A.Durga Prakash, I.V.G.Manohar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue4, July-August 2012, pp.1583-1591

A Design Of Low Density Parity Check Algorithm For Insertion/Deletion Channels

A.DURGA PRAKASH
M.Tech Student Scholar, DECS,
Dept of Electronics and Communication Engineering,
Nalanda Institute of Engineering and technology,
Sattenapalli (M); Guntur (Dt); A.P, India

I.V.G.MANOHAR
M.Tech, Asst Professor
Dept of Electronics and Communication Engineering,
Nalanda Institute of Engineering and technology,
Sattenapalli (M); Guntur (Dt); A.P, India

Abstract

Low-density parity-check (LDPC) code is a linear error correcting code, a method of transmitting a message over a noisy transmission channel, and is constructed using a sparse bipartite graph. LDPC codes are capacity-approaching codes, LDPC codes are finding increasing use in applications requiring reliable and highly efficient information transfer over bandwidth or return channel–constrained links in the presence of data-corrupting noise. In this Paper we investigating a promising technique for insertion, deletion and substitution errors. Information exchange between the inner decoder and outer decoder is not allowed in iterative decoding. Through numerical evaluations, we first find the marker code structures which offer the ultimate achievable rate when standard bit-level synchronization are performed. The output results which confirm the advantage of the newly designed codes over the ones optimized for the standard additive white Gaussian noise (AWGN) channels, In AWGN channel severe synchronization problems

Keywords- Insertion/deletion channel, marker codes, synchronization, LDPC code design.

I. INTRODUCTION

Low-density parity-check (LDPC) codes are a class of linear block LDPC codes. The name comes from the characteristic of their parity-check matrix which contains only a few 1's in comparison to the amount of 0's. Their main advantage is that they provide a performance which is very close to the capacity for a lot of different channels and linear time complex algorithms for decoding. Furthermore are they suited for implementations that make heavy use of parallelism. LDPC codes are finding increasing use in applications requiring reliable and highly efficient information transfer over bandwidth or return channel–constrained links in the presence of data-corrupting noise.[1]. Although implementation of LDPC codes has lagged behind that of other codes, notably turbo codes, the absence of encumbering software patents has made LDPC attractive to some. LDPC codes were forgotten until Gallager's work was discovered in 1996. Turbo codes, another class of capacity approaching codes discovered in 1993, became the coding scheme of choice in the late 1990s, used for applications such as deep space satellite communications.

However, in the last few years, the advances in low-density parity-check codes have seen them surpass turbo codes in terms of error floor and performance in the higher code rate range, leaving turbo codes better suited for the lower code rates only. A memory less channel with synchronization errors is defined as a channel in which each input symbol independently of other symbols is transformed into a word of random (including also zero) length, and at the output of the channel the ordinal number of the input symbol from which the given output symbol was obtained is unknown[2]. For such channels Shannon's theorem on transmission rates for which noise stable coding methods exist, is formulated and proved. The binary symmetric channel, where each bit is independently received in error with probability p, and the binary erasure channel, where each bit is erased with probability p, enjoy a long and rich history. Shannon developed the fundamental results on the capacity of such channels in the 1940's, and in recent years, through the development and analysis of low-density parity-check codes and related families of codes, we understand how to achieve near-capacity performance for such channels extremely efficiently[3].

Before beginning, it is worth asking why this class of problems is important. From a strictly practical perspective, such channels are arguably harder to justify than channels with errors or erasures. While codes for synchronization have been suggested for disk drives, watermarking, or general channels where timing errors may occur, immediate applications are much less clear than for advances in erasure-correcting and error-correcting codes. However, this may be changing. In the past, symbol synchronization has been handled separately from

coding, using timing recovery techniques that were expensive but reasonable given overall system performance [4]-[5]. Indeed, even the model we suggest assumes some high-level synchronization, as both sender and receiver know that n bits are being sent in a transmission; still, the model appears most natural and appropriate, and there is clear goal in handling transmission sizes n efficiently and with a high coding rate.

With recent improvements in coding theory, it may become increasingly common that synchronization errors will prove a bottleneck for practical channels. Because we are currently so far away from having good coding schemes for even the most basic synchronization channels, in practice coding is rarely if ever considered as a viable solution to synchronization. If efficient codes for synchronization problems can be found, it is likely that applications will follow. If such codes are even a fraction as useful as codes for erasures or errors have been, they will have a significant impact [6]-[9]. The work we describe on capacity lower bounds demonstrates that there is more potential here than has perhaps been realized.

Of course, coding theory often has applications outside of engineering, and channels with deletions and insertions prove no exception, appearing naturally in biology. Symbols from DNA and RNA are deleted and inserted (and transposed, and otherwise changed) as errors in genetic processes. Understanding deletion channels and related problems may eventually give us important insight into these genetic processes.

The paper is organized as follows. In the next section, the system model is described. In Section III, we review the standard bit-level MAP detection algorithm and numerically evaluate the ultimate rate achievable by interleaved concatenated coding schemes. In Section IV, we introduce the symbol-level MAP detection algorithm and compare the relevant achievable rates with those characterizing the standard bit-level approach. Error-rate results for a practical LDPC coded scheme are also reported for both bit-level and symbol level detection[10]-[15]. The EXIT chart-based LDPC code design process for the insertion and deletion channels is provided in Section V along with example designs. Finally, concluding remarks are given in Section VI.

II. SYSTEM DESCRIPTION

We consider transmission over binary channels impaired by insertion, deletion, and substitution errors, according to the model proposed . Let $x_1^T = \{xk\} Tk=1$ and $y_1^R = \{yn\} R\ n=1$ be the sequences of bits at the channel input and channel output, respectively, where the number of transmitted bits T is a constant system parameter while the number of received bits R is a random variable depending on the realization of the insertion/deletion process. We can think of the channel as the cascade of two sub-channels where, in the first sub-channel, each input bit gets deleted (with probability P_d), or experiences an insertion error (with probability P_i), or is correctly transmitted (with probability $P_t = 1 - P_d - P_i$), while the second sub-channel is a BSC with substitution probability P_s.[1] As proposed an input bit experiencing an insertion error gets replaced by two uniformly distributed random bits, we point out that different models for the insertion process exist. We assume that insertion, deletion, and substitution errors are all IID, and that the transmitter and the receiver have no information on the positions at which the errors occur[16].

We adopt the coding scheme depicted in Figure 1, which consists of the interleaved serial concatenation of an outer error-correcting code with an inner marker code. Specifically, the information bits are first encoded by means of a powerful channel code (e.g., a turbo or LDPC code), then the transmitted sequence is formed by inserting pilot bits, which are often referred to as markers, to the interleaved sequence of coded bits. The marker bits and their positions in the transmitted sequence are known to the receiver, which exploits this information in the MAP detector to recover the synchronization errors due to insertions/deletions, as explained later. For simplicity, we only focus on the case of regular marker codes with rate[17].

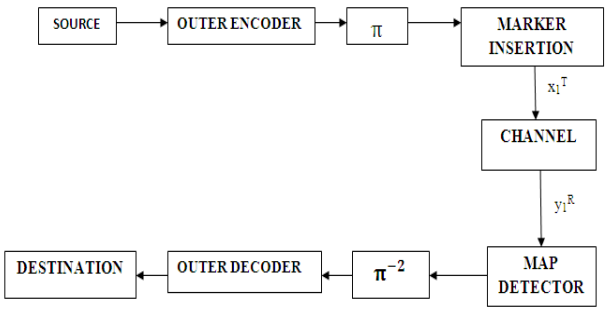


Figure .1 Block diagram of the considered concatenated coding scheme. Interleaving and deinterleaving blocks are denoted by Π and Π^{-1}, respectively.

$$r_M = \frac{N_C}{N_C + N_M}, \quad (1)$$

i.e., the case when the same marker consisting of NM consecutive bits is inserted every N_C bits at the output of the outer encoder. Hence, if the outer code rate is denoted by r_C, the overall code rate is $r = r_C r_M$. We notice that the same coding scheme was considered. while a similar scheme adopting watermark codes instead of marker codes was considered. At the receiver side, given the a

priori log-likelihood ratios (L-values) log (x_k=0)/(x_k=1) , the MAP detection is first executed to generate the conditional probability $\xi_k(x_k) = P(y_1^R|x_k)$ for $k \in \{1, 2, \ldots, T\}$ and $x_k \in \{0, 1\}$ by exploiting the perfect a priori information from the marker code. Then the extrinsic information on the transmitted bits can be easily obtained as log $(y_1^R|x_k=0)/(y_1^R|x_k=1)$ = log $\xi_k(0)/\xi_k(1)$. After being deinterleaved, the a posteriori information, i.e., the sum of a priori and extrinsic L-values, feeds the outer decoder, which finally generates an estimate of the information bits. We point out that decoding performance can be improved by adopting iterative schemes based on the exchange of extrinsic information between the MAP detector and the outer decoder. But, since the MAP detector is typically the bottleneck of the receiver in terms of latency, we assume that the MAP detection is executed only once in Sections III and IV. Iterative detection/decoding is considered in Section V where specific outer code designs are pursued9[18].

III. BIT-LEVEL SYNCHRONIZATION

Let us first review the bit-level MAP detection algorithm for the considered channel model. The algorithm, which already appeared with some differences in the channel model, is similar to the general forward backward algorithm (FBA), but it cannot be derived by means of the standard approach discussed because the channel model is not a finite-state Markov chain. According to the turbo principle, the code constraints induced by the outer code are neglected in the derivation of the algorithm, and the bits x_1^T are considered to be statistically independent, namely the *a priori* probability $P(x_1^T \Pi)$ is factorized as π_1^T =1 $P(x_k)$, where $P(x_k)$ is 1/2 if x_k is a code bit, while it is 0 (or 1) if x_k is a pilot bit.

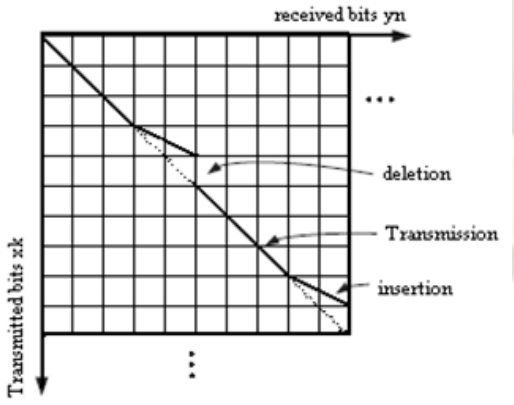


Figure 2: Synchronization represented by a path on a two dimensional grid

A. Bit Level MAP Detection:

Let us define the binary event $D_{k,n}$, with $k \in \{1, 2, \ldots, T\}$, and $n \in \{0, 1, \ldots, R\}$, which denotes whether, of the first k transmitted bits, exactly n bits are received, possibly after being corrupted by the channel or not. We are interested in the exact "frame synchronization" scenario, in which $D_{0,0}$ and $D_{T,R}$ are true with probability one, the values of T and R being known to the receiver. This assumption is not critical since frame synchronization can be obtained with great accuracy. For a better illustration of the resynchronization process, a two-dimensional grid is created to represent the synchronization errors. As the rows and columns on the grid correspond to the transmitted and received bits x_k, $k \in \{1, \ldots, T\}$ and y_n, $n \in \{1, \ldots, R\}$, respectively. The solid line refers to a particular channel realization and the dotted line indicates the channel without any insertion or deletion errors. There are only three possible moves to reach a certain state. A diagonal move from the top left corner to the bottom right corner on the grid indicates a successful transmission, i.e., no insertion or deletion, but the bit may not be correctly received. An insertion event is represented by a diagonal move in two adjacent blocks and a vertical move denotes a deletion event. Let us also define the function and the coefficients[19].

$$F(x_n, y_n) = \begin{cases} 1 - P_s & \text{if } y_n = x_k \\ P_s & \text{if } y_n \neq x_k \end{cases} \quad (2)$$

$$\alpha_k(n) = P(y_1^n, D_{k,n}), \quad (3)$$

$$\beta_k(n) = P(y_{1+1}^R \mid D_{k,n}) \quad (4)$$

These coefficients can be computed by means of the following forward recursion (where the differences with respect to are due to the adopted channel model being different):

$$\alpha_k(n) = \frac{P_i}{4}\alpha_{k-1}(n-2) + P_d\alpha_{k-1}(n) + P_t\alpha_{k-1}(n-1)\sum_{x_k} P(x_k)F(x_k, y_n) \quad (5)$$

and the following backward recursion:

$$\beta_k(n) = \frac{P_i}{4}\beta_{k+1}(n+2) + P_d\beta_{k+1}(n) + P_t\beta_{k+1}(n+1)\sum_{x_k} P(x_{k+1})F(x_{k+1}, y_{n+1}) \quad (6)$$

which are both initialized by exploiting the "frame synchronization" assumption. Finally, the target conditional probability can be computed as

$$P(y_1^R \mid x_k) = \frac{P_i}{4}\sum_{n=0}^{\min(2k,R)}\alpha_{k-1}(n-2)\beta_k(n) + P_d\sum_{n=0}^{\min(2k,R)}\alpha_{k-1}(n)\beta_k(n) + P_t\sum_{n=0}\alpha_{k-1}(n-2)\beta_k(n)F(x_k, y_n) \quad (7)$$

B. **Achievable Rates by a Specific Marker Code***:*

An interesting information-theoretic problem that arises is the following: what is the ultimate rate at which we can reliably transmit information through the considered concatenated coding scheme? An approximate solution to this problem can be found where the authors investigate the BCJR-once bound and characterize the capacity of a BSC with a time-varying substitution probability and conjecture that it provides an accurate characterization of the information rate. Here, we pursue a more precise solution to the problem. First, we notice that the ultimate rate rC for the outer code that can be achieved through the considered concatenated coding system is given by the mutual information between the independent and uniformly distributed bits at the input of the interleave at the transmitter side and the soft information at the output of the deinterleaved at the receiver side (see Figure. 1). Because of the complicated MAP detector, this mutual information cannot be computed in closed form, but it can be easily evaluated through Monte Carlo simulations with a large number of channel realizations by obtaining the histogram of the distribution of the extrinsic information (L-values).[20]-[21]. The reason we choose histograms instead of the Arnold-Loeliger algorithm is that the latter only gives the no-interleaving mutual information while our focus is mainly on interleaved systems using a soft damper, as discussed in Section IV. Interestingly, this numerical method is equivalent to the evaluation of the EXIT chart for the MAP detector, particularly of its left-most point . In fact, the left-most point of a detection EXIT chart gives the ultimate rate achievable by the outer code when it is concatenated with the inner detector through an interleave and iterative detection/decoding is not allowed. Hence, for a given marker code with rate r_M, we can evaluate the ultimate value of r_C by means of this numerical method, and then compute the ultimate overall rate as $r = r_C r_M$. We will exploit this result in the next subsection to find optimal marker codes for channels with insertions and deletions.

C. **Marker Code Optimization***:*

In this section, we study the problem of selecting a good marker code. We first notice that a lower marker code rate or smaller NC leads to better synchronization capabilities since the positions of the insertions and deletions can be located more precisely; however, this is obtained with an increased overhead. This argument suggests that an optimal marker code rate rM exists for different marker codes used over an insertion/deletion channel.

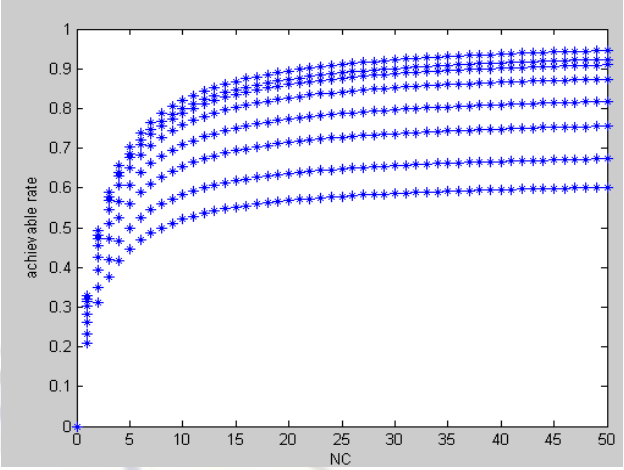


Figure. 3. Achievable rates for different deletion channels for the marker "01" inserted every Nc bits

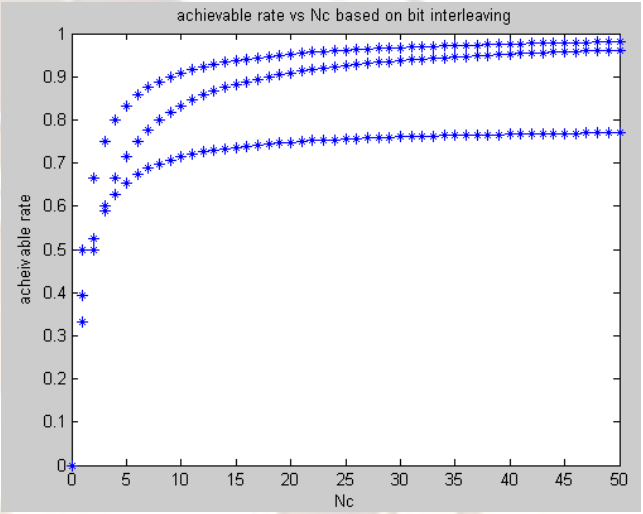
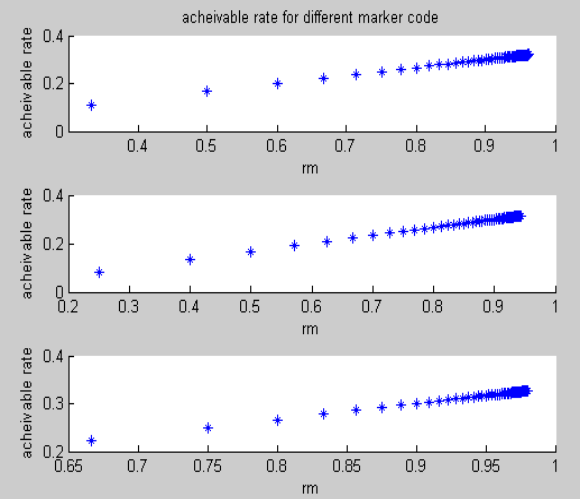


Figure 4. Achievable rates for different insertion and deletion channels for the marker "01" inserted every Nc bits

Some results obtained by means of the proposed information rate evaluation method of the previous subsection. Particularly, in Figure 3, it is shown how the overall rate varies, for different deletion channels ($P_i = P_s = 0$), as a function of N_C, when the two bit marker "01" is inserted every N_C information-carrying bits[22]. For each value of the deletion probability P_d, a clear maximum is obtained, which determines the marker code rate that is information-theoretically optimal. Not surprisingly, as deletions become more frequent, the achievable rate decreases, so does the rate of the optimal marker code, since an effective synchronization process requires more pilot bits. As another example, compares the impacts of insertion, deletion and substitution errors on the achievable rates with the constraint that $P_i + P_d + P_s = 0.03$. It is clear that for this particular example, the deletion errors cause more severe damage than the insertion errors to the performance while the substitution errors degrade the capacity much less than the synchronization errors. Note that these achievable rates are only valid if a

single synchronization stage is employed (single-pass decoding) and they are violated when an iterative decoding/synchronization scheme is adopted. We also note that the gap to the existing Shannon capacity lower bounds is also large. For instance, a lower bound for the capacity of an i.i.d. deletion channel with Pd = 0.05 is 0.728 while the maximum rate found is less than 0.6.



The proposed approach can be used not only to find the optimal rate for a given marker code, but also to compare different marker codes. As an example, the marker code "00" is clearly not a good choice compared to "01" since there is no transition between the two bits and the receiver cannot determine as precisely whether an insertion or deletion error happens prior to the specific marker. On the other hand, for "01", there is a transition in the marker sequence and a single deletion or insertion can be easily identified. In Figure. 5, we compare four regular marker codes obtained by inserting the markers "0", "01", "001", and "010" every NC information bits, for the case of a deletion/substitution channel with P_d = P_s = 10−2. The results, which are given in terms of the overall rate r as a function of the marker code rate r_M, show that for this particular channel the best choice of marker code among the three candidates is to insert the pilot bits "01" every 18 information bits, which provides an overall rate of about 0.75. It is also not surprising to see that the marker "001" outperforms "010" for higher marker code rates. This is attributed to the following: it is more likely that more than one bit get deleted between two adjacent markers and hence the marker "010" may not be able to detect these synchronization errors while the marker "001" still can. We conclude this section by noting that for all the studied scenarios, the guidelines for marker code design that we obtain through our analysis are in good agreement with the approximate analysis proposed[23].

IV. SYMBOL-LEVEL SYNCHRONIZATION

One key observation is that, since insertion/deletion channels have memory, the soft information at the output of the MAP detector corresponding to two bits with different time indices is correlated. Hence, information is lost when such correlations are neglected, which is exactly what is done in our concatenated system because of the presence of the interleaver/deinterleave [26]. On the other hand, interleaving is fundamental because it allows us to split the decoding process into two serial steps, namely the inner detection and the outer decoding; the other option being joint detection/decoding, which would be computationally infeasible. In the following, we propose a solution that allows us to recover part of the information loss while preserving the interleaving process, hence also its advantage of splitting the decoding process into an inner detection and an outer decoding.

A. Symbol Level MAP Detection:

We introduce MAP detection at the symbol level, defining a symbol as a group of m consecutive bits. Consequently, the T transmitted bits are partitioned into **Ts** symbols $S_k = X_{m(k+1)}^{mk}, k\epsilon\{1,2,\ldots\ldots,T_S\}$, taking values on $\{0,1\}^m$. The last symbol, however, may consists of less than m bits, but we assume that T/m = **Ts** is an integer for simplicity. In this case, synchronization can be carried out by means of a symbol-level FBA which is obtained by extending the bit level derivation given to the symbol-level case. In the following, we provide the details of the algorithm.

Let us re-define the binary event $D_{k,n}$ with $k \epsilon \{1,2,\ldots\ldots T_S\}$ and $\epsilon \{1,2,\ldots\ldots R\}$ which denotes whether, of the first k transmitted symbols (i.e., km bits), exactly n bits are received or not, possibly after being corrupted by the channel. With this redefinition of the event $D_{k,n}$ the definitions in (2) and (3) still hold. As in the bit-level case, the coefficients can be computed by means of the forward/backward recursions. For simplicity, we give here the formulations for the case m = 2, i.e., bits $\{x_{2k-1}, x_{2k}\}$ are grouped as one symbol, noting that the extension to the case of m > 2 is straightforward. In this case, there are 9 possible ways to reach a certain state on the trellis, and the resulting recursions are given as shown in (7) and (8), respectively, and are both initialized by exploiting again the exact "frame synchronization" assumption. Finally, the target extrinsic information can be computed as shown in (9).

B. Achievable Rate Improvement with Symbol Level Synchronization:

As an example use of the proposed algorithm compares the mutual information between

the symbols at the input of the interleaver at the transmitter side and the soft information at the output of the deinterleaver at the receiver side, for the case of one-bit symbols and two-bit symbols. Specifically, it is shown how the overall achievable rate varies, for an IID deletion channel ($Pi = Ps = 0$, $Pd = 0.01$), as a function of, N_C when the two-bit marker "01" is inserted every N_C information-carrying bits. For comparison, the mutual information computed in the absence of interleaving, i.e., by evaluating the expectations $E[\log P(y_1^R)]$ and $E[\log P(x_1^T, y_1^R)]$ using Monte Carlo techniques with a large number of channel simulations, and obtaining $I(x_1^T; y_1^R)$ as $T - E[\log P(y_1^R)] + E[\log P(x_1^T, y_1^R)]$ is also shown[25]-[26].

$$\alpha_k(n) = P_d^2\,\alpha_{k-1}(n) + P_d P_t \alpha_{k-1}(n-1)\sum_{i=0}^{1}\sum_{x_{2k-1}} P(x_{2k-i}, y_n)$$

$$+P_t^2\,\alpha_{k-1}(n-1)\sum_{x_{2k-1}} P(x_{2k-1})\,F(x_{2k-1}, y_{n-1}).\sum_{x_{2k}} P(x_{2k})\,F(x_{2k}, y_n)$$

$$+\frac{P_i}{4}P_t\alpha_{k-1}(n-2).2+\frac{P_i}{16}\alpha_{k-1}(n-4)$$

$$+\frac{P_i}{4}P_t\alpha_{k-1}(n-3)\sum_{i=0}^{1}\sum_{x_{2k-1}} P(x_{2k-i})\,F(x_{2k-i}, y_{n-2i})$$

(8)

$$\beta_k(n) = P_d^2\,\beta_{k+1}(n) + P_d P_t \beta_{k+1}(n+1)\sum_{i=1}^{2}\sum_{x_{2k+1}} P(x_{2k+i})F(x_{2k+i}, y_{n+1})$$

$$+P_t^2\,\beta_{k+1}(n+2)\sum_{x_{2k-1}} P(x_{2k+1})\,F(x_{2k+1}, y_{n+1}).\sum_{x_{2k+2}} P(x_{2k+2})\,F(x_{2k+2}, y_{n+2})$$

$$+\frac{P_i}{4}P_t\beta_{k-1}(n+2).2+\frac{P_i}{16}\beta_{k+1}(n+4)$$

$$+\frac{P_i}{4}P_t\beta_{k+1}(n+3)\sum_{i=0}^{2}\sum_{x_{2k+1}} P(x_{2k+i})\,F(x_{2k+i}, y_{n+2i-1})$$

(9)

$$p(y_1^R|x_{2k-1}, x_{2k}) = P_d^2\sum_{n=0}^{\min(4k,R)}\alpha_{k-1}(n)\,\beta_k(n)$$

$$+P_d P_t\sum_{n=0}^{\min(4k,R)}\sum_{i=0}^{1}\alpha_{k-1}(n-1)\,\beta_k(n)F(x_{2k-i}, y_n)$$

$$+P_t^2\sum_{n=0}^{\min(4k,R)}\alpha_{k-1}(n-2)\,\beta_k(n)F(x_{2k-1}, y_{n-1})\,F(x_{2ki}, y_n)$$

$$+\frac{P_i}{4}P_d\sum_{n=0}^{\min(4k,R)}\alpha_{k-1}(n-2)\,\beta_k(n).2$$

$$+\frac{P_i}{4}P_t\sum_{n=0}^{\min(4k,R)}\sum_{i=0}^{1}\alpha_{k-1}(n-3)\,\beta_k(n)\,F(x_{2k-i}, y_{n-2i})+\frac{P_i^2}{16}\alpha_{k-1}(n-4)\,\beta_k(n).$$

(10)

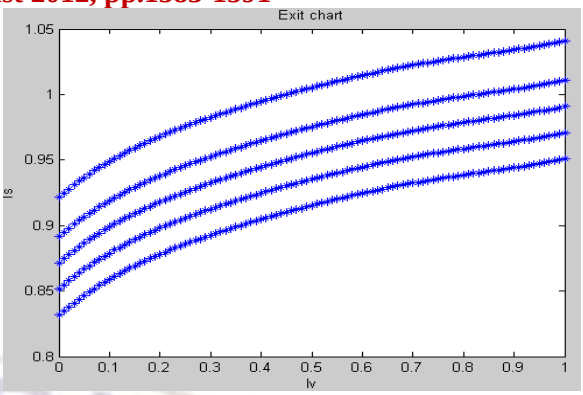


Figure 6: Achievable rate improvement through symbol level decoding for the marker "01" inserted every Nc bits.

This curve quantifies the transmission rate loss due to interleaving. Since the complexity of the algorithm grows exponentially in the group size m which makes it infeasible for large values of m, only the achievable rate for the case of 2- bit interleaving is shown. It is clear that adopting symbol-level detection recovers a significant part of the interleaving loss, particularly as the marker code rate increases. For instance, by comparing the two relevant maximum achievable rates, we can conclude that symbol-level detection is about 5% better in capacity for the given example. Although omitted from this paper, other simulation results also show similar gains for different channels, e.g., the insertion-only channel and insertion/deletion channels.

C. Exploiting Correlation via Demapper/Detector:

In this section, we consider a practical coding scheme with the aim of confirming the performance gain predicted by our information-theoretic analysis for the symbol-level detection over the bit-level detection. Specifically, we adopt a binary LDPC code of length 16383 and rate = 0.87 concatenated with a marker code with rate r_M = 30/32, obtained by inserting the marker "01" every 30 LDPC-coded bits. Hence, r = 0.8156 is obtained for the overall code rate. We compare the performance obtained by feeding the LDPC decoder with the soft information produced by the bit level detector and the symbol-level detector (with m = 2 and m = 3). In the bit-level detection case, the output of the detector directly feeds the LDPC decoder, which performs 100 self iterations and then produces the estimate of the information bits. In the symbol-level detection case, the output of the detector cannot directly feed the LDPC decoder, which is binary and cannot manage symbol-level soft information. Hence, to convert the symbol-level information to bit-level information, we adopt the soft demapper module proposed. Specifically, we use iterative soft demapping (see [27] for detailed formulations): for every 10 self iterations of the LDPC decoder, we perform one

iteration of the soft demapper, so that the total number of 100 self iterations of the LDPC decoder is preserved for a fair comparison with the bit-level case.

The resulting frame-error rate (FER) and bit-error rate (BER) curves are compared for the case of a deletion only channel. For comparison, the ultimate deletion probability Pd at which a scheme with the considered marker code and an outer code with rate = 0.87 can provide reliable communications is also shown— these values are obtained by means of the information-theoretic analysis described in the previous sections. An interesting fact is that a BER lower than is obtained by means of MAP detection with m = 3 at values of the deletion probability at which bit-level detection cannot converge even in the presence of an information-theoretically optimal code. The improvement provided by the symbol-level detection is evident: for a given BER, using a MAP detector with m = 2 allows the receiver to work with a deletion probability increased by about 10^{-3} with respect to the bit-level one, and the MAP detector with m = 3 provides an even greater robustness to deletion errors.

V. EXIT CHART-BASED OUTER LDPC CODE DESIGN FOR INSERTION/DELETION CHANNELS

In the previous sections, with the interest of reducing decoding latency, we focused on the case of single-pass decoding for the outer code concatenated with the inner marker code over insertion/deletion channels. We now consider an iterative scheme where extrinsic information is exchanged between the MAP detector (synchronization) block and the outer decoder. This is motivated by the observation that when iterative decoding is allowed, specifically designed LDPC codes for insertion and deletion channels may provide performance gains over

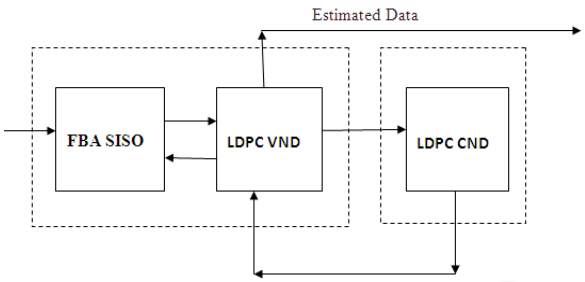


Figure. 7 Detailed decoder/detector block diagram at the receiver side.

In this section, we consider an LDPC code consisting of N variable nodes and $N - K$ check nodes connected by an edge interleaver with rate $r_c = K/N$. For simplicity, only check regular LDPC codes are considered, i.e., every parity-check equation involves a constant number of variable nodes, denoted by d_c We emphasize that joint design of variable and check nodes may offer a better performance but the check-regular LDPCs already give good results as reported in the previous literature. Suppose I is the total number of different variable node degrees of the LDPC code denoted by $d_{v,i}$ i = 1, . . . , I. Let a_i to be the fraction of variable nodes with degree $d_{v,i}$. The goal of code design is to find the set of parameters $\{\lambda i\}$ that provides the best decoding performance where

$$\sum_{i=1}^{I} \lambda_i = 1, 0 \le \lambda_i \le 1 \qquad \lambda_i = \frac{d_{v,i}}{(1-r_C)d_C} \cdot a_i$$

(11)

Because of the first constraint, we need I ≥ 3 to have any flexibility in our code design

A. EXIT Chart Based Analysis of the Decoding Performance:

Since the outer LDPC decoder can be partitioned into LDPC variable node detector (VND) and LDPC check node detector (CND), for multiple-pass decoding, the information exchange between the inner MAP detector and outer LDPC decoder is further illustrated in Figure. 7, where Block A consists of two sub-blocks which are referred to as FBA SISO and LDPC VND. Mutual information between the LDPC-coded bits and the corresponding L-values, {IA, IB, IS, } [0, 1], are exchanged between these blocks during the iterative decoding process. It is worth mentioning that only the extrinsic information, i.e., the difference between the *a posteriori* and the *a priori* L-values, is exchanged.

As stated in Section III, in the sub-block FBA SISO, MAP detection is applied on the received sequence $\{y_k\}$ with soft input *a priori* information given by IV and extrinsic L-values of the transmitted bits are generated. measures the reliability of these L-values. It is difficult to describe the relationship between and in closed form, instead, Monte Carlo simulations are performed to generate the so called detection EXIT chart. A detection EXIT chart example for insertion and deletion channels is shown in Figure. 9 using bit level synchronization and the marker code "01". Marker code rates are chosen based on the scheme proposed in Section III-C. The variable nodes take as the *a priori* information and perform the standard sum-product algorithm (SPA) with information received from the LDPC CND. The EXIT curve of the combined FBA SISO and LDPC VND is described by the relationship between IA and IB given by where the function $J(\sigma)$ is defined as

$$I_A(I_B d_V) = J\left(\sqrt{(d_V - 1)[J^{-1}(I_B)]^2 + [J^{-1}(I_S)]^2}\right),$$

(12)

$$J(\sigma) = 1 - \int_{-\infty}^{\infty} \frac{e^{-(\xi - \frac{\sigma^2}{2})^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} \cdot log_2\,[1 + e^{-\xi}]d\xi$$

(13)

In this case, can be numerically evaluated from the detection EXIT chart using a polynomial approximation with input $I_V(I_B d_V) = J(\sqrt{d_V \cdot J^{-1}(I_B)})$. For instance, when $Pi = Pd = 0.01$, we can write

$$I_S = 0.41491.I_V^5 - 1.1518.I_V^4 + 1.2405.I_V^3 - 0.71698.I_V^2 + 0.33549.I_V + 0.83146. \quad (14)$$

For a certain variable node degree distribution, the effective VND transfer curve is thus

$$I_A(I_B) = \sum_{i=1}^{I} \lambda_i . I_A(I_B, d_{v,i})$$

$$= \sum_{i-1}^{I} \lambda_i . J\left(\sqrt{(d_{v,i}-1)[J^{-1}(I_B)]^2 + [J^{-1}(I_S)]^2}\right) \quad (15)$$

At the CND, "box plus" operation is performed to generate **IA** from **IB** which can be approximately written (it is useful to express it as the inverse function).

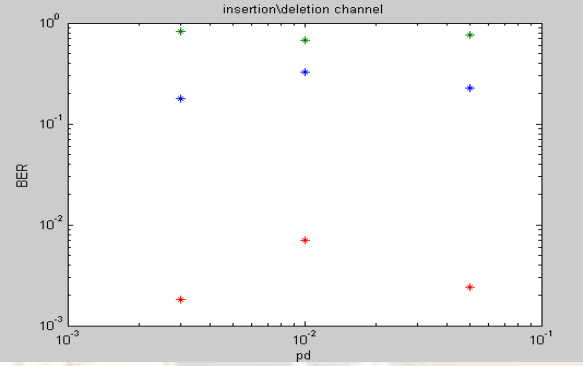


Figure . 8 Detection EXIT chart for several insertion and deletion channels for the marker "01" inserted every Nc bits.

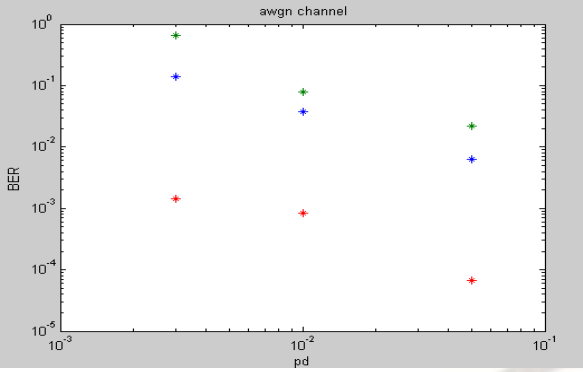


Figure. 9 Detection EXIT chart for several insertion and deletion channels for AWGN the marker "01" inserted every Nc bits

B. LDPC Code Design Example for Insertion/Deletion Channels:

Design examples are using the bit level synch**ronization** algorithm for several insertion/deletion channels, deletion only channels and insertion-only channels, respectively. We choose $I = 3$ and fix the average variable node degree $\overline{d_v}$ to be 3. Listed LDPC code degree distributions guarantee convergence with the highest code rate for different deletion/insertion rates. Therefore, the overall code rate r, product of r_M and r_C denotes the highest achievable rate when iterative decoding is performed. For deletion probabilities of 0.01 and 0.1, the overall rates are obtained as 0.860 and 0.486, respectively, where the capacity lower bound is 0.919 for $P_d = 0.01$ and 0.531 for $P_d = 0.1$. The corresponding gaps are 0.059 and 0.045 bits/channel use for the two cases, which are clearly smaller than the one demonstrated in Section III-C. We also expect that the gap to the capacity bound can be further narrowed by allowing $I > 3$ and not fixing $\overline{d_v}$ *to be 3.*

The length of the LDPC codeword is set to be $N = 5000$ and the selected marker code rate is determined to maximize the transmission rate. we calculate the ratio of the BERs of the two codes (optimized one versus the AWGN-only code) when the codes optimized for insertion and deletion channels attain a BER. The higher the ratio, the greater the improvement. Clearly, all of the codes outperform the ones designed for AWGN channels. However, the gap becomes less obvious as the insertion/deletion rate decreases. This is not a surprising result because for low insertion/deletion probabilities, the detection EXIT charts tend to be flat as illustrated. This is similar to the one for a memory less AWGN channel[28]. In this case, specific design of an LDPC code for insertion/deletion channel may not be required since the gain is negligible. Similar conclusions are drawn for ISI channels with short channel impulse responses. Also, when the symbol-level detection is performed, the left-most point in the detection EXIT chart is much better than the bit-level case as explained in Section IV-B. The rightmost point in the detection EXIT chart is identical for both cases since MAP detector achieves ideal synchronization in this case. Therefore, the detection EXIT chart for symbol level detection is flatter than the one for the bit-level case. This observation suggests that for channels with low insertion/deletion rates, it is more likely that symbol-level detection itself already yields a good performance and iterative decoding and LDPC code design may not be needed, which is also an obvious fact, since when $m = T$, optimal detection (i.e., for synchronization purposes) is achieved and there is no gain with iterative decoding/demapping. Clearly, this is not feasible in practice since the detection complexity in m is exponential and T is typically large.

VI. CONCLUSION

We have studied performance of an outer LDPC code concatenated with an inner marker code for data transmission over insertion/deletion channels.

Two decoding strategies are considered: single-pass decoding and multi-pass decoding with information exchange between the inner detector and the outer decoder. For the first case, through numerical mutual information analyses, we have developed a technique that allows us to optimize the marker code based on the ultimate rate achievable by the concatenated scheme. Moreover, we have presented a new symbol-level detection algorithm, which has been proved to outperform the standard bit-level one in terms of achievable rates. An iterative detector/demapper is also designed which is able to exploit the results of the symbol level synchronizer. Finally, when iterative decoding is allowed, we have shown that by choosing good variable and check node degree distributions, LDPC codes designed for insertion/deletion channels offer better error correcting capabilities than those optimal for the AWGN-only channels. Simulation results related to practical LDPC codes showing clear performance gains have been provided for both cases under consideration. Although we only focus on the marker codes (as the inner synchronization code), similar analyses and design procedure can also be applied to other concatenated coding schemes, e.g., an LDPC code concatenated with an inner watermark code.

References

[1]. Feng Wang, Dario Fertonani, and Tolga M. Duman" Symbol-Level Synchronization and LDPC Code Design for Insertion/Deletion Channels " IEEE Trans On Comm , Vol. 59, No. 5, May 2011.

[2] R. L. White, R. M. H. New, and R. F. W. Pease, "Patterned media: a viable route to 50 Gbit/in2 and up for magnetic recording?."

[3] R. L. Dobrushin, "Shannon's theorems for channels with synchronization Errors."

[4] M. Mitzenmacher, "A survey of results for deletion channels and related synchronization channels."

[5] D. Fertonani, T. M. Duman, and M. F. Erden, "Bounds on the capacity of channels with insertions, deletions and substitutions."

[6] F. F. Sellers Jr., "Bit loss and gain correction code."

[7] N. J. A. Sloane, "On single-deletion-correcting codes."

[8] Z. Liu and M. Mitzenmacher, "Codes for deletions and insertion channels with segmented errors."

[9] A. S. Helberg and H. C. Ferreira, "On multiple insertion/deletion correcting codes."

[10] L. McAven and R. Safavi-Naini, "Classification of the deletion correcting capabilities of Reed-Solomon codes of dimension 2 over prime fields."

[11] S. E. Tavares and M. Fukada, "Matrix approach to synchronization recovery for binary cyclic codes."

[12] G. Seguin, "On synchronizable binary cyclic codes."

[13] L. Cheng and H. Ferreira, "Rate-compatible path-pruned convolutional codes and their applications on channels with insertion, deletion and substitution errors."

[14] T. G. Swart and H. C. Ferreira, "Insertion/deletion correcting coding schemes based on convolution coding."

[15] M. F. Mansour and A. H. Tewfik, "Convolutional codes for channels with substitutions, insertions, and deletions."

[16] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals."

[17] H. C. Ferreira, T. G. Swart, and M. P. dos Santos, "Using parallelinterconnected Viterbi decoders to correct insertion/deletion errors."

[18] L. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions."

[19] M. C. Davey and D. J. Mackay, "Reliable communication over channels with insertions, deletions and substitutions."

[20] E. A. Ratzer, "Error-correction on non-standard communication channels."

[21] "Marker codes for channels with insertions and deletions." Annals of Telecommun.

[22] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition."

[23] J. Chen, M. Mitzenmacher, C. Ng, and N. Varnica, "Concatenated codes for deletion channels."

[24] R. G. Gallager, "Sequential decoding for binary channels with noise and synchronization errors."

[25] T. M. Cover and J. A. Thomas, Elements of Information Theory. John Wiley & Sons, Inc., 1991.

[26] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection."

[27] M. Franceschini, G. Ferrari, and R. Raheli, "Does the performance of LDPC codes depend on the channel?."

[28] N. Varnica and A. Kav´ci´c, "Optimized low-density parity-check codes for partial response channels."