Modeling a Sliding Window Decoder for Spatially Coupled LDPC Codes

Min Zhu*, David G. M. Mitchell†, Michael Lentmaier‡, and Daniel J. Costello, Jr.§
*State Key Laboratory of ISN, Xidian University, Xi'an, P. R. China, zhunanzhumin@gmail.com
†Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM, USA, dgmm@nmsu.edu
‡Department of Electrical and Information Technology, Lund University, Lund, Sweden, michael.lentmaier@eit.lth.se
§Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA, dcostel1@nd.edu

2021 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-2390-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/GCWkshps52748.2021.9682064

***Abstract*—Due to their capacity achieving performance with sliding window decoding (SWD), spatially coupled LDPC (SC-LDPC) codes are emerging as candidates for next generation channel coding applications. In this paper we present a general model of SWD of SC-LDPC codes and develop an analysis that allows us to estimate error probability performance under decoder error propagation conditions that can occur when low latency operation is desired. We also show how the model parameters can be estimated and indicate how the model can be used to predict the performance of code doping techniques used to mitigate the effects of decoder error propagation.**

***Index Terms*—SC-LDPC codes, sliding window decoding, code doping, decoder error propagation**

I. Introduction

Low-density parity-check (LDPC) convolutional codes [1], [2], also known as *spatially coupled* LDPC (SC-LDPC) codes [3], [4], have been shown to have two advantages compared to LDPC block codes (LDPC-BCs): (1) they exhibit *threshold saturation*, i. e., the **suboptimal** *belief propagation* (BP) iterative decoding threshold of spatially coupled code ensembles over memoryless binary-input symmetric-output channels coincides with the **optimal** MAP threshold of their underlying block code ensembles [2]–[4], and (2) *sliding window decoding* (SWD) can be employed to reduce decoding latency, memory, and complexity [5]. For SC-LDPC codes to achieve near optimal performance at moderate-to-high *signal-to-noise* ratios (SNRs), the decoder *window size* W should satisfy $W \geq 6v$, where v is the *decoding constraint length*. However, due to the fact that low-latency operation is desirable in practice, a smaller window size is often required. Under such low latency operating conditions, infrequent but severe *decoder error propagation* can sometimes occur. During the decoding process, error propagation is triggered when, after a decoding error occurs, the decoding of subsequent bits is also affected. This in turn can cause a continuous string of errors, resulting in an unacceptable performance loss. This effect is particularly harmful for large frame lengths or a continuous (streaming) transmission scenario.

The effect of SWD error propagation on the performance of SC-LDPC codes was studied in [6], where the authors proposed adapting the number of decoder iterations and/or shifting the window position in order to limit the effects of error propagation. As an alternative to modifying the design of the decoder, *check node* (CN) *doped* [7] and *variable node* (VN) *doped* [8] SC-LDPC codes were proposed to limit error propagation by altering the encoder design. In this approach, doped CNs or VNs inserted periodically into the encoded sequence allow the decoder to recover from error propagation. In [9], a scaling law to predict the performance of periodically-doped SC-LDPC code ensembles streamed over the binary erasure channel (BEC) was presented, and [10] proposed a model characterizing the BEC performance of SWD as a function of W.

To help understand the error propagation problem in SWD of SC-LDPC codes, a simplified two-state decoder models was proposed in [7]. In this paper, we extend and generalize this concept in a way that allows us to estimate decoder performance for a wide variety of error conditions. While the simplified model is primarily useful for studying situations where most decoding errors are due to error propagation, the multi-state models developed here allow us to analyze decoders that can produce error bursts of various lengths. This added capability gives us the flexibility of analyzing the performance of different code designs over a broad range of encoder/decoder parameters and channel conditions.

II. Modeling and Analysis of SWD

We consider SC-LDPC codes constructed by coupling together a sequence of L disjoint (J,K)*-regular* LDPC-BC *protographs* into a single coupled chain, where infinite L results in an *unterminated* code and finite L results in a *terminated* code. Without loss of generality, we consider (3,6)-regular SC-LDPC codes as shown in Fig. 1. As an example, we begin with an independent (uncoupled) sequence of (3,6)-regular LDPC-BC protographs with *base matrix* $\mathbf{B} = [3 \;\; 3]$ (see Fig. 1(a)). The unterminated (3,6)-regular SC-LDPC code chain is obtained by applying the *edge-spreading* technique of [4] to the uncoupled protographs. In this case, the edge spreading is defined by a set of component base matrices $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = [1 \;\; 1]$ that satisfy $\mathbf{B} = \mathbf{B}_0 + \mathbf{B}_1 + \mathbf{B}_2$ (see Fig. 1(b)). In general, an arbitrary edge spreading must satisfy $\mathbf{B} = \sum_{i=0}^{\omega} \mathbf{B}_i$, where ω is the *coupling width*. Applying the *lifting factor* M to the SC-LDPC protograph of Fig. 1(b) results in an unterminated ensemble of (3,6)-regular SC-LDPC codes in which each time unit represents a *block* of $2M$ coded bits (VNs). SWD, first proposed in [5], was applied to SC-LDPC codes to reduce decoding latency, memory,

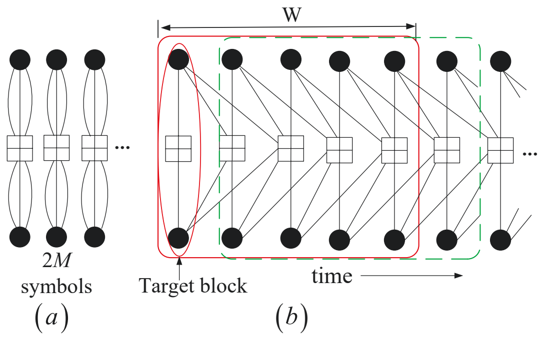

Fig. 1. (a) A sequence of independent (uncoupled) protographs; (b) spreading edges to the $\omega = 2$ nearest neighbors.

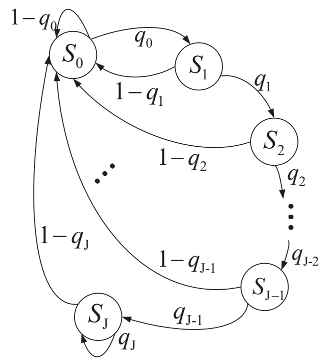

Fig. 2. The state diagram of a general decoder model for SWD.

and complexity. As shown in Fig. 1(b), the rectangular box represents a decoding window of size W blocks. To decode on a binary-input *additive white Gaussian noise* (AWGN) channel, (1) a BP flooding schedule is applied to all the nodes in the window until some stopping criterion is met, up to some *maximum number of iterations* I, (2) the *target block* of $2M$ symbols in the first window position is decoded according to the signs of their *log-likelihood ratios* (LLRs), and (3) the window shifts one time unit (block) to the right. Decoding continues in the same fashion until the entire chain is decoded, where the decoding latency in bits is given by $2MW$.

In order to describe the behavior of SWD under low latency (small W) operating conditions, we introduce a generalization of the simplified decoder model of [7] that includes a *random error state* for normal decoder behavior, some number of *intermediate states* that account for finite-length error bursts, and a *burst error state* that allows for the possibility of unlimited decoder error propagation. The corresponding state transition diagram is shown in Fig. 2, where S_0 represents the random error state, $\{S_i\}$, $i = 1, 2, \ldots, J-1$, represents the intermediate states, and S_J represents the burst error state. We let q_0 denote the *block error rate* (BLER) in state S_0, q_i, $i = 1, 2, \ldots, J-1$, denote the BLER in state S_i, and q_J denote the BLER in state S_J.

Referring to Fig. 2, the decoder starts in the random error state S_0, transitions to intermediate state S_1 with probability q_0 when the first block error occurs, and then either makes a second block error and transitions to state S_2 with probability q_1 or decodes correctly and returns to state S_0 with probability $1 - q_1$, resulting in a single block decoding error. Generally, $q_1 \geq q_0$ since one block decoded in error means that there are some *incorrectly* decoded symbols still connected to the window (see Fig. 1(b)) that influence the decoding of the next block. If, with probability q_2, a third block error occurs and the decoder transitions to state S_3, the decoding window now contains incorrectly decoded symbols from the past two blocks, which, using the same reasoning as above, implies that $q_2 \geq q_1 \geq q_0$, while the decoder returns to state S_0 with probability $1 - q_2$, resulting in a double burst error. Extending the same argument and noting that there are in general ω previously decoded blocks still connected to the window during the decoding of a target block, we have that $q_{\mathrm{J}-1} \geq q_{\mathrm{J}-2} \geq \ldots \geq q_1 \geq q_0$, and a correctly decoded block in intermediate state S_{i} causes the decoder to return to state S_0 and results in a length i burst error, $i = 1, 2, \ldots, J-1$. If J consecutive block errors occur, the decoder transitions to the burst error state S_{J} and remains there with probability $q_{\mathrm{J}} \geq q_{\mathrm{J}-1}$ as long as decoding errors continue, returning to state S_0 with probability $1 - q_{\mathrm{J}}$, resulting in a burst error of length J or more.[1] If the influence of the ω previously decoded incorrect blocks is strong enough, such that $q_{\mathrm{J}} \rightarrow 1$, unlimited error propagation can result, i.e., the decoder will typically not be able to escape the burst error state.

For a given spatially coupled protograph with coupling width ω, the *channel parameter* SNR, the *decoder parameter* W, and the *code parameter* M will all influence the values of $q_0, q_1, \ldots, q_J$. Generally, these probabilities are decreasing functions of all three of these parameters. However, as M increases, the larger number of decoded symbols still connected to the window has a stronger influence on future decoded blocks. As a result, strong codes (large M), although they have smaller values of q_0 and thus are less likely to reach state S_J, once in the burst state they have a high probability of staying there, i.e., a high probability of unlimited decoder error propagation. Typically, they suffer only a very few short burst errors, and their BLER performance is dominated by error propagation. Weak codes (small M), on the other hand, have larger values of q_0, and thus reach state S_J more often, but are less likely to suffer from unlimited decoder error propagation. Instead, their BLER performance is typically dominated by larger numbers of burst errors of varying lengths.[2] We now proceed to derive expressions for the BLER, as functions of $q_0, q_1, \ldots, q_J$, of SC-LDPC codes based on this model for both unterminated ($L \rightarrow \infty$) and terminated (finite L) transmission.

[1]We note here that, when the decoder returns to state S_0 after one or more blocks errors, there are up to $w-1$ incorrectly decoded blocks still connected to the window, which could effect the value of q_0. The fact that the most recent block was decoded correctly, however, suggests that this effect is minor, and for simplicity we choose to ignore it in the model.

[2]The relative strength of a code increases with ω as well as with M, while decoder strength increases with both W and I.

A. Asymptotic ($L \to \infty$) Analysis

Case I: No intermediate states ($J = 1$), $L \to \infty$.[3]

Let p_{i} be the probability of being in state S_{i}, $i = 0, 1, \ldots, J$. Then $p_0 = p_0(1-q_0) + p_1(1-q_1)$, and hence,

$$p_1 = \frac{p_0 - p_0(1-q_0)}{1-q_1} = \frac{p_0 q_0}{1-q_1}. \tag{1}$$

Now the *average BLER* can be written as

$$P_{\mathrm{BL}}^{(\infty)} = p_0 q_0 + p_1 q_1 = p_0 q_0 + p_0 q_1 \left(\frac{q_0}{1-q_1}\right) = p_0 \left(\frac{q_0}{1-q_1}\right). \tag{2}$$

Since $p_0 + p_1 = p_0 + p_0\left(\frac{q_0}{1-q_1}\right) = p_0\left(\frac{1-q_1+q_0}{1-q_1}\right) = 1$,

$$p_0 = \frac{1-q_1}{1-q_1+q_0}, \tag{3}$$

and using (3) in (2) it follows that

$$P_{\mathrm{BL}}^{(\infty)} = \frac{q_0}{1-q_1+q_0} = \frac{r_1}{1-q_1+r_1}, \tag{4}$$

where $r_1 \triangleq q_0$. Note that when $q_1 \to 1$, $\lim\limits_{q_1 \to 1} P_{\mathrm{BL}}^{(\infty)} = 1$, which corresponds to unlimited error propagation.

Example 1: Choose $q_0 = 0.01, q_1 = 0.99$.[4] Then $P_{\mathrm{BL}}^{(\infty)} = 0.5$. □

Case II: One intermediate state ($J = 2$), $L \to \infty$.

In this case, we have

$$\begin{cases} p_0 = p_0(1-q_0) + p_1(1-q_1) + p_2(1-q_2) \\ p_1 = p_0 q_0, p_2 = p_1 q_1 + p_2 q_2, \end{cases} \tag{5}$$

where p_2 can also be written as $p_2 = p_1 \frac{q_1}{1-q_2} = p_0 \frac{q_0 q_1}{1-q_2}$. Since $p_0 + p_1 + p_2 = p_0 + p_0 q_0 + p_0 \frac{q_0 q_1}{1-q_2} = 1$, it follows that

$$p_0 = \frac{1}{1 + q_0 + \frac{q_0 q_1}{1-q_2}} = \frac{1-q_2}{1-q_2+q_0-q_0 q_2+q_0 q_1}. \tag{6}$$

Now the average BLER can be written as

$$P_{\mathrm{BL}}^{(\infty)} = p_0 q_0 + p_1 q_1 + p_2 q_2 = \frac{q_0 - q_0 q_2 + q_0 q_1}{1-q_2+q_0-q_0 q_2+q_0 q_1}. \tag{7}$$

Defining $r_2 \triangleq q_0(1-q_2+q_1)$, (7) can be written as

$$P_{\mathrm{BL}}^{(\infty)} = \frac{r_2}{1-q_2+r_2}, \tag{8}$$

and we see that $\lim\limits_{q_2 \to 1} P_{\mathrm{BL}}^{(\infty)} = 1$ (unlimited error propagation).

Example 2: Choose $q_0 = 0.01$, $q_1 = 0.1$, $q_2 = 0.99$. It follows that $r_2 = 0.0011$ and $P_{\mathrm{BL}}^{(\infty)} = 0.099$. □

Case III: Two intermediate states ($J = 3$), $L \to \infty$.

In this case, we have

$$\begin{cases} p_0 = p_0(1-q_0) + p_1(1-q_1) + p_2(1-q_2) + p_3(1-q_3), \\ p_1 = p_0 q_0, p_2 = p_1 q_1, \\ p_3 = p_2 q_2 + p_3 q_3 = p_1 q_1 q_2 + p_3 q_3 = p_0 q_0 q_1 q_2 + p_3 q_3, \end{cases}$$

from which it follows that

$$p_3 = p_0 \frac{q_0 q_1 q_2}{1-q_3}.$$

Now, since $p_0 + p_1 + p_2 + p_3 = 1$, we have

$$\begin{aligned} p_0 &= \frac{1}{1 + q_0 + q_0 q_1 + \frac{q_0 q_1 q_2}{1-q_3}} \\ &= \frac{1-q_3}{1-q_3+q_0-q_0 q_3+q_0 q_1-q_0 q_1 q_3+q_0 q_1 q_2}. \end{aligned}$$

Defining $r_3 \triangleq q_0(1-q_3+q_1-q_1 q_3+q_1 q_2)$, it follows that $p_0 = \frac{1-q_3}{1-q_3+r_3}$, and the average BLER can be expressed as

$$\begin{aligned} P_{\mathrm{BL}}^{(\infty)} &= p_0 q_0 + p_1 q_1 + p_2 q_2 + p_3 q_3 \\ &= p_0 q_0 + p_0 q_0 q_1 + p_0 q_0 q_1 q_2 + p_0 \frac{q_0 q_1 q_2 q_3}{1-q_3} \\ &= p_0 \frac{r_3}{1-q_3} = \frac{r_3}{1-q_3+r_3}, \end{aligned} \tag{9}$$

and $\lim\limits_{q_3 \to 1} P_{\mathrm{BL}}^{(\infty)} = 1$ (unlimited error propagation).

Example 3: Choose $q_0 = 0.01$, $q_1 = 0.1$, $q_2 = 0.5$, and $q_3 = 0.9999$. Then $r_3 = 5.011 \times 10^{-4}$ and $P_{\mathrm{BL}}^{(\infty)} = 0.834$.[5] □

Case IV: $J \geq 4$, $L \to \infty$.

Starting with $J = 4$, we have

$$\begin{cases} p_0 = p_0(1-q_0) + p_1(1-q_1) + p_2(1-q_2) \\ \qquad + p_3(1-q_3) + p_4(1-q_4) \\ p_1 = p_0 q_0, p_2 = p_1 q_1, p_3 = p_2 q_2, \\ p_4 = p_3 q_3 + p_4 q_4 = p_0 q_0 q_1 q_2 q_3 + p_4 q_4, \end{cases} \tag{10}$$

from which it follows that $p_4 = p_0 \frac{q_0 q_1 q_2 q_3}{1-q_4}$. Since $p_0 + p_1 + p_2 + p_3 + p_4 = 1$, we have

$$p_0 = \frac{1}{1 + q_0 + q_0 q_1 + q_0 q_1 q_2 + \frac{q_0 q_1 q_2 q_3}{1-q_4}} = \frac{1-q_4}{1-q_4+r_4}, \tag{11}$$

where $r_4 \triangleq q_0(1-q_4+q_1-q_1 q_4+q_1 q_2-q_1 q_2 q_4+q_1 q_2 q_3)$. Then, using the same approach as in (9), the average BLER can be expressed as $P_{\mathrm{BL}}^{(\infty)} = r_4/(1-q_4+r_4)$, and $\lim\limits_{q_4 \to 1} P_{\mathrm{BL}}^{(\infty)} = 1$ (unlimited error propagation).

Example 4: Choose $q_0 = 0.01$, $q_1 = 0.1$, $q_2 = 0.5$, $q_3 = 0.9$, $q_4 = 0.999$. Then $r_4 = 4.6 \times 10^{-4}$ and $P_{\mathrm{BL}}^{(\infty)} = 0.316$. □

For $J > 4$, the general expression for r_J is:

$$\begin{aligned} r_J = & q_0\{1 - q_J + q_1\{1 - q_J + \\ & q_2[1 - q_J + \cdots + q_{J-2}(1 - q_J + q_{J-1})]\}\} \end{aligned} \tag{12}$$

and the general expression for the average BLER is given by

$$P_{\mathrm{BL}}^{(\infty)} = \frac{r_J}{1-q_J+r_J}. \tag{13}$$

B. Finite L Analysis

Case V: No intermediate states ($J = 1$), finite L.[3]

Let $d_0 = 1/q_0$ denote the *average dwell time* in state S_0, i.e., the average number of time units the decoder stays in the random error state, and let $d_1 = 1/(1-q_1)$ denote the average dwell time in state S_1. In this case, there is one *cycle* in the graph, the *average cycle time* is given by

$$x = d_0 + d_1 = \frac{1}{q_0} + \frac{1}{1-q_1} = \frac{1-q_1+q_0}{q_0(1-q_1)} = \frac{1-q_1+r_1}{q_0(1-q_1)}, \tag{14}$$

[3] $J = 1$ is the "pure" error propagation case, where a single error puts SWD in the burst state S_J.

[4] The parameter values here and in subsequent examples were chosen to be representative of those encountered in practice.

[5] Note that the large value of q_3 in this case leads to significant decoder error propagation, which severely degrades performance.

and the *average number of cycles* is $y = L/x$.[6]

Now we write $y = \lfloor y \rfloor + z$, where $z < 1$. Let $U_0 = \frac{d_0}{x} = \frac{d_0}{d_0+d_1}$ be the *average fraction of a cycle* spent in S_0, $U_1 = 1 - U_0 = \frac{d_1}{d_0+d_1}$ be the average fraction of a cycle spent in S_1, $\bar{n}_0$ be the *average number of block errors* in state S_0, and $\bar{n}_1$ be the average number of block errors in state S_1. Since the decoder makes an error each time it leaves state S_0, we have

$$\bar{n}_0 = \begin{cases} \lfloor y \rfloor d_0 q_0 = \lfloor y \rfloor, \text{for } z < U_0 \, (\text{since } d_0 q_0 = 1) \\ (\lfloor y \rfloor + 1) d_0 q_0 = \lfloor y \rfloor + 1, \text{for z} \geq U_0, \end{cases} \tag{15}$$

$$\bar{n}_1 = \begin{cases} \lfloor y \rfloor d_1 q_1 = \lfloor y \rfloor \frac{q_1}{1-q_1}, \text{for } z < U_0 \\ \left(\lfloor y \rfloor + \frac{(z-U_0)}{U_1} \right) \frac{q_1}{1-q_1}, \text{for z} \geq U_0, \end{cases} \tag{16}$$

and the *average BLER* can be expressed as

$$P_{\rm BL}^{(L)} = \frac{\bar{n}_0 + \bar{n}_1}{L}. \tag{17}$$

Example 5: Choose $L = 1000$, q_0 = 0.01, and $q_1 = 0.99$. Hence we have $d_0 = 100$, $d_1 = 100$, $x = 200$, $y = 5$, $z = 0$, and $U_0 = U_1 = 0.5$. Substituting these values into (15) and (16), we obtain $\bar{n}_0 = 5$, $\bar{n}_1 = 495$, and the average BLER is given by $P_{\rm BL}^{(1000)} = \frac{5+495}{1000} = 0.5$.

In this case, for any $L > d_0 = 100$, if $q_1 \to 1$, we have $d_1 \to \infty$, $x \to \infty$, $y = z = \frac{L}{x} > U_0 = \frac{d_0}{x}$, and $\lfloor y \rfloor = 0$. Then $\bar{n}_0 = 1$, $\bar{n}_1 = \frac{z-U_0}{U_1} d_1 q_1 = (L - d_0) q_1 = L - 100$, and $\lim\limits_{L\to\infty} P_{\rm BL}^{(L)} = \frac{1+(L-100)}{L} = 1$. Finally, we note that, for any L such that $z = 0$, i.e., when the decoder traverses an integer number of cycles, $P_{\rm BL}^{(L)} = P_{\rm BL}^{(\infty)}$. $\square$

Case VI: One intermediate state ($J = 2$), finite L.

For the case of one intermediate state, there are two possible cycles: $S_0S_1S_0$ and $S_0S_1S_2S_0$, called type 1 (denoted $C^{(1)}$) and type 2 (denoted $C^{(2)}$), respectively. We denote the average dwell times in each cycle as $d_i^{(k)}, i = 0, 1, 2, k = 1, 2$, where we see that $d_0^{(1)} = d_0^{(2)} = \frac{1}{q_0}$, $d_1^{(1)} = d_1^{(2)} = 1$, and $d_2^{(1)} = 0$, $d_2^{(2)} = \frac{1}{1-q_2}$, reflecting the facts that the dwell time in the intermediate state is always 1 time unit and state S_2 is never reached in cycle $C^{(1)}$.

Now let $x^{(1)} = {d_0}^{(1)} + {d_1}^{(1)} = \frac{1}{q_0} + 1$ be the average cycle time for $C^{(1)}$, $x^{(2)} = {d_0}^{(2)} + {d_1}^{(2)} + {d_2}^{(2)} = \frac{1}{q_0} + 1 + \frac{1}{1-q_2}$ be the average cycle time for $C^{(2)}$, and $x = x^{(1)} P\left(C^{(1)}\right) + x^{(2)} P\left(C^{(2)}\right)$ be the *overall average cycle time*, where $P\left(C^{(k)}\right)$ is the probability that a cycle is of type $k, k = 1, 2$. Then, since $P\left(C^{(1)}\right) = 1 - q_1$, $P\left(C^{(2)}\right) = q_1$, and $P\left(C^{(1)}\right) + P\left(C^{(2)}\right) = 1$, we have

$$\begin{aligned} x &= \frac{1+q_0}{q_0} (1 - q_1) + \frac{q_0 + (1+q_0)(1-q_2)}{q_0(1-q_2)} q_1 \\ &= \frac{1 - q_2 + q_0(1 - q_2 + q_1)}{q_0(1-q_2)} = \frac{1 - q_2 + r_2}{q_0(1-q_2)}. \end{aligned} \tag{18}$$

Next let $y = \frac{L}{x} = \lfloor y \rfloor + z$, $z < 1$, be the average number of cycles, $\bar{d}_0 = d_0^{(1)} = d_0^{(2)} = \frac{1}{q_0}$, $\bar{d}_1 = d_1^{(1)} = d_1^{(2)} = 1$, and $\bar{d}_2 = d_2^{(1)} \cdot P\left(C^{(1)}\right) + {d_2}^{(2)} \cdot P\left(C^{(2)}\right) = \frac{q_1}{1-q_2}$ be the *overall average dwell time* in state S_2. Then $U_0 = \frac{\bar{d}_0}{x} = \frac{1}{q_0 x}$ is the average fraction of a cycle spent in S_0, $U_1 = \frac{\bar{d}_1}{x} = \frac{1}{x}$ is the average fraction of a cycle spent in S_1, and $U_2 = \frac{\bar{d}_2}{x} = \frac{q_1}{(1-q_2)x}$ is the average fraction of a cycle spent in S_2, where

$$\begin{aligned} U_0 + U_1 + U_2 &= \frac{1}{q_0 x} + \frac{1}{x} + \frac{q_1}{(1-q_2)x} \\ &= \frac{(1-q_2) + q_0(1-q_2) + q_0 q_1}{q_0(1-q_2)x} \\ &= \frac{1 - q_2 + r_2}{q_0(1-q_2)x} = 1. \end{aligned}$$

Letting $\bar{n}_i$ be the average number of block errors in state $S_i, i = 0, 1, 2$, we then have[7]

$$\overline{n}_0 = \begin{cases} \lfloor y \rfloor \bar{d}_0 q_0 = \lfloor y \rfloor \quad \text{for} \quad z < U_0 \\ (\lfloor y \rfloor + 1) \bar{d}_0 q_0 = \lfloor y \rfloor + 1 \quad \text{for} \quad z \geq U_0 \end{cases} \tag{19}$$

$$\overline{n}_1 = \begin{cases} \lfloor y \rfloor \bar{d}_1 q_1 = \lfloor y \rfloor q_1 \quad \text{for } z < U_0 + U_1 \\ (\lfloor y \rfloor + 1) \bar{d}_1 q_1 = (\lfloor y \rfloor + 1) q_1 \quad \text{for} \quad z \geq U_0 + U_1 \end{cases} \tag{20}$$

$$\overline{n}_2 = \begin{cases} \lfloor y \rfloor \bar{d}_2 q_2 = \lfloor y \rfloor \frac{q_1 q_2}{1-q_2} & \text{for } z < U_0 + U_1 \\ \left(\lfloor y \rfloor + \frac{z-U_0-U_1}{U_2} \right) \frac{q_1 q_2}{1-q_2} & \text{for } z \geq U_0 + U_1, \end{cases} \tag{21}$$

and the average BLER can be written as

$$P_{\rm BL}^{(L)} = \frac{\overline{n}_0 + \overline{n}_1 + \overline{n}_2}{L}. \tag{22}$$

Example 6: Choose $L = 1000$, $q_0 = 0.01$, $q_1 = 0.1$, and $q_2 = 0.99$. Then we have $r_1 = 0.0011$, $\bar{d}_0 = 100$, $\bar{d}_1 = 1$, $\bar{d}_2 = 10$, $x = 111$, $y = 9.009$, $\lfloor y \rfloor = 9$, $z = 0.009$, $U_0 = 0.9009$, $U_1 = 0.009$, and $U_2 = 0.0901$ $(z < U_0 < U_0 + U_1)$. Now using (19), (20), and (21), we obtain $\overline{n}_0 = 9$, $\overline{n}_1 = 0.9$, $\overline{n}_2 = 89.1$, and $P_{\rm BL}^{(1000)} = \frac{99}{1000} = 0.099$. In this case, we see that $P_{\rm BL}^{(1000)} \approx P_{\rm BL}^{(\infty)}$ (from ***Ex. 2***), since the last cycle never reaches state S_2 on the average ($z < U_0 + U_1$). $\square$

Case VII:$J \geq 3$, finite L.

From (14) and (18), it follows that

$$x = \frac{1 - q_J + r_J}{q_0(1-q_J)} \tag{23}$$

is the general expression for the average cycle time and

$$U_i = \frac{\bar{d}_i}{x}, \quad i = 0, 1, \ldots, J. \tag{24}$$

Letting $C^{(k)}$ represent cycle $S_0S_1S_2 \ldots S_kS_0$, $k = 1, 2, \ldots, J$, we can write

$$\begin{cases} P\left(C^{(1)}\right) = 1 - q_1 \\ P\left(C^{(k)}\right) = q_1 q_2 \cdots q_{k-1}(1-q_k), k = 2, 3, \ldots, J-1 \\ P\left(C^{(J)}\right) = q_1 q_2 \cdots q_{J-1}, \end{cases} \tag{25}$$

[6] Unlike the asymptotic analysis, the finite L analysis must introduce the concepts of average dwell time and average cycle time to account for the fact that frames typically end somewhere in the middle of a cycle.

[7] We note that (1) the decoder makes exactly one error in state S_0 per cycle, (2) the decoder makes an error in state S_1 with probability q_1 per cycle, and (3) during cycle $C^{(2)}$, the decoder makes errors in state S_2 with probability q_2.

where we note that each graph contains exactly J cycles. Now it follows that the overall average dwell times are

$$\begin{cases} \bar{d}_0 = \dfrac{1}{q_0}, \quad \bar{d}_1 = 1, \\ \bar{d}_i = 1 - \displaystyle\sum_{k=1}^{i-1} P\left(C^{(k)}\right) = q_1 q_2 \cdots q_{i-1}, i = 2, 3, \ldots, J-1 \\ \bar{d}_J = \dfrac{q_1 q_2 \cdots q_{J-1}}{1 - q_J}, \end{cases} \tag{26}$$

where the average dwell times $d_i{}^{(k)} = 1, k = J-i-1, J-i, \ldots, J$, for each intermediate state $i = 1, 2, \ldots, J-1$.

The average number of block errors in each state are then

$$\bar{n}_0 = \begin{cases} \lfloor y \rfloor \bar{d}_0 q_0 = \lfloor y \rfloor & \text{for } z < U_0 \\ (\lfloor y \rfloor + 1) \bar{d}_0 q_0 = \lfloor y \rfloor + 1 & \text{for z} \geq U_0 \end{cases} \tag{27}$$

$$\bar{n}_j = \begin{cases} \lfloor y \rfloor \bar{d}_j q_j = \lfloor y \rfloor q_1 q_2 \cdots q_j \\ \quad \text{for } z < U_0 + U_1 + \cdots + U_j \\ (\lfloor y \rfloor + 1) \bar{d}_j q_j = (\lfloor y \rfloor + 1) q_1 q_2 \cdots q_j, \\ j = 1, 2, \ldots, J-1, \quad \text{for z} \geq U_0 + U_1 + \cdots + U_j \end{cases} \tag{28}$$

$$\bar{n}_J = \begin{cases} \lfloor y \rfloor \bar{d}_J q_J = \lfloor y \rfloor \frac{q_1 q_2 \cdots q_J}{1 - q_J} \\ \quad \text{for } z < U_0 + U_1 + \cdots + U_{J-1} \\ \left(\lfloor y \rfloor + \frac{z - U_0 - U_1 - \cdots - U_{J-1}}{U_J}\right) \frac{q_1 q_2 \cdots q_J}{1 - q_J} \\ \quad \text{for z} \geq U_0 + U_1 + \cdots + U_{J-1}, \end{cases} \tag{29}$$

and the average BLER is given by

$$P_{\text{BL}}^{(L)} = \sum_{i=0}^{J} \bar{n}_i / L, \tag{30}$$

which represents the general expression for the BLER in the finite L case.

III. Estimating the Model Parameters

We now describe how the model parameters are determined. First, we choose an operating channel SNR of interest, typically somewhat below the iterative decoding threshold of the underlying LDPC-BC. Then we simulate the BLER performance of the SC-LDPC code at that SNR and produce a data file of the burst length distribution of all *finite-length error bursts*, i.e., error bursts that return to state S_0. Given that we have simulated a total of N frames, each of length L, for a total of LN simulated blocks[8], this data file gives the total number of *finite-length error bursts* of each length contained in all N frames. We also produce a second data file that gives the burst length distribution of all *end of frame (EOF) error* bursts, i.e., bursts of one or more block errors at the end of a frame.

Next we decide the number of states to be included in the model, i. e., we must set the value of J. Typically, most finite-length error bursts are short, since longer bursts tend to lead to unlimited error propagation. In order to limit the size of the model, we normally choose J just large enough to include those burst lengths that occur most often, while combining the occasional longer finite-length bursts with the EOF bursts.[9]

Once J has been set, it is straightforward to determine the model parameters. We begin by letting λ_j be the number of burst errors of length j, $j = 1, 2, \ldots, J$, where $\lambda_J = \lambda_{\text{FL}} + \lambda_{\text{EOF}}$, λ_{FL} is the number of finite-length error bursts of length J or greater (if any), and λ_{EOF} is the number of EOF bursts. We also let $\delta_J = \delta_{\text{FL}} + \delta_{\text{EOF}}$ be the total number of block errors in the error bursts that comprise λ_J. Then, recalling that the dwell time in each intermediate state is exactly one time unit, the total number of time units T_j spent in state S_j is

$$T_j = \sum_{i=j}^{J} \lambda_i, \quad j = 1, 2, \ldots, J-1, \tag{31}$$

the total number of block errors E_j made in state S_j is

$$E_j = \sum_{i=j+1}^{J} \lambda_i, \quad j = 1, 2, \ldots, J-1, \tag{32}$$

and it follows that

$$q_j = E_j / T_j, \quad j = 1, 2, \ldots, J-1. \tag{33}$$

To find q_0, we note that the total number of time units T_0 spent in state S_0 equals the total number of correctly decoded blocks, which is given by

$$T_0 = LN - \sum_{i=1}^{J-1} i\lambda_i - \delta_J, \tag{34}$$

the total number of block errors E_0 made in state S_0 is

$$E_0 = \sum_{i=1}^{J} \lambda_i = T_1, \tag{35}$$

and it follows that

$$q_0 = E_0 / T_0. \tag{36}$$

Finally, to compute q_J, we note that the total number of time units T_J spent in state S_J equals δ_J minus the number of block errors that occurred prior to reaching S_J. So,

$$T_J = \delta_J - (J-1)\lambda_J, \tag{37}$$

the total number of block errors E_J made in state S_J[10] is $E_J = T_J - \lambda_J$, and it follows that

$$q_J = E_J / T_J. \tag{38}$$

Example 7: $N = 20{,}000$ frames of length $L = 5000$ were simulated on a binary-input AWGN channel with BPSK modulation for the SC-LDPC code of Fig. 1 with $M = 1000$, $W = 12$, and $E_b/N_0 = 0.9$ dB, which is 0.2 dB below the threshold of the underlying (3,6)-regular LDPC-BC, resulting

[8]Each simulated frame actually contains $L + W$ time units, but only the first L decoded blocks are considered for the burst length distribution. This is done to avoid the decoding window overlapping the termination nodes in the graph, since frame termination is not taken into account in the model.

[9]In the case that there are EOF bursts of length less than J, the burst length distribution data files are modified to count these as finite-length bursts rather than as EOF bursts.

[10]Once in the burst error state S_J, the decoder remains there after each subsequent decoding error and only returns to the random error state S_0 after a finite-length burst error (of length J or greater) or an EOF burst.

in a decoded BLER = 0.4670.[11] The results of the simulation were then used to create a model with $J = 5$, where J was chosen to be the smallest burst length with less than 100 simulated block errors of that length. From the data files, we determined that $\lambda_1 = 7957$, $\lambda_2 = 1762$, $\lambda_3 = 666$, $\lambda_4 = 261$, $\lambda_{\mathrm{FL}} = 178$, $\lambda_{\mathrm{EOF}} = 15,104$, $\delta_{\mathrm{FL}} = 118,157$, and $\delta_{\mathrm{EOF}} = 46,557,890$. Based on these empirical results, the model parameters $q_0 = 4.866 \times 10^{-4}$, $q_1 = 0.6931$, $q_2 = 0.9020$, $q_3 = 0.9589$, $q_4 = 0.9832$, and $q_5 = 0.9997$ can be calculated using (31) - (38). Then, from (12) and (13), the asymptotic average BLER is given by $P_{\mathrm{BL}}^{(\infty)} = 0.4670$, where we note that the exact agreement in this case is due to the fact that the simulated frames were quite long ($L = 5000$).

Now considering the finite length analysis, we can use (12) and (23) - (29) to compute the average number of block errors in each state as $\bar{n}_0 = 1$, $\bar{n}_1 = 0.6931$, $\bar{n}_2 = 0.6252$, $\bar{n}_3 = 0.5995$, $\bar{n}_4 = 0.5894$, and $\bar{n}_5 = 1797.2976$, from which we observe that almost all the block decoding errors occur in state S_5, the burst error state. Finally, it follows from (30) that the average simulated BLER is given by $P_{\mathrm{BL}}^{(5000)} = 0.3602$.[12] □

Finally, we note that the models developed from a single simulation at a given frame length L can be used to estimate BLER performance for different frame lengths, and hence to predict the performance gain of code doping techniques [7], [8], without having to recalculate the model parameters. This follows from the reasonable assumption that the probability of a finite-length error burst that returns the decoder to state S_0 does not depend on the length of the frame being simulated, and hence the model parameters $q_0, q_1, \ldots, q_{J-1}$ are essentially independent of L.[13] Also, the value of q_J can be modified by adjusting the lengths of the simulated error propagation bursts to account for different values of L. With this modification, it is then straightforward to predict the performance of doping by performing the analysis for frame length $L/2$, which corresponds to a single doping point.

Example 7 (Cont.): For frame length $L = 2500$, we left the values of $q_0, q_1, \ldots, q_4$ unchanged and modified the value of q_5 to reflect the fact that the maximum length of an error propagation burst is now only 2500. This results in the slightly modified value $q_5 = 0.9996$.[14] Again using (12) and (23) - (30), we obtain $P_{\mathrm{BL}}^{(2500)} = 0.1781$, a roughly 50% reduction in the estimated BLER compared to $L = 5000$, which reflects the expected performance gain that can be achieved with doping.[15] The simulated BLER in this case is given by $P_{\mathrm{BL}}^{(2500)} = 0.2967$, which also represents a substantial reduction compared to the simulated BLER for $L = 5000$. □

[11] A low SNR value was chosen for the simulation in order to illustrate the problems caused by decoder error propagation, which typically has little effect on performance at high SNRs.

[12] The accuracy of the finite length estimate can be improved by including more states in the model and by assigning a probability distribution to the dwell times in states S_0 and S_J, but at a cost of added complexity.

[13] For the same reason noted in Footnote 1, this statement is not exact.

[14] The small change in the value of q_5 in this case reflects the fact that, for these values of M, W, and E_b/N_0, it is very unlikely that the decoder escapes from S_J before the end of a frame, regardless of the value of L.

[15] The expected "doping gain" depends on the values of M, W, and E_b/N_0. As noted in [7]–[9], doping only improves performance under conditions for which the decoder suffers from error propagation.

IV. Conclusion

We introduced a general model to analyze the BLER performance of SWD of SC-LDPC codes. Specifically, the model accounts for the problem of decoder error propagation, whereby, under low latency operating conditions at SNRs near capacity, block decoding errors introduce correlation into the decoding process that can sometimes trigger an unlimited burst of further errors. We then showed how the model parameters can be estimated from a simulation run at a particular SNR and indicated how the model can be used to predict decoder performance for different values of the frame length L without recomputing the model parameters, which allows us to predict the BLER performance of code doping techniques [7], [8] for mitigating the effect of decoder error propagation.

Acknowledgment

This work was supported in part by the NSFC under Grant 61701368 and by the National Science Foundation under Grant Nos. ECCS-1710920, OIA-1757207, and HRD-1914635.

References

[1] A. J. Felström, and K. Sh. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181-2191, Sep. 1999.

[2] M. Lentmaier, A. Sridharan, D. J. Costello, Jr., and K. Sh. Zigangirov, "Iterative decoding threshold analysis for LDPC convolutional codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 5274-5289, Oct. 2010.

[3] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803-834, Feb. 2011.

[4] D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, "Spatially coupled LDPC codes constructed from protographs," *IEEE Trans. Inf. Theory*, vol. 61, no. 9, pp. 4866-4889, Sep. 2015.

[5] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Coralli, and G. E. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2303-2320, Apr. 2012.

[6] K. Klaiber, S. Cammerer, L. Schmalen, and S. ten Brink, "Avoiding burst-like error patterns in windowed decoding of spatially coupled LDPC codes," in *Proc. IEEE 10th Int. Symp. on Turbo Codes & Iterative Inf. Processing*, Hong Kong, China, Dec. 3-7, 2018, pp. 1-5.

[7] M. Zhu, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "A novel design of spatially coupled LDPC codes for sliding window decoding," in *Proc. IEEE Int. Symp. Information Theory*, Los Angeles, CA, USA, June 21-26, 2020, pp. 473-478.

[8] M. Zhu, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "Decoder error propagation mitigation for spatially coupled LDPC codes," in *Proc. International Symposium on Information Theory and Its Applications*, Kapolei, HI, USA, October 24-27, 2020, pp. 175-179.

[9] R. Sokolovskii, A. G. i Amat, and F. Brannström, "On doped SC-LDPC codes for streaming," *IEEE Commun. Letters*, vol. 25, no. 7, July 2021.

[10] R. Sokolovskii, A. Graell i Amat, and F. Brännström, "Finite-length scaling of spatially coupled LDPC codes under window decoding over the BEC," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 5988-5998, 2020.