Spatially Coupled Low-Density Parity-Check Codes Over Fading Channels

submitted in partial fulfillment of the requirements of the degree of

Bachelor of Technology

by

Adway Girish

Roll no. 180070002

Supervisor: Prof. Kumar Appaiah

Department of Electrical Engineering

Indian Institute of Technology Bombay

July – November 2021

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date: 26^{th} November, 2021

Student name: Adway Girish

Roll no.: 180070002

Spatially Coupled Low-Density Parity-Check Codes Over Fading Channels

Adway Girish

Abstract

Low-density parity-check (LDPC) codes have achieved great fame as capacity-approaching codes. A surprising result has been the marked improvement in decoding thresholds and error performance obtained on 'coupling' blocks of LDPC codes into a single chain in a convolutional manner, called spatially coupled LDPC (SC-LDPC) codes. We look at an application of these codes over practical fading channels, with interleaving to maximise performance. Since the promised improvement requires large blocklengths, causing large decoding latency, we also study the performance of an efficient, low-complexity windowed decoder.

Contents

1 Motivation **1**

2 Background and Prior Work **3**

2.1 From Codes to LDPC Codes 3

2.2 Decoding LDPC Codes : Belief-Propagation 4

2.2.1 Binary Erasure Channel 5

2.2.2 Binary Additive White Gaussian Noise Channel 5

2.3 From LDPC Codes to SC-LDPC Codes 7

2.3.1 Protographs 7

2.3.2 Spatial Coupling 9

2.3.3 Performance of LDPC Codes vs. SC-LDPC Codes 10

2.4 Fading Channels and Diversity 11

2.4.1 The Fading Channel Model 11

2.4.2 Fast and Slow Fading 12

2.4.3 Diversity and Interleaving 13

2.4.4 Existing Work on SC-LDPC Codes Over Fading Channels 15

2.5 Windowed Decoding 15

2.5.1 Existing Work 15

3 Our Results **17**

3.1 Over Fading Channels With Interleaving 17

3.1.1 The Trade-off Involved 18
3.1.2 Problem Setting 18
3.1.3 Results 18
3.1.4 Comments 23
3.2 Low-Latency Windowed Decoding 23
3.2.1 Results 23
3.2.2 Comments 23

4 Conclusion **25**

List of Figures

2.1 Tanner graph for the (7,4,3) Hamming code 4
2.2 Channel models for the BEC and BAWGNC 5
2.3 Protograph representations of LDPC and terminated SC-LDPC codes 8
2.4 BP thresholds over the BEC of (J, K)-regular and associated $\mathcal{C}(J, K, L)$ ensembles 11
2.5 How interleaving helps 14
2.6 Windowed Decoding of a $\mathcal{C}(J, K, L)$ code 16

3.1 BER vs. SNR for $f_\mathrm{d} = 10^{-3}$ 19
3.2 BER vs. SNR for $f_\mathrm{d} = 10^{-4}$ 20
3.3 BER vs. SNR for $f_\mathrm{d} = 10^{-5}$ 21
3.4 BER vs. SNR for the IID channel 22
3.5 Windowed decoding results 24

List of Tables

3.1 Diversity with and without interleaving for $f_{\mathrm{d}} = 10^{-3}$ 19
3.2 Diversity with and without interleaving for $f_{\mathrm{d}} = 10^{-4}$ 20
3.3 Diversity with and without interleaving for $f_{\mathrm{d}} = 10^{-5}$ 21
3.4 Diversity with and without interleaving for the IID channel 22

List of Algorithms

1 BP for the BEC 6
2 BP for the BAWGNC 7

Chapter 1

Motivation

The need for faster, more efficient and more reliable communication paradigms in the present digital age is obvious. Ever since Shannon's landmark 1948 result [1] which guarantees that with proper encoding, it is possible to transmit across noisy channels with arbitrarily low errors as long as the rate of transmission is below the channel capacity, we have been aware of several such 'codes', termed as *error-correcting codes.* The earliest of these (such as Hamming, Golay, Bose–Chaudhuri–Hocquenghem, Reed-Solomon codes) had algebraic or topological structures to allow for efficient decoding, but Shannon's results were obtained as averages of random ensembles and these codes fell short of the promised performance at finite blocklengths.

It remained this way until the the early 1990s, when *turbo codes* [2] were discovered – random-like but with enough structure to be recovered by low-complexity *iterative decoding.* This also led to the re-discovery of *low-density parity-check* (LDPC) *codes* [3], originally proposed by Gallager in the 1960s, but discarded due to limitations in the computational power available at the time. LDPC codes have since been shown to achieve performances extremely close to the Shannon limit [4], and have lower error floors than turbo codes. These are two instances of more general *sparse graph codes.*

The *decoding threshold* of a code is the maximum noise at which we can still obtain arbitrarily low error rates. Maximum *a posteriori* (MAP) decoding is the optimal decoding strategy – no other decoder can do better, i.e. the MAP threshold is the highest – but it has a high complexity. On the other hand, the belief-propagation (BP) decoding algorithm for sparse graph codes has a low complexity, at the cost of lower BP thresholds in general. This is where 'spatial coupling' comes in – the method of 'unwrapping' a cyclic block code into a convolutional structure. It has been proven that for spatially coupled LDPC (SC-LDPC) codes (also called convolutional LDPC codes), the BP threshold is numerically indistinguishable from the MAP threshold of the underlying ensembles [5, 6] – we obtain the performance of a high-complexity decoder using a low-complexity decoder.

To see the effect of this improvement in action, we look at the performance of SC-LDPC codes over correlated *fading channels*, where the channel strength varies with time and frequency. These channels are known to be favourable to convolutional codes [7], so it is expected that convolutional LDPC codes do well over them too [8]. The large blocklengths needed to realise the performance improvement induce large decoding delays, which are not acceptable in a practical implementation. This is dealt with by using a *windowed decoder* [9, 10], which performs BP decoding on small sections of the code as they are received instead of waiting for the entire message. Studying the performance of such a decoder is important since it makes the remarkable theoretical result above practically useful.

Chapter 2

Background and Prior Work

In this chapter, we review the existing literature and provide the preliminaries that form the basis for the study of SC-LDPC codes that follow.

2.1 From Codes to LDPC Codes

We start by quickly looking at some basic concepts and setting up notation, provided for completeness. A *code* $\mathcal{C}$ of length n (called the blocklength) and cardinality C over a field $\mathbb{F}$ is a set of C elements from $\mathbb{F}^n$. The *rate* of a code is the number of information symbols per transmitted symbol, given by $R = \frac{1}{n}\log_{|\mathbb{F}|} C$. Here we only look at *binary codes*, where $\mathbb{F}$ is taken to be the Galois Field $\mathbb{F}_2$ (elements are 0 and 1; addition and multiplication are modulo 2).

Linear codes are those which are closed under addition and scalar multiplication, i.e. it forms a subspace of $\mathbb{F}^n$. Let the dimension of this subspace be k ($0 \leq k \leq n$). Then $\mathcal{C}$ has $|\mathbb{F}|^k$ codewords and a linearly independent basis containing k vectors of length n. These vectors form a $k \times n$ matrix $\mathbf{G}$ called the *generator matrix* of $\mathcal{C}$. We also have an equivalent representation of a linear code $\mathcal{C}$ in terms of a *parity-check matrix* $\mathbf{H}$, with the property that for every codeword $\mathbf{x} \in C$, $\mathbf{H}\mathbf{x}^\intercal = \mathbf{0}$. Since $\mathbf{G}$ has a dimension of k, $\mathbf{H}$ has a dimension of $m = n - k$ by the rank-nullity theorem.

The parity-check matrix $\mathbf{H}$ is an $m \times n$ matrix. To this we can associate a bipartite graph with m *check nodes* and n *variable nodes*, where there is an edge from the i^{th} check node to the j^{th} variable node if the element at the i^{th} row and j^{th} column (denoted as H_{ij}) is 1 – this is called a *Tanner graph* representation of the code $\mathcal{C}$. Since the parity-check matrix is not unique for a code, neither is the Tanner graph. As an example, consider the parity-check matrix $\mathbf{H}$ of a (7,4,3)-Hamming code given in Eqn. 2.1 and its associated Tanner graph given in Fig. 2.1.

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{2.1}$$

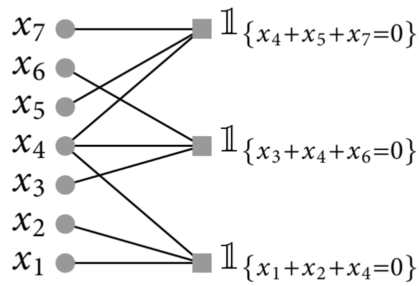


Figure 2.1: Tanner graph for the parity-check matrix $\mathbf{H}$ given in Eqn. 2.1 [11]

Linear codes that have at least one sparse Tanner graph are called *low-density parity-check* (LDPC) codes. LDPC codes where every check node has the same degree, say K, and every variable node has the same degree, say J, are called (J, K)*-regular* LDPC codes. The parity-check matrix $\mathbf{H}$ of such a code has K ones in each row and J ones in each column. These codes have a rate of $R \geq 1 - \frac{J}{K}$, with equality only if $\mathbf{H}$ has full rank.

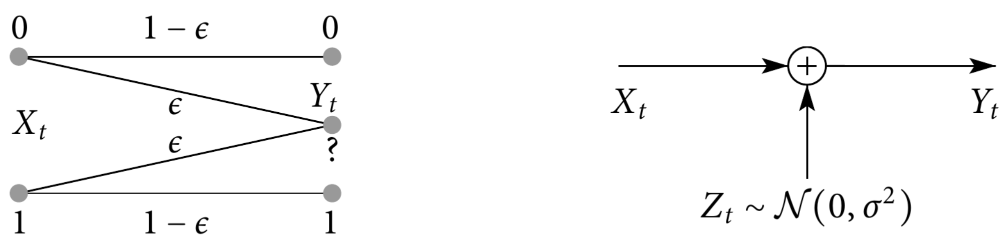
LDPC codes where the degrees of the check and variable nodes have some fixed distribution are called *irregular* LDPC codes.

2.2 Decoding LDPC Codes : Belief-Propagation

We now look at how to decode a corrupted message received as the output of an LDPC codeword through some noisy channel. The bipartite graph representation of the code comes in handy here. We separate variable and check nodes to different sides, then 'pass messages' iteratively from one to the other. There are combining rules at each node to evaluate the message that it sends out in the next iteration. At each step, our knowledge of the codeword gets stronger till we can do no better (either the codeword is completely decoded or we are stuck in some *stopping set*). This is called the *message-passing* decoder or the *belief-propagation* (BP) algorithm. In particular, we look at the BP algorithms for LDPC codes over the Binary Erasure Channel (BEC) and the Binary Additive White Gaussian Channel (BAWGNC) respectively. Since both these channels are symmetric with respect to a binary input, we assume that the codeword sent is the all-zero codeword without loss of generality.

2.2.1 Binary Erasure Channel

The BEC is characterised by its *erasure probability* ϵ. It takes a binary input (0 or 1), and deletes it to form an erasure ε with probability ϵ. This is represented in Fig. 2.2a. The BP algorithm for the Binary Erasure Channel (BEC) is given in Alg. 1. The output obtained is the decoded codeword of length n. The codeword sent is assumed to be $\mathbf{x} = (0, \ldots, 0)$ of length n (changes for a general $\mathbf{x}$ are given as comments).



(a) BEC model

(b) BAWGNC model

Figure 2.2: Channel models – X_t denotes the input, Y_t denotes the output [11]

2.2.2 Binary Additive White Gaussian Noise Channel

We consider the BAWGNC with Binary Phase Shift Keying (BPSK) modulation, i.e. $0 \mapsto +1$ and $1 \mapsto -1$. Then the transmitted codeword becomes $\mathbf{x} = (+1, \ldots, +1)$ of length n. The

Algorithm 1 BP for the Binary Erasure Channnel (BEC)

Input: received noisy codeword $\mathbf{y} \in \{0, \varepsilon\}^n$, and the Tanner graph $\mathcal{C}$ corresponding to the LDPC code used ▷ $\mathbf{y} \in \{0, 1, \varepsilon\}^n$

Output: variable node outgoing messages after all iterations

outgoing message from variable node $\leftarrow$ bit value $\in \{0, \varepsilon\}$ ▷ $\in \{0, 1, \varepsilon\}$
while at least one bit was corrected in the previous iteration **do**
    for each check node **do**
        if any incoming message is ε **then**
            outgoing message = ε
        else
            outgoing message = 0 ▷ modulo-2 sum of incoming messages
        end if
    end for
    for each variable node **do**
        if all incoming messages are ε **then**
            outgoing message = ε
        else
            outgoing message = 0 ▷ 0 or 1, all incoming non-erasure messages coincide
        end if
    end for
end while

received message is given by $\mathbf{y} = \mathbf{x} + \mathbf{w}$ where $\mathbf{w} \sim \mathcal{N}(0, \sigma^2)$, i.e. $\mathbf{w}$ is a Gaussian noise vector of variance σ^2 as shown in Fig. 2.2b. We define the *signal-to-noise ratio* (SNR) as the ratio of the energy of the signal to that of the noise, σ^2, i.e. SNR $= \frac{1}{\sigma^2}$.

Since the output vector takes continuous values, we use soft-decision decoding to obtain the best performance. For this we require a quantity called the *log-likelihood ratio* (LLR) for each bit $i \in \{1, \ldots, n\}$, given the transmitted codeword $\mathbf{x} = (x_1, \ldots, x_n)$ and the received message $\mathbf{y} = (y_1, \ldots, y_n)$, denoted by

$$l_i = \log \frac{\Pr\{x_i = +1 \mid y_i\}}{\Pr\{x_i = -1 \mid y_i\}}; \qquad \text{for BAWGNC, } l_i = \frac{2y_i}{\sigma^2} \tag{2.2}$$

The LLR vector $L = (l_1, \ldots, l_n)$ represents the strength of our 'belief' of the bit value. It is a *sufficient statistic* for $\mathbf{y}$ with respect to decoding. The algorithm for BP decoding over BAWGNC using LLRs is given in Alg. 2. The calculation of the outgoing messages from check nodes can be greatly simplified with a log-max approximation as in Eqn. 2.3 (we use the fact that for $f(x) = \log \tanh |x|$, $f(x) = f^{-1}(x)$ and $|f|$ is decreasing for $x > 0$).

$$\begin{aligned} l &= 2\tanh^{-1}\left(\prod_j \tanh\left(\frac{l_j}{2}\right)\right) \\ \implies \left|\log\tanh\left(\frac{|l|}{2}\right)\right| &= \sum_j \left|\log\tanh\left(\frac{|l_j|}{2}\right)\right| \approx \max_j \left|\log\tanh\left(\frac{|l_j|}{2}\right)\right| \\ \implies |l| &\approx \min_j |l_j| \end{aligned} \tag{2.3}$$

The sign of l is determined by checking the number of positive and negative l_j.

Algorithm 2 BP for the Binary Additive White Gaussian Noise Channel (BAWGNC)

Input: received noisy codeword $\mathbf{y} \in \mathbb{R}^n$, and the Tanner graph $\mathcal{C}$ corresponding to the LDPC code used
Output: variable node outgoing messages after all iterations

outgoing message from variable node $i \leftarrow$ LLR value $l_i \in \mathbb{R}$ at all i
for a maximum number of iterations **do**
  for each check node c **do**
    outgoing message to variable node v:
      $l = 2\tanh^{-1}\left(\prod_j \tanh\left(\frac{l_j}{2}\right)\right)$, j covers all incoming edges to c except from v
  end for
  for each variable node v **do**
    outgoing message to check node c:
      $l = \sum_j l_j$, j covers all incoming edges to v except from c
  end for
end for

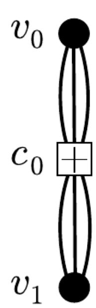
2.3 From LDPC Codes to SC-LDPC Codes

Before we move on to the construction of spatially coupled LDPC (SC-LDPC) codes, we study at a protograph-based code construction [12].

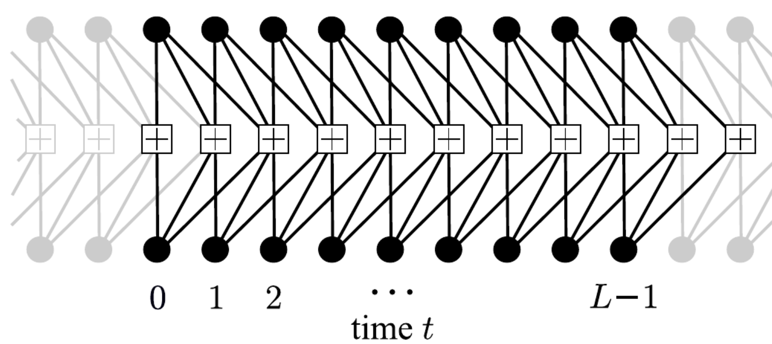
2.3.1 Protographs

We can simplify the construction of large codes using smaller representations containing their essence. Consider a small bipartite graph with b_v variable nodes and b_c ($< b_v$) check nodes, and its associated parity-check matrix $\mathbf{B}$. The graph is called the *protograph* and the matrix $\mathbf{B}$ is called the *base matrix* of the parity-check matrix (and hence code) that will be constructed. The protograph can be expanded to a code of blocklength $n = Mb_v$ by a process called *M-lifting*, where each edge e in the protograph (connected to variable node v and check node c) is replaced by a collection of M edges connecting M copies of v to M copies of c; these connections are then randomly permuted among themselves. Concretely, to obtain the M-lifted parity-check matrix $\mathbf{H}$ from the protograph $\mathbf{B}$, replace each non-zero element $\mathbf{B}_{ij}$ in $\mathbf{B}$ by the sum of $\mathbf{B}_{ij}$ $M \times M$ permutation matrices and each zero element by an all zero $M \times M$ matrix. This new parity-check matrix is an $Mb_c \times Mb_v$ matrix and the code represented by it has the same rate as that of the protograph. The collection of all codes represented by a protograph is called an *ensemble*. The *design rate* of this ensemble is given by $R = 1 - \frac{b_c}{b_v}$ (this actually gives a lower bound to the rates of the codes in the ensemble – when some check nodes are redundant we get higher rates – this difference is negligible and we henceforth refer to them interchangeably).

For example, consider a $(3, 6)$-regular LDPC code of any arbitrary blocklength. This can be concisely represented by the protograph shown in Fig. 2.3a (each check node has six edges going to variable nodes, and each variable node has three edges going to check nodes – hence the check node degree is 6, and the variable node degree is 3).



(a) A (3,6)-regular protograph



(b) Terminated protograph for the $\mathcal{C}(3, 6, L)$ ensemble with coupling width w

Figure 2.3: Protograph representations of LDPC and terminated SC-LDPC codes [12]

2.3.2 Spatial Coupling

In short, *spatial coupling* refers to linking a sequence of individual block protographs into a chain by spreading some edges to neighbouring protographs to obtain a convolutional protograph. The number of block protographs coupled together can be finite or infinite (one-sided or two-sided); we restrict our attention to the former.

Suppose we have L (> 0) block protographs to couple together (L is then called the *coupling length*). Place them at positions indexed by $t = 0, \dots, L-1$. Consider the protograph at position t. Take the $\mathbf{B}_{ij}$ edges connecting variable node v_j to check node c_i and spread them over to check nodes c_i at the $w+1$ positions given by t, $t+1$, $\dots$, $t+w$ (keeping them fixed to the v_j at position t), where w (> 0) is called the *coupling width* of the code. This is illustrated in Fig. 2.3b. The convolutional protograph thus generated has $(L+w)b_c$ check nodes and Lb_v variable nodes. Note that the check nodes at the ends of this protograph have a lower degree. This leads to a rate loss, i.e. the design rate of this protograph is $R_L = 1 - \frac{(L+w)b_c}{Lb_v} = 1 - \frac{(L+w)}{L}(1-R) < R$, where R is the design rate of the underlying block protograph. The strength of these codes lies in the depth of the coupling, represented by the *constraint length*, $\nu = M(w+1)b_v$. Such a construction gives what is called a *terminated SC-LDPC* ensemble, as shown in Fig. 2.3b. An alternative method is to spread the edges modulo L – this gives us *tail-biting ensembles* with a protograph having Lb_c check nodes and Lb_v variable nodes (the rate and degree distribution remain the same as the underlying protograph; their properties are closer to unterminated convolutional codes). Especially useful to us will be the terminated ensemble of coupling length L where the underlying protograph is that of a (J, K)-regular ensemble. This will be referred to as the $\mathcal{C}(J, K, L)$ ensemble.

The spreading can be represented by $(w+1)$ number of $b_c \times b_v$ *component base matrices* $\mathbf{B}_i(t)$ at each position index t satisfying $\sum_{i=0}^{w} \mathbf{B}_i(t) = \mathbf{B}$ for all t. $\mathbf{B}_i(t)$ then represents the mini protograph formed between the variable nodes at position t and the check nodes at position $t+i$. We will look only at time-invariant edge-spreading where $\mathbf{B}_i(t) = \mathbf{B}_i$ for all t. We can now represent our newly generated terminated convolutional protograph in terms of these component base matrices as the $(L+w)b_c \times Lb_v$ matrix given by

$$\mathbf{B}_{[0,L-1]} = \begin{bmatrix} \mathbf{B}_0 & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & \\ \vdots & \mathbf{B}_1 & \ddots & \\ \mathbf{B}_w & \vdots & \ddots & \mathbf{B}_0 \\ & \mathbf{B}_w & & \mathbf{B}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{B}_w \end{bmatrix}. \tag{2.4}$$

2.3.3 Performance of LDPC Codes vs. SC-LDPC Codes

The *capacity* of a channel is the maximum possible rate of transmission over the channel using any code such that the error probability can be made arbitrarily low by increasing the blocklength. Similarly, the *threshold* of a code ensemble over a general channel is the maximum channel 'noise' such that arbitrarily low error rates can be obtained. For a BEC the threshold is the maximum erasure probability and for the BAWGNC it is the minimum SNR (or maximum σ). For any code, the optimal decoder is the maximum *a posteriori* (MAP) decoder, which maximises the probability of the chosen codeword having been the original, conditioned on the probability that we have observed the received codeword. The general MAP decoding problem has no efficient algorithms (that are of polynomial order in the blocklength) – this is a high-complexity task. Belief-Propagation is a low-complexity algorithm – unfortunately, it does not achieve MAP performance over general LDPC codes. But remarkably, it has been shown (first for the BEC [5] and later for general binary memoryless symmetric channels [6]) that BP decoding on SC-LDPC ensembles (in the limit where L goes to infinity) gives us thresholds that are numerically identical to the MAP thresholds of the underlying ensembles, which are strictly greater than their BP thresholds.

For example, consider the (3,6)-regular LDPC ensemble. This is a code of rate $R = 1 - \frac{3}{6} = 0.5$, so had it been capacity-achieving, the MAP threshold would have been 0.5, since the capacity of a BEC is given by $C = 1 - \epsilon$. It can be calculated that the MAP threhold is $\epsilon_{\text{MAP}}^{\text{LDPC}} = 0.4881$, and the BP threshold is $\epsilon_{\text{BP}}^{\text{LDPC}} = 0.429$. However, the limit of the BP threshold of the $\mathcal{C}(3, 6, L)$ as $L \to \infty$ is calculated to be $\epsilon_{\text{BP}}^{\text{SC-LDPC}} = 0.4881 = \epsilon_{\text{MAP}}^{\text{LDPC}}$. Some more such examples are shown in Fig. 2.4, along with their evolution as a function of L.

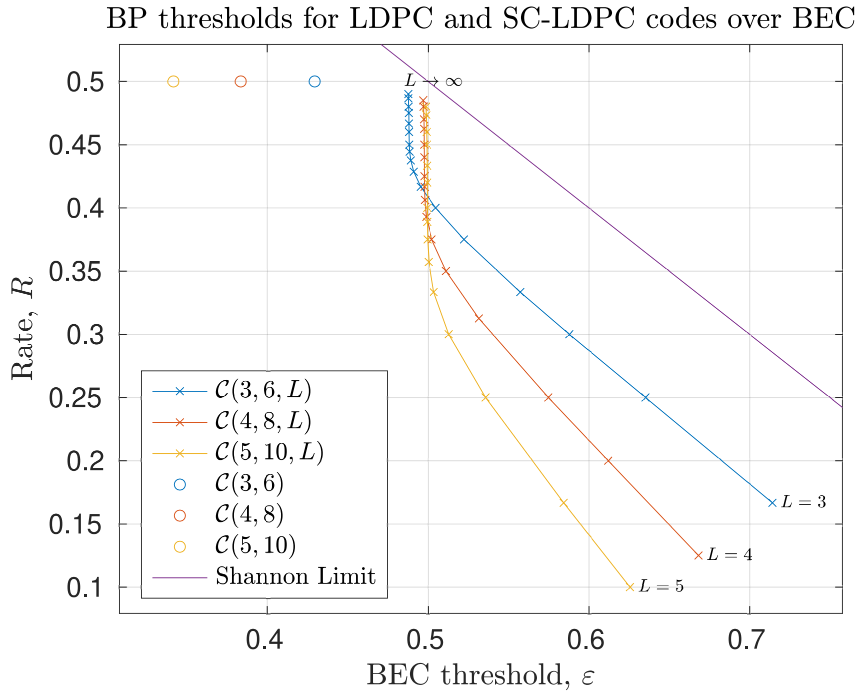


Figure 2.4: BP thresholds over the BEC of (J, K)-regular and associated $\mathcal{C}(J, K, L)$ ensembles for different values of L. Observe that as $L \to \infty$, the thresholds saturate at levels *greater* than the (J, K)-regular thresholds. The Shannon limit $R = 1 - \epsilon$ is shown for reference. (Plot generated by own program to find BP thresholds using density evolution algorithm in [12, Sec. III-B])

2.4 Fading Channels and Diversity

Having completed the required background on SC-LDPC codes, we digress to study the channel over which we look at an application of these codes – the fading channel.

2.4.1 The Fading Channel Model

Wireless channels have variations in the channel strength over both time and frequency. This is called *fading*. A channel which models such variations is called a *fading channel*. They also introduce zero-mean additive white Gaussian noise of variance σ^2. We look at the discrete-time complex baseband model. A detailed explanation of the physical mechanisms that lead to the mathematical models can be found in *Fundamentals of Wireless Communication* [13]. The channel itself can be modelled as a time-varying filter with the l^{th} filter tap given by $h_l[m]$ (a random variable) at time index m, and the effect of the channel on the input $\mathbf{x} = (\ldots, x[-1], x[0], x[1], \ldots)$ as

$$y[m] = \sum_l h_l[m]x[m-l] + w[m], \tag{2.5}$$

where $w[m]$ is a discrete-time complex zero-mean white Gaussian process. In particular, we will only study single-tap filter models for the fading channel, given by

$$y[m] = h[m]x[m] + w[m]. \tag{2.6}$$

The filter taps can be modelled in several ways depending on the physical mechanism of fading that most affects the channel strength. In the case of wireless communication channels where there are several scatterers in the path and no dominant line-of-sight component, we assume that $h_l[m]$ has both real and imaginary components distributed independently and identically with zero mean and variance $\frac{\sigma_l^2}{2}$, represented as $h_l[m] \sim \mathcal{CN}(0, \sigma_l^2)$ (circular symmetric complex Gaussian). This model is called *Rayleigh fading* because the effect of this model is that the magnitude of the input gets scaled by $|h_l[m]|$, which is Rayleigh distributed with probability distribution function (PDF) given by

$$f_R(r) = \frac{r}{\sigma_l^2} \exp\left(-\frac{r^2}{2\sigma_l^2}\right) \text{ for } r \geq 0. \tag{2.7}$$

2.4.2 Fast and Slow Fading

Note that the filter taps need not be independent across time – in fact, since the rate of channel variation is finite (arising as a Doppler shift from a moving antenna, for example), we have that realistic fading channels are necessarily temporally correlated. Such a fading model is called a *continuously-varying* fading channel, as opposed to *block* fading channels where the channel parameter takes an independent (random) value for each block of indices. The speed of variation of the channel (number of time indices after which the channel becomes uncorrelated with the initial value) is represented by the *normalised Doppler frequency*, denoted by f_d or $F_\mathrm{d}T_\mathrm{s}$ (F_d is the actual Doppler frequency and T_s is the symbol period for normalisation). The physical parameters this depends on is unnecessary for our purposes; it suffices to know that $f_\mathrm{d} = 10^{-3}$ typically represents a *fast fading* process, and $f_\mathrm{d} = 10^{-5}$ typically represents a *slow fading* process.

Since the channel parameters are random, for any rate of transmission, there is a non-zero probability that $|h_l[m]|$ is very small and leads to errors. Such a situation is called a *deep fade*. In fast fading channels, since we go through many independent fades over the duration of a single codeword, it is always likely that at least some of them will not be in deep fade and the codeword will be recoverable. But in slow fading channels, if the channel happens to be poor, it will be so over the duration of the entire codeword, making the message undecodable. Hence fast fading channels allow for better error performance than slow fading channels.

2.4.3 Diversity and Interleaving

To ensure that the message is recoverable with high probability, we send copies of the same signal over multiple (independently faded) paths in space, time, and frequency – giving us *diversity*. A simple example is to use a repeat each bit a fixed number of times – repetition coding – but we can do better by exploiting the degrees of freedom of the channel.

Quantitatively, we define the *diversity gain* or simply diversity d of a codeword transmitted over a channel with some SNR as follows. It can be shown that the plot of the error probability (bit error rate (BER)) with the SNR becomes nearly linear at high SNR values. The slope of this line is taken to be the diversity (for a channel with no diversity, P_e is observed to be $\approx \frac{1}{\mathrm{SNR}}$, i.e. $d = 1$).

$$d = - \lim_{\mathrm{SNR} \to \infty} \frac{\log P_e}{\log \mathrm{SNR}} \tag{2.8}$$

Increasing the diversity can be done by a combination of convolutional codes with *interleaving*, which is the process of re-arranging the bits so that those that are close to each other originally are now spaced apart. This is done by writing the bits into a matrix row-wise and reading from it column-wise. Let the codelength be n and let the *interleaving depth* be d, such that $n = kd$ (all positive integers). We write the input codeword $\mathbf{x}$ row-wise into a $d \times k$ matrix A and read it column-wise to obtain the interleaved codeword $\tilde{\mathbf{x}}$. Thus we have convolutional codes which spread a bit over adjacent bit positions, followed by interleaving to disperse these copies over far-off, independent fades.

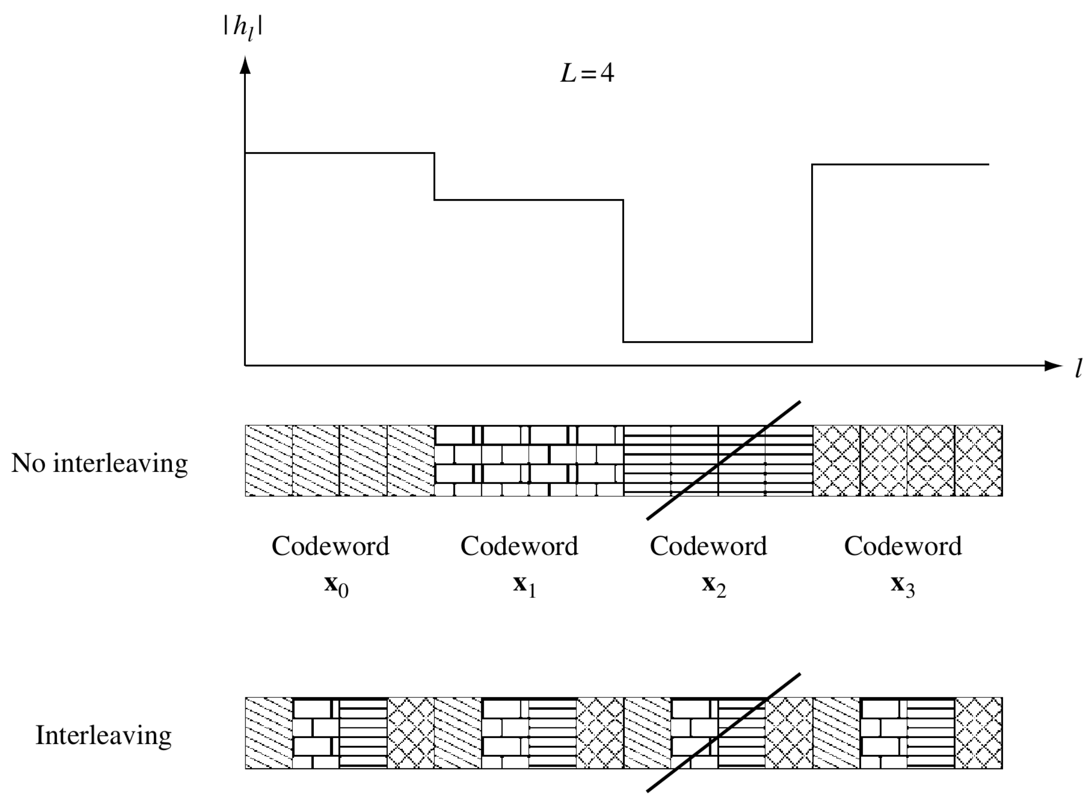


Figure 2.5: How interleaving helps ensure that at least some copies do not pass through deep fades, by passing multiple copied through independent fades [13]

2.4.4 Existing Work on SC-LDPC Codes Over Fading Channels

There has been some investigation into the performance of SC-LDPC codes over fading channels. Najeeb ul Hassan et al. [8] show that we can achieve remarkable diversity improvement by using SC-LDPC codes over block faded channels, but this is not a realistic representation of a practical fading channel. Further, there is no mention of interleaving or the latency introduced by the large blocklengths. There are a few works [14, 15] which look at interleaved SC-LDPC codes over continuously-varying fading channels, but they consider only uncorrelated, fast fading channels. We look to extend the study of interleaved SC-LDPC codes to correlated fading channels, and also study the effect of the speed of channel variation on the performance (Sec. 3.1).

2.5 Windowed Decoding

Recall from the section on BP decoding that the algorithm requires the knowledge of all bits in the received codeword before it can start decoding. This leads to a large latency when used in practical applications. A simple, low-complexity way to reduce this latency is by using a *windowed decoder* (WD). The idea behind this is straightforward – instead of performing BP decoding on the entire parity-check matrix, fix a *window length* W and perform the same BP algorithm over subcodes containing only W of the underlying protographs (*sections*), as opposed to all $L + w$. Start from the first W sections, then slide over by one when no more decoding can be done in the first section (completely decoded or no more improvement possible), and repeat until the last section is reached. Fig. 2.6 shows the region being looked at by the windowed decoder at an intermediate instant. The procedure above allows us to achieve a latency that is a $\frac{W}{L}$ fraction of the usual BP decoder.

2.5.1 Existing Work

It has been shown that the performance of the WD very quickly approaches that of the full-length BP decoder over binary erasure channels [9, 10] (exponentially fast in W), but it is sub-optimal in general. Attempts at extending this analysis to fading channels have been restricted to block fading channels [16], and to IID fading channels [15] where the WD is used to provide latency-efficient implementations, but there is no analysis of the variation in performance with the window length W. We will study the WD performance over continuously-varying, correlated fading channels, for different values of W and see how the performance scales qualitatively (Sec. 3.2).

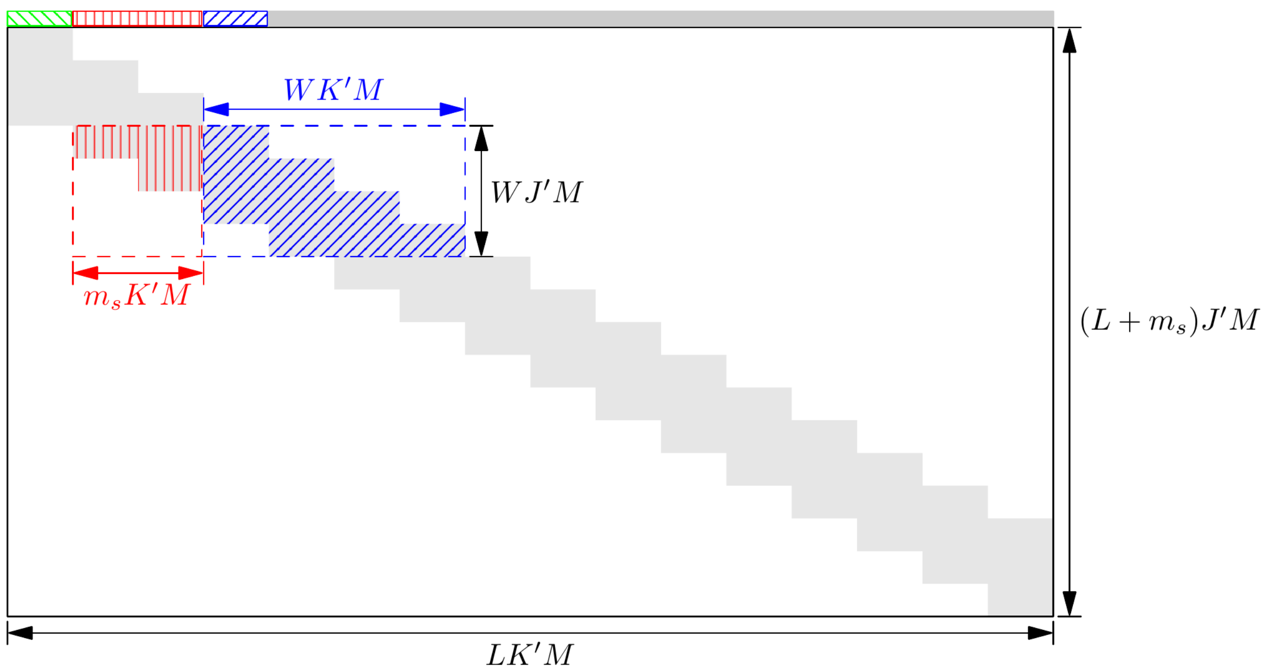


Figure 2.6: The windowed decoder in action over an M-lifted $\mathcal{C}(J,K,L)$ code. The green represents the processed bits – they will not be considered in further decoding. m_s here is the coupling width w. (J',K') is (J,K) divided by their greatest common divisor (GCD). The area in blue is the subcode on which BP is currently being performed. The red area is the bits that are still involved in the region through equations. [9]

Chapter 3

Our Results

We first look at an application of block-interleaved SC-LDPC codes over continuously-varying fading channels, subjected to a latency constraint. This shows us how much we can possibly improve our code performance, while still being practically relevant. We then look at the performance of a sliding windowed decoder which gives a small decoding latency, and see how much of the performance improvement obtained thanks to spatial coupling is still retained.

3.1 Over Fading Channels With Interleaving

We have seen in Sec. 2.4.3 that interleaved SC-LDPC codes are expected to do well over fading channels – we now verify that this does indeed happen for continuously-varying fading channels.

We study the performance of SC-LDPC codes of different blocklengths over continuously-varying fading channels of different channel variation speeds (f_{d}). If the codewords are (block) interleaved before transmission, the error performance improves on increasing the interleaving depth, at the cost of decoding latency. A higher blocklength suffers from a larger latency for the same interleaving depth. Hence we study, subject to a latency constraint, the trade-off between the maximum interleaving depth and blocklength to obtain the highest diversity.

3.1.1 The Trade-off Involved

An input codeword $\tilde{\mathbf{x}}$ of blocklength n is interleaved with interleaving depth d to obtain the interleaved codeword $\mathbf{x}$. Define the *latency* as the maximum distance between the position of a bit in $\tilde{\mathbf{x}}$ and in $\mathbf{x}$. It is easy to check that this maximum occurs at the last element of the first row in the $d \times k$ matrix A (as in Sec. 2.4.3). The position of this bit in $\tilde{\mathbf{x}}$ is k, and in $\mathbf{x}$ is $n-d+1$. The difference, hence the latency $L = n-d-k+1 = (k-1)(d-1)$. We restrict ourselves to interleaving depths under $\sqrt{n}$, giving us a monotonically increasing relation between d and L. A latency constraint is then equivalent to a maximum interleaving depth for a given blocklength. Additionally, given a fixed interleaving depth, a larger blocklength results in higher latency. We have two opposing phenomena that give us room for a trade-off.

3.1.2 Problem Setting

Suppose we are transmitting at a speed of 10 mega bits per second (Mbps), and are constrained to have a latency of under 1 ms, equivalent to $10 \times 10^6\,\text{bps} \times 1 \times 10^{-3}\,\text{s} = 10,000$ bits. For a code with length n, let d^* be the maximum interleaving depth such that this latency constraint is satisfied. We simulate BPSK modulated SC-LDPC codes of different blocklengths over fading channels with $f_\text{d} = 10^{-3}$, 10^{-4}, 10^{-5} (fast, moderate, slow fading resp.), both without interleaving and with (at this maximum interleaving depth d^*). The diversity order is calculated by Eqn. 2.8 as the slope of a least-square error linear fit. As a reference for maximum diversity, we take codes of the same lengths over an uncorrelated fading channel.

3.1.3 Results

Below are the plots and data obtained on performing the simulation described above with M-lifted $\mathcal{C}(3,6,100)$ codes with $w = 2$, and $M = 51, 52, 55, 60, 75$, giving blocklengths

BP implementation, Rayleigh process generated using IT++, `http://itpp.sourceforge.net/4.3.1/`

$n = 10200, 10400, 11000, 12000, 15000$ respectively. The program for the simulation was written in IT++ and the available BP decoder package was used.

$\underline{f_{\mathrm{d}} = 10^{-3}}$:

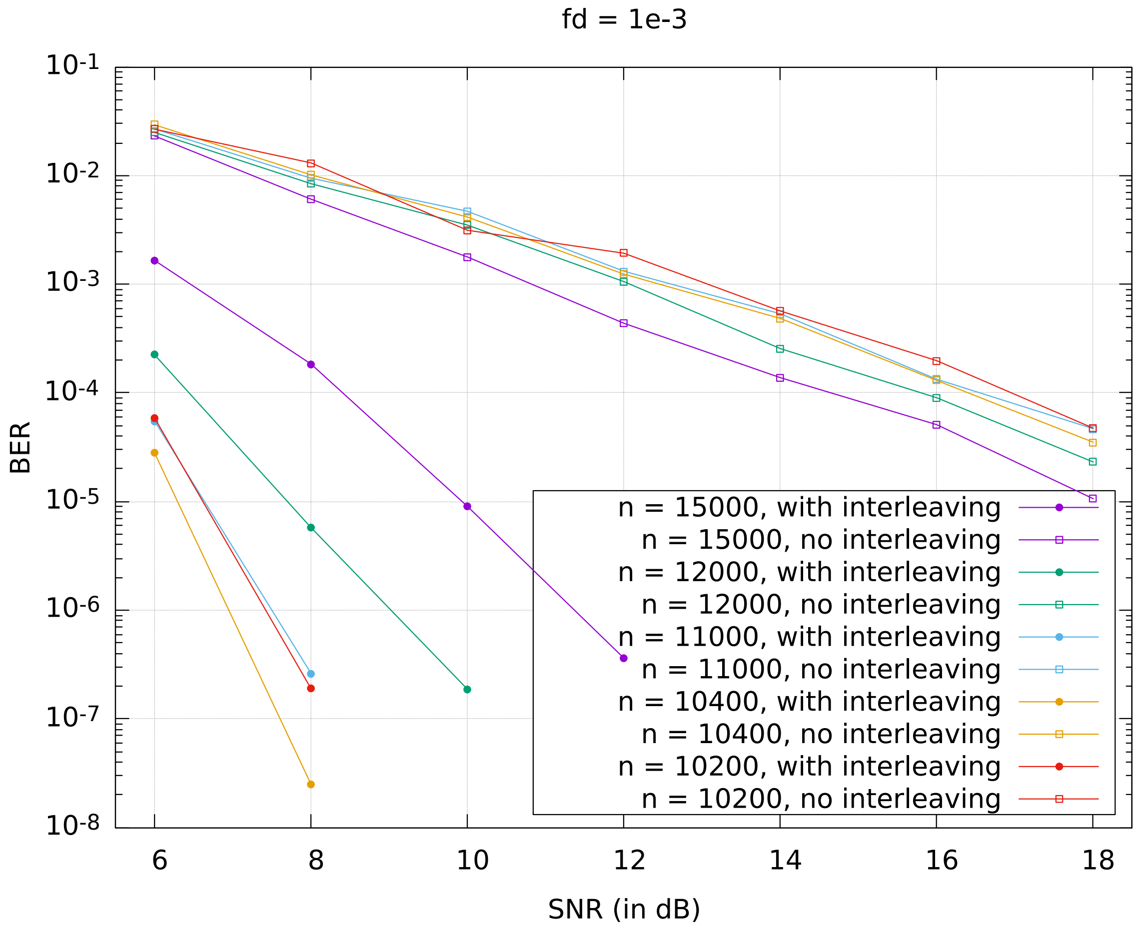


Figure 3.1: BER values for $\mathcal{C}(3, 6, 100)$ of different blocklengths, without and with interleaving; there is a large improvement with interleaving

n	d^*	Diversity	
		without interleaving	with interleaving
15000	3	2.73	6.14
12000	6	2.53	7.69
11000	11	2.31	11.6
10400	26	2.41	15.2
10200	51	2.26	12.5

Table 3.1: Diversity values calculated from Fig. 3.1; observe the rise and fall of diversity showing the trade-off

$\underline{f_{\mathrm{d}} = 10^{-4}}$:

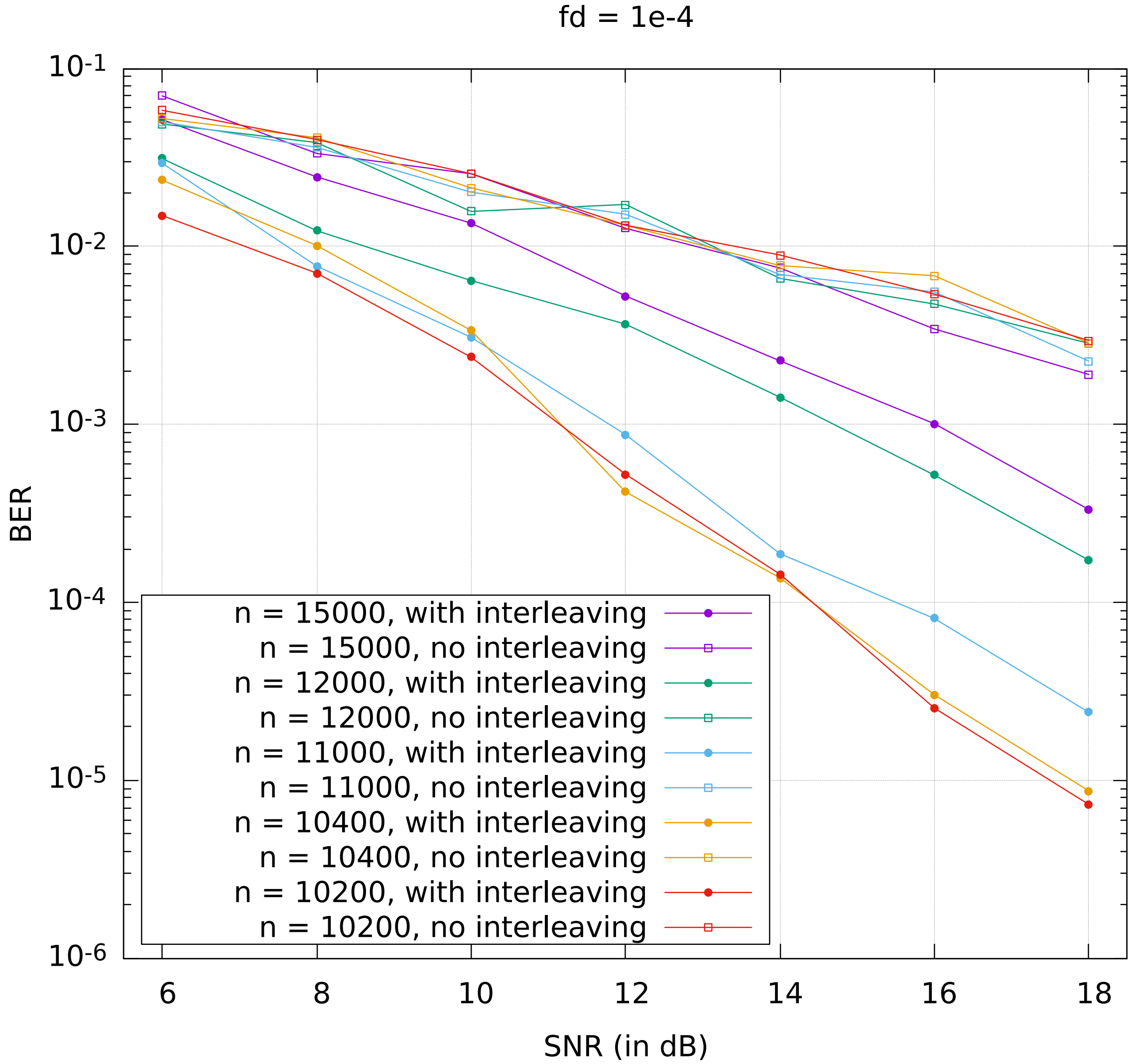


Figure 3.2: BER values for $\mathcal{C}(3, 6, 100)$ of different blocklengths, without and with interleaving; the improvement by interleaving has reduced, but there is still a clear advantage

n	d^*	Diversity	
		without interleaving	with interleaving
15000	3	1.29	1.81
12000	6	1.05	1.82
11000	11	1.09	2.58
10400	26	1.03	2.99
10200	51	1.08	2.86

Table 3.2: Diversity values calculated from Fig. 3.2; the trade-off is visible

$\underline{f_{\mathrm{d}} = 10^{-5}}$:

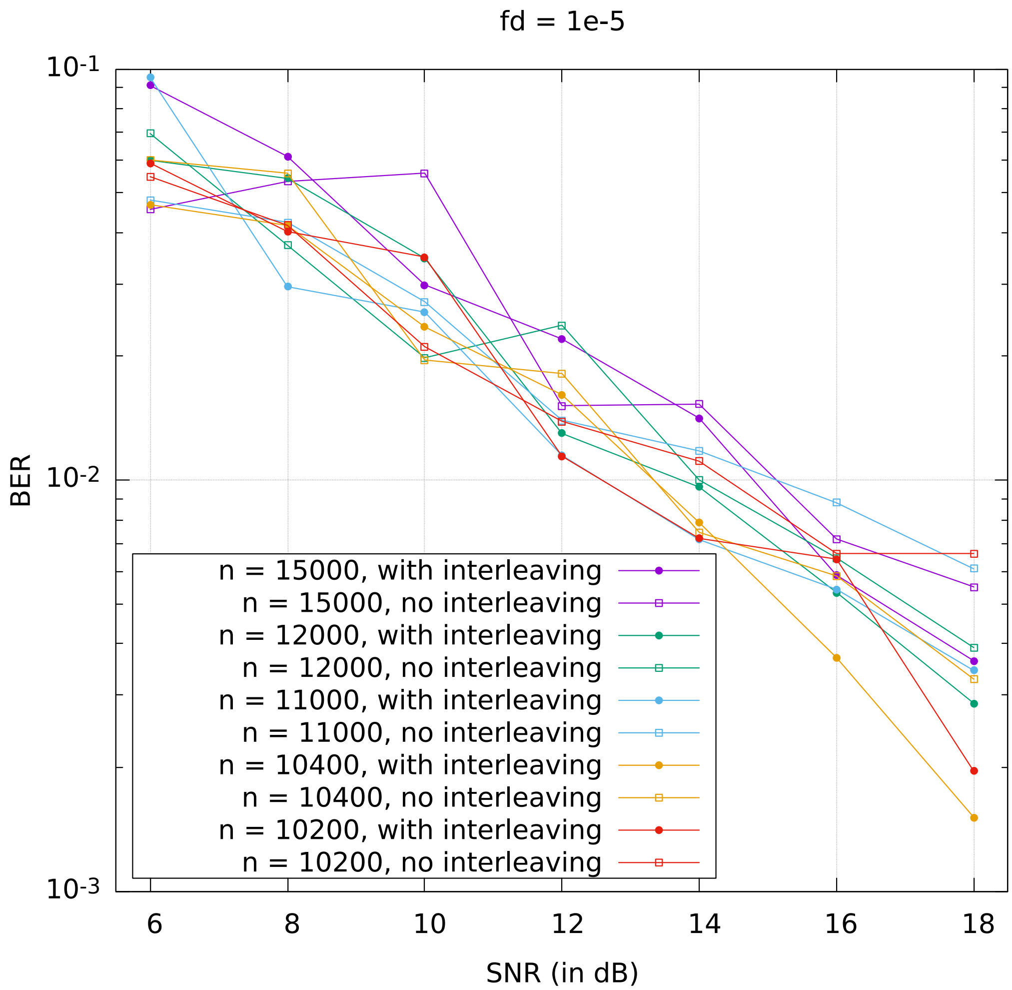


Figure 3.3: BER values for $\mathcal{C}(3, 6, 100)$ of different blocklengths, without and with interleaving; improvement produced by interleaving difficult to observe - almost overlapping (the line is not monotonic because it has not averaged over enough iterations; this suffices to observe the general trend)

n	d^*	Diversity	
		without interleaving	with interleaving
15000	3	0.9	1.17
12000	6	0.99	1.17
11000	11	0.79	1.13
10400	26	1.1	1.26
10200	51	0.83	1.2

Table 3.3: Diversity values calculated from Fig. 3.3; the trade-off is not very obvious

Independent Identically Distributed Channel:

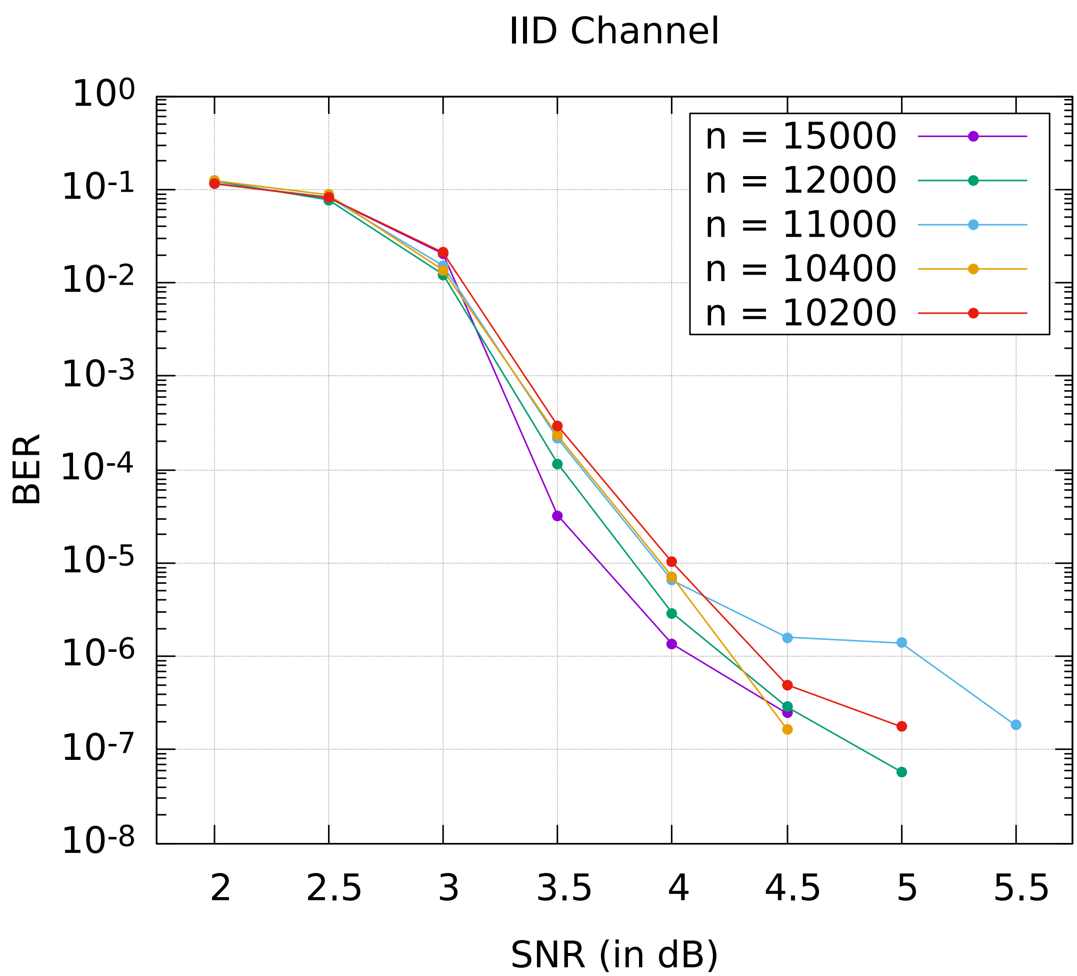


Figure 3.4: BER values for different blocklengths, over an IID fading channel without interleaving (isolated deep fades corrupted the results at high SNRs; requires more averaging to get an exact plot)

n	Diversity
15000	26
12000	23.9
11000	18.6
10400	24.8
10200	22.3

Table 3.4: Diversity values calculated from Fig. 3.4 (note that the actual diversity values are higher than these since there is some error due to the flattening of the BER curve at high SNR values)

3.1.4 Comments

We see that for a fast fading channel, interleaving helps obtain a large diversity even when subjected to a latency constraint by exploiting the trade-off and picking an appropriate blocklength and interleaving depth (as opposed to without interleaving, where there is a generally decreasing trend as blocklength is reduced). As the channel gets slower, the benefit due to interleaving becomes lesser and the trade-off is also less clear.

We are able to get closer to the diversity limit given by uncorrelated channels (≈ 15 with interleaving vs. 3 without against 26 for the uncorrelated channel) by picking the interleaving depth that gives the maximum for fast fading channels, but this gets harder for slower channels.

3.2 Low-Latency Windowed Decoding

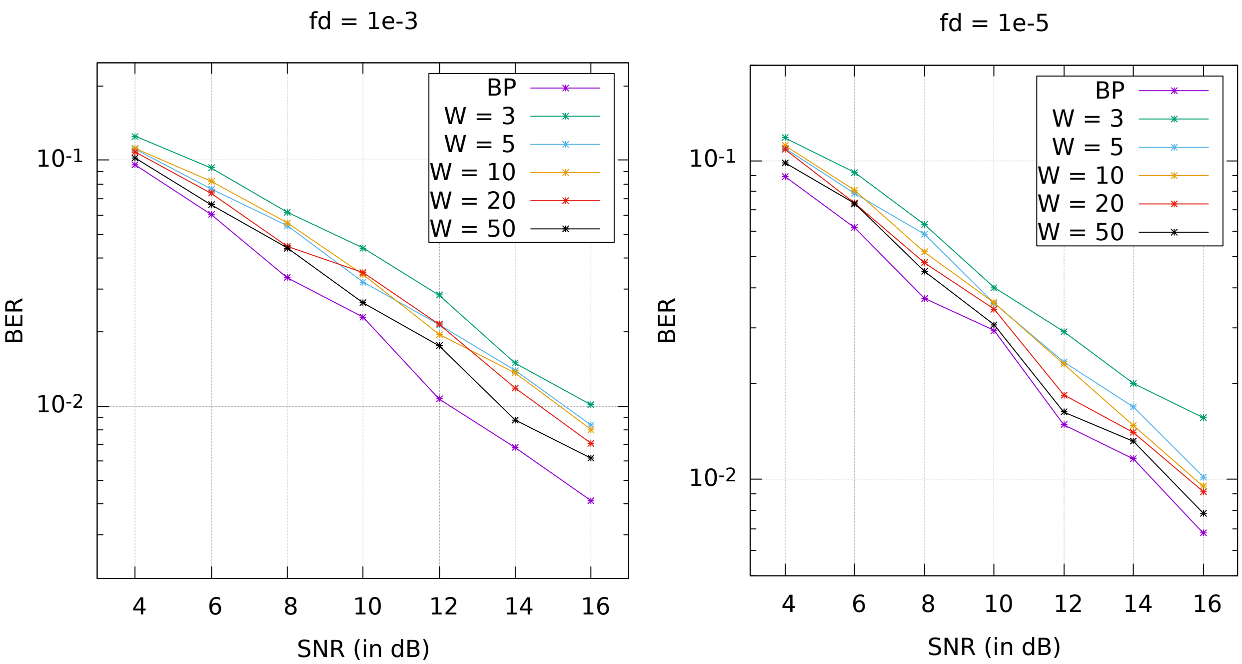
As seen in Sec. 2.5, we can use a windowed decoder that does not have to wait for the entire message to be received before starting the decoding process. This helps to reduce the decoding latency, but it comes at the cost of a worse error performance.

3.2.1 Results

We study the performance of the windowed decoder of window length $W = 3, 5, 10, 20, 50$ on 10-lifted $\mathcal{C}(3, 6, 200)$ codes with $w = 2$, for fast and slow fading channels ($f_\text{d} = 10^{-3}, 10^{-5}$ resp.) of different SNRs. The latency is reduced to $\frac{W}{L}$ times that with the full-length BP decoder. The BERs obtained are plotted against the SNR values to obtain Fig. 3.5.

3.2.2 Comments

We see that the windowed decoder performs fairly close to the full-length BP decoder even for small window lengths. In particular, for the slow fading case, $W = 5$ takes us very close to the BP performance with a reduction in latency by $\frac{W}{L} = \frac{5}{200} = 0.025$. For the fast fading case, the degradation in performance is greater but we still obtain a similar order of error performance, with large reduction in latency. This allows us to use larger blocklengths to obtain better performances without worrying about violating latency constraints.



(a) Fast fading channel

(b) Slow fading channel

Figure 3.5: Performance of windowed decoder of different window lengths over $\mathcal{C}(3, 6, 200)$ codes

BP implementation on MATLAB adapted from NPTEL: LDPC and Polar Codes in 5G Standard (Dr. Andrew Thangaraj), `https://nptel.ac.in/courses/117/106/108106137/`

Chapter 4

Conclusion

We have seen the construction and performance of Spatially Coupled LDPC Codes (Sec. 2.3). These codes can achieve significant improvements in performance over conventional LDPC codes, but realising this improvement requires large blocklengths, which cause large decoding latencies. To make these theoretical results useful from a practical perspective, we move in two contrary directions from the starting position, i.e. simply sending interleaved SC-LDPC codes over fading channels. On one hand, we look at obtaining the best performance with interleaving, without incurring unacceptably high latency (Sec. 3.1). On the other hand, we look to reduce the latency using windowed decoding, without suffering from a serious degradation in error performance (Sec. 3.2).

Bibliography

[1] C. E. Shannon. "A Mathematical Theory of Communication". In: *SIGMOBILE Mob. Comput. Commun. Rev.* 5.1 (Jan. 2001), pp. 3–55. ISSN: 1559-1662. DOI: 10.1145/584091.584093. URL: https://doi.org/10.1145/584091.584093.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: *Proceedings of ICC '93 - IEEE International Conference on Communications*. Vol. 2. 1993, 1064–1070 vol.2. DOI: 10.1109/ICC.1993.397441.

[3] R. Gallager. "Low-density parity-check codes". In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28. DOI: 10.1109/TIT.1962.1057683.

[4] Sae-Young Chung et al. "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit". In: *IEEE Communications Letters* 5.2 (2001), pp. 58–60. DOI: 10.1109/4234.905935.

[5] Shrinivas Kudekar, Thomas J. Richardson, and Rüdiger L. Urbanke. "Threshold Saturation via Spatial Coupling: Why Convolutional LDPC Ensembles Perform So Well over the BEC". In: *IEEE Transactions on Information Theory* 57.2 (2011), pp. 803–834. DOI: 10.1109/TIT.2010.2095072.

[6] Shrinivas Kudekar, Tom Richardson, and Rüdiger L. Urbanke. "Spatially Coupled Ensembles Universally Achieve Capacity Under Belief Propagation". In: *IEEE Trans-*

actions on Information Theory 59.12 (2013), pp. 7761–7813. DOI: 10.1109/TIT.2013.2280915.

[7] Ezio Biglieri, Giuseppe Caire, and Giorgio Taricco. "Coding for the Fading Channel: A Survey". In: *Signal Process.* 80.7 (July 2000), pp. 1135–1148. ISSN: 0165-1684. DOI: 10.1016/S0165-1684(00)00027-X. URL: https://doi.org/10.1016/S0165-1684(00)00027-X.

[8] Najeeb ul Hassan et al. "Improving code diversity on block-fading channels by spatial coupling". In: *2014 IEEE International Symposium on Information Theory.* 2014, pp. 2311–2315. DOI: 10.1109/ISIT.2014.6875246.

[9] Aravind R. Iyengar et al. "Windowed Decoding of Protograph-Based LDPC Convolutional Codes Over Erasure Channels". In: *IEEE Transactions on Information Theory* 58.4 (2012), pp. 2303–2320. DOI: 10.1109/TIT.2011.2177439.

[10] Aravind R. Iyengar et al. "Windowed Decoding of Spatially Coupled Codes". In: *IEEE Transactions on Information Theory* 59.4 (2013), pp. 2277–2292. DOI: 10.1109/TIT.2012.2231465.

[11] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory.* USA: Cambridge University Press, 2008. ISBN: 0521852293.

[12] David G. M. Mitchell, Michael Lentmaier, and Daniel J. Costello. "Spatially Coupled LDPC Codes Constructed From Protographs". In: *IEEE Transactions on Information Theory* 61.9 (2015), pp. 4866–4889. DOI: 10.1109/TIT.2015.2453267.

[13] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication.* USA: Cambridge University Press, 2005. ISBN: 0521845270.

[14] Yunlong Zhao, Yi Fang, and Zhaojie Yang. "Interleaver Design for Small-Coupling-Length Spatially Coupled Protograph LDPC-Coded BICM Systems Over Wireless Fading Channels". In: *IEEE Access* 8 (2020), pp. 33500–33510. DOI: 10.1109/ACCESS.2020.2974002.

[15] Sebastian Cammerer et al. "Spatially Coupled LDPC Codes and the Multiple Access Channel". In: *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. 2019, pp. 1–6. DOI: 10.1109/CISS.2019.8692899.

[16] Iryna Andriyanova et al. "SC-LDPC codes over the block-fading channel: Robustness to a synchronisation offset". In: *2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. 2015, pp. 97–101. DOI: 10.1109/BlackSeaCom.2015.7185094.