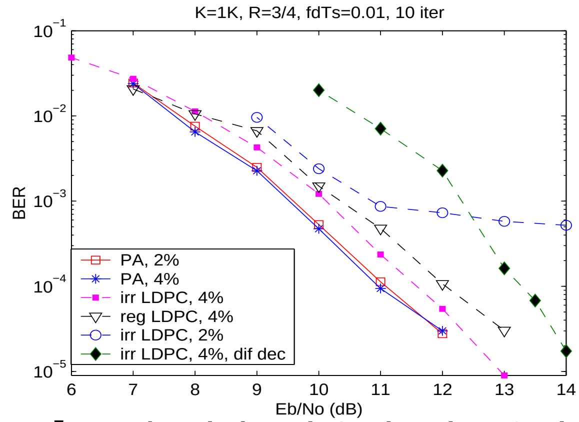

Figure 5: Noncoherently detected PA codes and LDPC codes on fast Rayleigh fading channels.

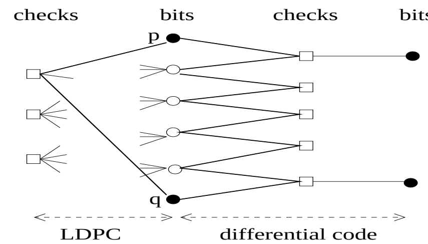

Figure 6: Defect in code structure when $\lambda_1' > 1 - R$.

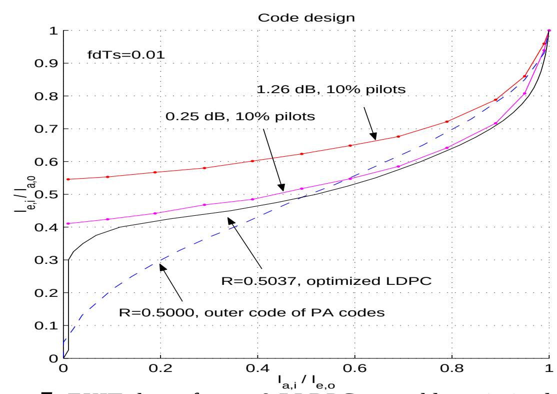

Figure 7: EXIT chart of a rate 0.5 LDPC ensemble optimized using convergence-evolution for differential coding.

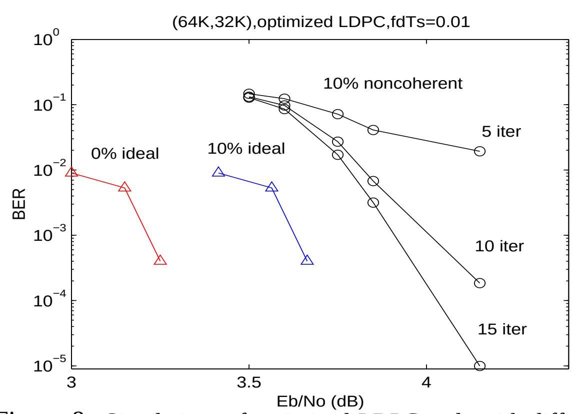

Figure 8: Simulations of optimized LDPC code with differential coding and iterative differential detection and decoding.