This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

Error Propagation Mitigation in Sliding Window Decoding of Spatially Coupled LDPC Codes

Min Zhu, *Member, IEEE,* David G. M. Mitchell, *Senior Member, IEEE,* Michael Lentmaier, *Senior Member, IEEE,* and Daniel J. Costello, Jr., *Life Fellow, IEEE*

This paper is dedicated to memory of Alexander Vardy, a friend, a scholar, and a giant in the field of coding theory.

***Abstract*—In this paper, we investigate the problem of decoder error propagation for spatially coupled low-density parity-check (SC-LDPC) codes with sliding window decoding (SWD). This problem typically manifests itself at signal-to-noise ratios (SNRs) close to capacity under low-latency operating conditions. In this case, infrequent but severe decoder error propagation can sometimes occur. To help understand the error propagation problem in SWD of SC-LDPC codes, a multi-state Markov model is developed to describe decoder behavior and to analyze the error performance of spatially coupled LDPC codes under these conditions. We then present two approaches - check node (CN) doping and variable node (VN) doping - to combating decoder error propagation and improving decoder performance. Next we describe how the performance can be further improved by employing an adaptive approach that depends on the availability of a noiseless binary feedback channel. To illustrate the effectiveness of the doping techniques, we analyze the error performance of CN doping and VN doping using the multi-state decoder model. We then present computer simulation results showing that CN and VN doping significantly improve the performance in the operating range of interest at a cost of a small rate loss and that adaptive doping further improves the performance. We also show that the rate loss is always less than that resulting from encoder termination and can be further reduced by doping only a fraction of the VNs at each doping position in the code graph with only a minor impact on performance. Finally, we show how the encoding problem for VN doping can be greatly simplified by doping only systematic bits, with little or no performance loss.**

***Index Terms*—Code doping, spatially coupled LDPC codes, sliding window decoding, decoder error propagation.**

I. Introduction

SPATIALLY coupled low-density parity-check (SC-LDPC) codes, also known as LDPC convolutional (LDPCC) codes [1], have been shown to have capacity achieving performance over memoryless binary-input symmetric-output channels. Specifically, SC-LDPC codes derived from *regular* LDPC block codes (LDPC-BCs) exhibit *threshold saturation*, i.e., the *suboptimal* belief propagation (BP) iterative decoding threshold of SC-LDPC code ensembles coincides with the *optimal* maximum a posteriori (MAP) threshold of their underlying LDPC-BC ensembles [2]–[5]. Further, *regular* SC-LDPC code ensembles not only have capacity approaching iterative decoding thresholds, but they are asymptotically good, i.e., their minimum distance grows linearly with the frame length [6]. Therefore, SC-LDPC codes combine the best features of both *regular* and *irregular* LDPC-BCs.

An iterative belief propagation sliding window decoding (SWD) algorithm was proposed for SC-LDPC codes in [7] in order to reduce decoding latency, memory, and complexity. Assuming an *additive white Gaussian noise* (AWGN) channel, in order to achieve the best possible performance over a range of *signal-to-noise ratios* (SNRs), Huang et al. showed empirically in [8] that the *decoder window size* W should be at least six times the decoding constraint length. However, in practice, lower latency operation is often desirable, thereby necessitating a smaller window size. In this case, infrequent but severe decoder error propagation can sometimes occur when using SWD, particularly at SNRs below the BP threshold of the underlying LDPC-BC. More specifically, during the SWD process, when a decoding error occurs, the decoding of subsequent symbols can also be affected, and a continuous string of decoding errors can result. This *decoder error propagation* phenomenon can result in unacceptable performance loss in continuous (streaming) transmission or large frame length applications or result in an unacceptable rate loss if frequent encoder termination is employed.

The effect of decoder error propagation on the performance of SWD of SC-LDPC codes was first mentioned in [9], whereas the first detailed study of decoder error propagation in SWD was done for the related class of braided convolutional codes in [10], [11]. For SWD of SC-LDPC codes, Klaiber et al. [12] proposed adapting the number of decoder iterations and/or shifting the window position in order to limit the effects of error propagation, both of which involve altering the decoding procedure. Using a different approach, we proposed *check node (CN) doped SC-LDPC codes* in [13], which employ reduced-degree CNs spaced throughout the coupled graph to help the decoder recover from error propagation. Similar to [12], this also requires altering the decoding procedure

This paper is based upon work supported by the NSFC under Grant 62271380 and National Science Foundation under Grant Nos. CCF-2145917, CNS-2148358, HRD-1914635, and OIA-1757207. This work was presented in part at the International Symposium on Information Theory, Los Angeles, CA, USA, June 2020, in part at the International Symposium on Information Theory and Its Applications, Kapolei, Hawaii, USA, October 2020, in part at the Information Theory Workshop, Riva del Garda, Italy, April 2021, in part at the Global Communications Conference Workshops, Madrid, Spain, December 2021, and in part at the International Symposium on Information Theory, Espoo, Finland, June 2022.

M. Zhu is with the State Key Lab. of ISN, Xidian University, Xi'an 710071, China, (e-mail: zhunanzhumin@gmail.com).

D. G. M. Mitchell is with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, Las Cruces, NM 88003, USA, (e-mail: dgmm@nmsu.edu).

M. Lentmaier is with the Department of Electrical and Information Technology, Lund University, 221 00 Lund, Sweden, (e-mail: michael.lentmaier@eit.lth.se).

D. J. Costello, Jr. is with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA, (e-mail: dcostel1@nd.edu).

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

whenever a doping position is reached. More recently, we proposed *variable node (VN) doped SC-LDPC codes* by fixing the code bits corresponding to certain VNs spaced throughout the coupled graph to a predetermined value [14]. Sokolovskii et al. subsequently studied the finite length scaling behavior of VN doped SC-LDPC codes on the binary erasure channel (BEC) in [15] and showed that doping effectively works by initiating a *decoding wave* at a doping position similar to what is observed at the beginning of an undoped coupled graph.[1] Unlike CN doping and the techniques of [12], VN doping allows recovery from error propagation without altering the decoding procedure, although fixing the value of certain VNs can present encoding challenges. Since the required predetermined distribution of doped positions in CN and VN doping may not completely eliminate error propagation, an adaptive doping strategy for SC-LDPC codes that relies on the availability of a noiseless binary feedback channel was proposed in [16]. Finally, a general model to compute the decoder error rate of SWD of SC-LDPC codes and predict the performance improvement achievable with doping was presented in [17].

Notably, the encoding challenges of doping were not addressed in these papers. With this in mind, we introduced a *systematic doping* procedure to simplify both the process of encoding and the procedure for recovering the decoded information sequence in VN doping [18]. *Systematic doping* employs systematic encoding and only dopes a fraction of the VNs, i.e., only systematic bits, at each doping position. This allows the doping to be done prior to encoding, thus greatly simplifying the encoding process, and also results in a straightforward procedure for recovering the decoded information sequence. This is particularly advantageous in the case of adaptive doping, where these operations must be performed "on the fly".

In this paper, we explain the problem of decoder error propagation in SWD of SC-LDPC codes and propose means of mitigating its effects on decoder error performance. Specifically, the CN doping, VN doping, adaptive doping, and systematic VN doping techniques introduced in [13], [14], [16], [18] to combat error propagation are summarized and presented in a unified framework. These doping techniques are shown to mitigate the effects of decoder error propagation without terminating encoding, thus enabling streaming transmission or large frame length applications. In addition, the rate loss due to doping is shown to be always less than that resulting from encoder termination. Finally, to better understand the error propagation problem in SWD of SC-LDPC codes, the multi-state decoder model introduced in [17] is used to analyze the behavior of decoders that suffer from error propagation. This modeling capability allows one to determine the effect of doping on the performance of different code designs over a broad range of encoder/decoder parameters and channel conditions.

[1]The concept of code doping was first introduced in a different context by ten Brink [20].

II. Review of SC-LDPC codes with SWD decoding

A. SC-LDPC Codes with SWD

The construction of SC-LDPC codes can be viewed as a protograph-based edge-spreading technique [6] in which a sequence of L disjoint $(d_{\rm v}, d_{\rm c})$-regular LDPC-BC protographs are coupled together into a single graphical chain, where $d_{\rm v}(d_{\rm c})$ represents the VN(CN) degree, $L \to \infty$ results in an *unterminated* coupled chain, and finite L results in a *terminated* coupled chain. We demonstrate the approach using (3,6)-regular SC-LDPC codes as an example, but this can be generalized in a straightforward way, as shown in Sec. V.B for (3,9)-regular and (4,6)-regular codes. We begin with the independent (uncoupled) sequence of (3,6)-regular LDPC-BC protographs formed from an $n_{\rm c} \times n_{\rm v} = 1 \times 2$ *base parity-check matrix* $\mathbf{B} = [3, 3]$ shown in Fig. 1(a), where each multi-edge protograph contains $n_{\rm c} = 1$ CN and $n_{\rm v} = 2$ VNs, $d_{\rm v} = 3n_{\rm c} = 3$ and $d_{\rm c} = 3n_{\rm v} = 6$, and the LDPC-BC *design rate* is $R = 1 - \frac{d_{\rm v}}{d_{\rm c}} = 1 - \frac{n_{\rm c}}{n_{\rm v}} = \frac{1}{2}$. Fig. 1(b) shows the resulting unterminated (3,6)-regular SC-LDPC code chain obtained by applying the *edge-spreading* technique to the uncoupled protographs. The edge spreading in this case is defined by a set of *component base matrices* $\mathbf{B}_0 = \mathbf{B}_1 = \mathbf{B}_2 = [1 \;\; 1]$ that must satisfy $\mathbf{B} = \mathbf{B}_0 + \mathbf{B}_1 + \mathbf{B}_2$, where $m = 2$ is referred to as the *coupling width*, and the base parity-check matrix of the coupled chain is illustrated in Fig. 1(c). In general, an arbitrary edge spreading must satisfy

$$\mathbf{B} = \sum\nolimits_{i=0}^{m} \mathbf{B}_i, \tag{1}$$

and termination after L time units results in an additional m CNs at the end of the graph. As a consequence, letting $n'_{\rm c}$ ($n'_{\rm v}$) be the number of CNs (VNs) in the coupled chain, the design rate of the SC-LDPC code of length L is given by $R_L = 1 - \frac{n'_{\rm c}}{n'_{\rm v}} = 1 - \frac{n_{\rm c}(L+m)}{n_{\rm v}L} = 1 - \left(\frac{L+m}{L}\right)(1-R)$. Applying a 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).

Sliding window decoding (SWD) of SC-LDPC codes [7], as opposed to using a standard *flooding decoding schedule* over the entire terminated graph, can be employed to reduce decoding latency, memory, and complexity for long code chains and for *unterminated* streaming (continuous transmission) applications. For example, in Fig. 1(b), the red rectangular box represents a decoding window of *size* $W = 5$ (blocks). For an AWGN channel, SWD consists of ① applying a BP flooding schedule to all the nodes in the window until some *stopping criterion* is met or some maximum number of iterations $I_{\max}$ is reached, ② decoding the block of $n_{\rm v}M$ *target symbols* in the first window position according to the signs of their *log-likelihood ratios* (LLRs), ③ shifting the window one time unit (block) to the right. Decoding continues in the same fashion until the entire chain is decoded, where the *decoding latency* in symbols is given by $n_{\rm v}MW$.

B. Error propagation in SWD

In SWD of SC-LDPC codes, when a block of target symbols at time t is decoded, the window shifts to include the most

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

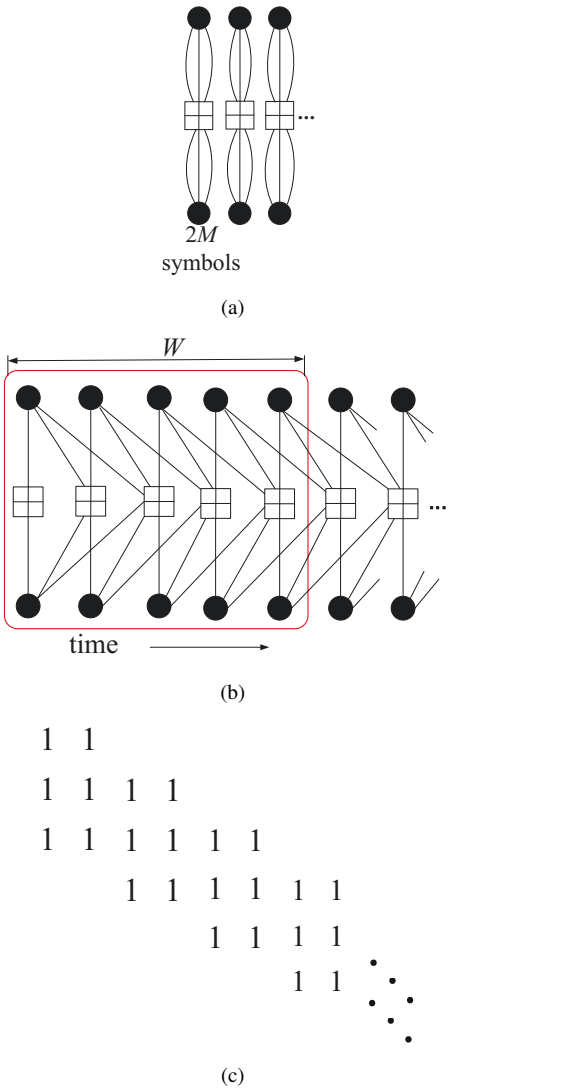


Fig. 1. A (3,6)-regular SC-LDPC code protograph obtained from an underlying (3,6)-regular LDPC-BC protograph. The black circles represent VNs and the "plus" squares represent CNs. The figure illustrates (a) a sequence of independent (uncoupled) LDPC-BC protographs, (b) spreading edges to the $m = 2$ nearest neighbors to form an SC-LDPC protograph, and (c) the resulting coupled parity-check matrix.

recent block of received symbols at time $t+W$, and decoding commences on the block of target symbols at time $t+1$. During the decoding of the time $t+1$ block, the final LLRs of the m past decoded blocks, from time $t-m+1$ to time t, remain involved in the decoding process, although these LLRs are no longer updated, as illustrated in Fig. 2 for the (3, 6)-regular SC-LDPC code example with $W = 3$ and $m = 2$.

Once a block of $n_{\mathrm{v}}M$ target symbols has been decoded, we say that a *block error* has occurred if it contains one or more LLRs with incorrect sign. Typically, if only a few symbols have incorrect LLRs and most of the correct symbols have strong LLRs, the LLRs of the incorrectly decoded block will have only a small effect on the decoding of subsequent blocks, and the decoder will recover and continue to decode correctly. This type of operation typically results in randomly distributed block errors.

However, if an error block contains a large number of incorrect LLRs, particularly if they have large absolute values and many of the LLRs associated with the correctly decoded

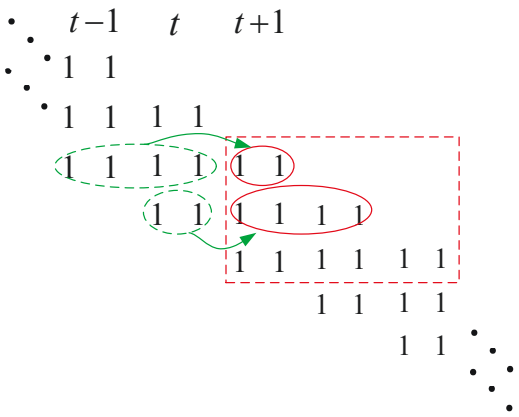


Fig. 2. Final VN LLRs at times $t-1$ and t are used to update the CNs in the window during the decoding of the target symbols at time $t+1$.

symbols have small absolute values, those "unreliable" LLRs can negatively affect the decoding of the next block of target symbols, causing a block error that would not have occurred under normal operating conditions. This, in turn, can trigger additional block errors, resulting in *decoder error propagation*, i.e., a continuous sequence or burst of incorrectly decoded blocks.

In an application where information is transmitted in *frames* of length L, with graph termination following the last block of transmitted VNs, any error propagation will be limited and decoding will start fresh with the next frame. However, if the frame length L is large, a significant number of blocks can be affected by error propagation, thus severely degrading performance. In a streaming application, with no termination, the situation could be catastrophic,[2] resulting in a *block error rate* (BLER) that asymptotically tends to 1.

III. Modeling and Analysis of SWD

Before proceeding with a discussion of methods to circumvent decoder error propagation, it is instructive to consider the difficulty of assessing the extent of the problem using conventional Monte Carlo simulation techniques. Typically, in order to limit the time duration of submitted simulation jobs, decoded BLERs are determined by simulating some number N of frames, each of length L, such that the total number of simulated code symbols $n_{\mathrm{v}}MLN$ is large enough to gather reliable error statistics. Under normal operating conditions, when block errors occur randomly, this process works perfectly well. But when the decoder experiences error propagation, the assumption of randomly distributed block errors is no longer valid, and BLER statistics can be severely affected by the particular combination of L and N chosen for the simulation.

We now give an example illustrating the effect of decoder error propagation on simulated BLER statistics, assuming an AWGN channel with binary phase-shift-keyed (BPSK)

[2]This behavior is analogous to the well-known phenomenon of catastrophic error propagation in classical convolutional codes. However, in that case, the problem is caused by poor encoder design, whereas here it is due to the decoder design.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

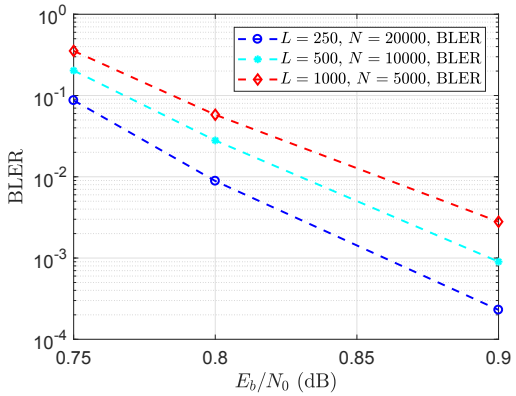


Fig. 3. SWD BLER performance of a (3,6)-regular SC-LDPC code for three different combinations of frame length and number of frames simulated, all with the same total number of simulated blocks.

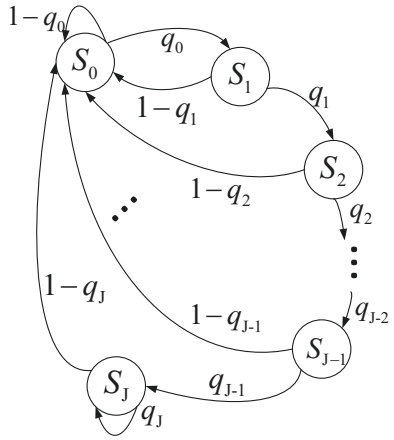


Fig. 4. The state transition diagram of a general decoder model for SWD.

signaling. The simulated BLER performance of SWD of a (3,6)-regular SC-LDPC code based on the coupled protograph in Fig. 1 is shown in Fig. 3, where $W = 12$ and $M = 2000$. The figure represents the simulation of a total of $LN = 5\times10^6$ blocks (or $n_{\rm v}MLN = 2 \times 10^{10}$ bits) for three different combinations of L and N. From the figure, we observe that, with increasing L, the BLER performance becomes worse, even though there are relatively few error propagation frames overall, thus confirming the above observation.[3] Indeed, the worst scenario occurs when a single long frame is used to gather error statistics. In this case, once error propagation begins, it continues all the way to the end of the simulation job, causing the BLER to grow without bound.

Based on the fact that SC-LDPC codes have better BP decoding thresholds than their underlying LDPC-BCs, it is desirable to operate at values of E_b/N_0 below the LDPC-BC threshold, while keeping the window size W (latency) as small as possible. From the above example, however, we see that decoder error propagation can present a significant challenge to the reliable operation of SWD under such operating conditions. Thus, in order to understand the behavior of SWD under these conditions and assess the need for methods of mitigating decoder error propagation, we now introduce a general *Markov decoder model* for SWD. The model characterizes the decoded BLER as a function of the extent to which unreliable LLRs from past decoding decisions can negatively affect future decoding decisions. It 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 state transition diagram of the decoder model is shown in Fig. 4, 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_i denote the *block error rate* (BLER) in state S_i, $i = 0, 1, 2, \ldots, J$.

Referring to Fig. 4, 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 connected to the window (see Fig. 2) 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, we have that $q_{J-1} \geq q_{J-2} \geq \ldots \geq q_1 \geq q_0$, 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 $q_J \geq q_{J-1}$ as long as decoding errors continue, returning to state S_0 with probability $1 - q_J$, resulting in a burst error of length J or more.[4] If the influence of the m previously decoded incorrect blocks is strong enough, such that $q_J \to 1$, 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, the *channel parameter* E_b/N_0, the *decoder parameter* W, and the *code parameter* M will all influence the probabilities $q_0, q_1, \ldots, q_J$. 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 and decoders (large M and W) have smaller values of q_0 and thus are less likely to reach state S_J, but once they enter the burst state they have a high probability of staying there, i.e., a high probability of unlimited decoder error

[3]Fig. 3 represents only a narrow range of E_b/N_0 values, below the BP decoding threshold of the underlying (3,6)-regular LDPC-BC ensemble, where error propagation presents a significant problem. For larger values of E_b/N_0 and/or W, SWD typically recovers from error propagation without terminating the frame.

[4]We note here that, when the decoder returns to state S_0 after one or more blocks errors, there are up to $m - 1$ 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, so we choose to ignore it in the model.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

propagation. Typically, they suffer only a very few short burst errors, and their BLER performance is therefore dominated by error propagation. Weak codes and decoders (small M and W), 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.[5]

We now proceed to derive expressions for the decoded BLER, as functions of $q_0, q_1, \ldots, q_J$, of SC-LDPC codes based on this model, for unterminated ($L \to \infty$) transmission.[6] The derivation of the expressions for the BLER for terminated (finite L) transmission is given in Appendix A, and a procedure for estimating the model parameters and predicting the BLER performance from a single simulation run is presented in Appendix B.

Case I: No intermediate states ($J = 1$), $L \to \infty$.[7]

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

$$p_1 = \frac{p_0 - p_0 (1 - q_0)}{1 - q_1} = \frac{p_0 q_0}{1 - q_1}. \tag{2}$$

Now the *average BLER* can be written as

$$\begin{aligned} P_{\mathrm{BL}}^{(\infty)} &= p_0 q_0 + p_1 q_1 = p_0 q_0 + p_0 q_1 \left(\frac{q_0}{1 - q_1} \right) \\ &= p_0 \left(\frac{q_0}{1 - q_1} \right). \end{aligned} \tag{3}$$

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

$$p_0 = \frac{1 - q_1}{1 - q_1 + q_0}, \tag{4}$$

and using (4) in (3) it follows that

$$P_{\mathrm{BL}}^{(\infty)} = \frac{q_0}{1 - q_1 + q_0} = \frac{r_1}{1 - q_1 + r_1}, \tag{5}$$

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 $q_0 = 0.01, q_1 = 0.99$.[8] Then $P_{\mathrm{BL}}^{(\infty)} = 0.5$. □

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

In this case, we have

$$\begin{cases} p_0 = p_0 (1 - q_0) + p_1 (1 - q_1) + p_2 (1 - q_2) \\ p_1 = p_0 q_0, \quad p_2 = p_1 q_1 + p_2 q_2, \end{cases} \tag{6}$$

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

$$p_0 = \frac{1}{1 + q_0 + \frac{q_0 q_1}{1-q_2}} = \frac{1 - q_2}{1 - q_2 + q_0 - q_0 q_2 + q_0 q_1}. \tag{7}$$

Now the average BLER can be written as

$$\begin{aligned} P_{\mathrm{BL}}^{(\infty)} &= p_0 q_0 + p_1 q_1 + p_2 q_2 \\ &= \frac{q_0 - q_0 q_2 + q_0 q_1}{1 - q_2 + q_0 - q_0 q_2 + q_0 q_1} = \frac{r_2}{1 - q_2 + r_2}, \end{aligned} \tag{8}$$

where $r_2 \triangleq q_0 (1 - q_2 + q_1)$. Again we see that $\lim\limits_{q_2 \to 1} P_{\mathrm{BL}}^{(\infty)} = 1$ (unlimited error propagation).

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

Case III: More than two intermediate states ($J \geq 3$), $L \to \infty$.

Starting with $J = 3$, we have

$$\begin{cases} p_0 = p_0 (1 - q_0) + p_1 (1 - q_1) + p_2 (1 - q_2) + p_3 (1 - q_3), \\ p_1 = p_0 q_0, \quad p_2 = p_1 q_1, \\ p_3 = p_2 q_2 + p_3 q_3 = p_1 q_1 q_2 + p_3 q_3 = p_0 q_0 q_1 q_2 + p_3 q_3, \end{cases} \tag{9}$$

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

$$\begin{aligned} p_0 &= \frac{1}{1 + q_0 + q_0 q_1 + \frac{q_0 q_1 q_2}{1-q_3}} \\ &= \frac{1 - q_3}{1 - q_3 + q_0 - q_0 q_3 + q_0 q_1 - q_0 q_1 q_3 + q_0 q_1 q_2}. \end{aligned} \tag{10}$$

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

$$\begin{aligned} P_{\mathrm{BL}}^{(\infty)} &= p_0 q_0 + p_1 q_1 + p_2 q_2 + p_3 q_3 \\ &= p_0 q_0 + p_0 q_0 q_1 + p_0 q_0 q_1 q_2 + p_0 \frac{q_0 q_1 q_2 q_3}{1 - q_3} \\ &= p_0 \frac{r_3}{1 - q_3} = \frac{r_3}{1 - q_3 + r_3}, \end{aligned} \tag{11}$$

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 $P_{\mathrm{BL}}^{(\infty)} = 0.834$.[9] □

For $J > 3$,

$$r_J \triangleq q_0 \Big\{ 1 - q_J + q_1 \big\{ 1 - q_J + q_2 \left[1 - q_J + \cdots + q_{J-2} (1 - q_J + q_{J-1}) \right] \big\} \Big\}$$

, and the general expression for the average BLER is given by

$$P_{\mathrm{BL}}^{(\infty)} = \frac{r_J}{1 - q_J + r_J}. \tag{12}$$

IV. Code Doping

In order to mitigate the decoder error propagation problem in SWD, we present three code doping techniques: (i) *CN doping* based on occasionally inserting additional CNs into the protograph of a regular SC-LDPC code, (ii) *VN doping* based on occasionally inserting known VNs into a coupled protograph, and (iii) *adaptive doping* based on inserting doping positions into a coupled chain in response to requests from the decoder transmitted over a noiseless binary feedback channel. In each case, the resulting structured irregularity

[5]The relative strength of a code increases with m as well as with M, while decoder strength increases with both W and $I_{\max}$.

[6]This is equivalent to deriving the steady state probabilities in a Markov process.

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

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

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

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

slightly reduces the code rate but has the beneficial effect of partitioning a long or continuous graph into shorter "sections", thus limiting the effects of error propagation. To illustrate the doping process, we again consider the (3,6)-regular SC protograph of Fig. 1(b) as an example, while noting that the design can be applied in the same way to any $(d_{\rm v}, d_{\rm c})$-regular SC protograph.

A. Construction of a CN Doped Protograph

For the (3,6)-regular SC protograph example, all the VNs have degree 3, while all the CNs, except the ones at the graph boundaries, have degree 6, where we note that the reliable information generated by the reduced-degree CNs at the boundaries is responsible for the threshold saturation property of SC-LDPC codes [2]–[5]. Motivated by this fact and our desire to limit the effects of decoder error propagation in long frames, the CN doped code design introduces similar reduced-degree CNs at different positions in the coupled chain, as shown in Fig. 5, where the VNs at time $t = \tau_1$ (colored red) spread their three edges to the CNs at times $t = \tau_1 + 1$, $t = \tau_1 + 2$, and $t = \tau_1 + 3$; the red VNs at time $t = \tau_2$ spread their three edges to the CNs at times $t = \tau_2 + 2$, $t = \tau_2 + 3$, and $t = \tau_2 + 4$, and so on, i.e., if the red VNs at time $t = \tau_j$ are chosen as the jth *doping position*, their edges are connected to the CNs at times $\tau_j + j$, $\tau_j + j + 1$, and $\tau_j + j + 2$. Additionally, the VNs between doping positions (colored black) are coupled in the same way as the preceding red VN pair. This construction has the effect of inserting an additional CN at each doping position, and as a result three degree 4 CNs (colored green) are generated. At a cost of a slight rate loss, these reduced degree CNs at the doping positions result in stronger "local" codes compared to the standard (3,6)-regular design, which facilitates the ability of a SWD to truncate decoder error propagation.[10] In the special case where the doping positions are equally spaced in the coupled chain, i.e., $\tau_k = \tau_1 + (k-1)\,s$, the chain is said to be *periodically doped* with *doping period* s, and we refer to this as *periodic CN doping*.[11]

To decode a CN doped (3,6)-regular SC-LDPC code, SWD can be applied to the doped chain, with a slight difference in the way the window shifts compared to standard (3,6)-regular SWD. The window-shifting schedule for CN doped (3,6)-regular SC-LDPC codes is illustrated in Fig. 5. For a window of size W, the block of $2M$ symbols at the leftmost position in the window represents the target symbols. BP decoding is performed within the window, either using a fixed number of iterations or a stopping rule. After a block of target symbols is decoded, the window shifts by one time unit. When a doping position (red VN pair) becomes the target block, the window shifts by one VN time unit to include one new block of VNs, as before, but it shifts by two CN time units to include two new blocks of CNs (and thus still including the same total number of CNs).[12] After the red target symbols are decoded, the window again shifts by one time unit until the next doping position is reached, and the same process is repeated. Generally, for the VN block of (red) target symbols at doping position τ_j, the decoding window covers the VNs from times τ_j to $\tau_j + W - 1$ and the CNs from time $\tau_j + j$ to $\tau_j + j + W - 1$.

B. Construction of a VN Doped Protograph

Although CN doping is very effective at reducing decoder error propagation, as we will see later in this section, it requires alterations to the decoding procedure, as illustrated in Fig. 5. VN doping is based on occasionally inserting known (or fixed) VNs into a coupled chain protograph. Similar to CN doping, this facilitates the truncation of any error propagation in the iterative decoding process, again at a cost of a slight rate loss.

Fig. 6 shows a general VN doping scheme for a (3,6)-regular SC-LDPC code, where each time unit again represents a *block* of $2M$ coded symbols. The VNs at time $t = \tau_1$ (the green empty circles) are doped by setting the $2M$ coded bits corresponding to these VNs to be "0". As a result, the CNs at times $t = \tau_1, \tau_1 + 1$, and $\tau_1 + 2$ (colored red and shaded) can be viewed as degree 4, rather than degree 6, CNs, thus emulating CN doping without actually altering the graph structure. Similarly, if the VNs at time $t = \tau_2$ are doped, the CNs at times $t = \tau_2, \tau_2 + 1$, and $\tau_2 + 2$ (colored red and shaded) can be viewed as degree-4 CNs.[13] Similar to CN doping, inserting doped VNs periodically into the coupled chain, i.e., *periodic VN doping*, results in the doping positions being equally spaced in the coupled chain. In this case, the doping positions (the green VNs) at times $t = \tau_1$, τ_2, and τ_3 shown in Fig. 6 are fixed to known values and spaced s time units apart, i.e., $\tau_k = \tau_1 + (k-1)s$.

The decoding process is the same as for undoped codes, except that the code symbols at the doping positions are treated as known, i.e., during the decoding process we set the LLRs of the doped symbols to a large constant positive value Γ. These known symbols have the effect of transmitting perfectly reliable information to their neighbour nodes, thus helping the decoder recover from error propagation. An important implementation advantage of VN doping compared to CN doping is that the shape of the decoding window remains unaltered, so the decoding procedure is essentially the same as for undoped codes.

C. Adaptive Doping

A modification of the doping process, called *adaptive doping*, can be applied to both CN and VN doping to further improve decoder performance when a noiseless binary feedback channel is available. For simplicity, we assume that the feedback is instantaneous, although it is not difficult to incorporate some fixed feedback delay into the analysis.

[10] The CN doping technique described here can also be viewed as a form of code extension.

[11] Whenever the doping positions must be fixed in advance, periodic doping gives the best performance (see Sec. IV.D).

[12] We note here that the time scales differ for VNs and CNs, since CN doping interrupts the regular pattern of exactly one CN for every two VNs, i.e., at every doping position there is an extra CN corresponding to the pair of VNs.

[13] The VN doping technique described here can be viewed as a form of code shortening.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

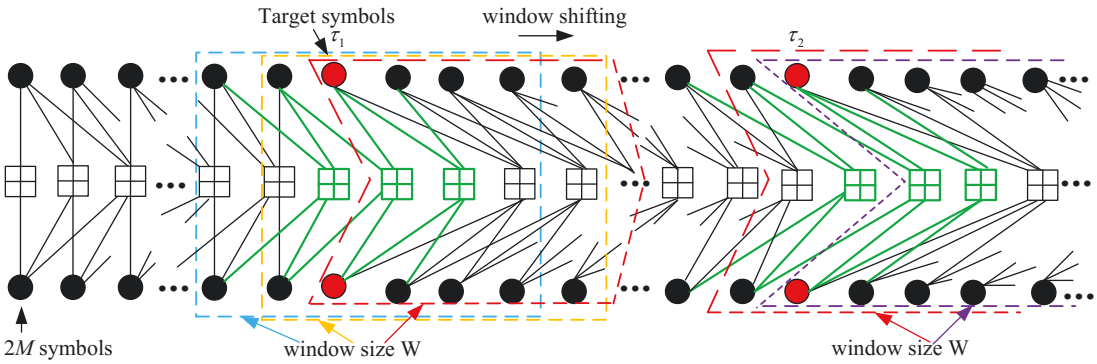


Fig. 5. CN doping for a (3,6)-regular SC-LDPC code with occasional CNs of reduced degree spaced throughout the chain.

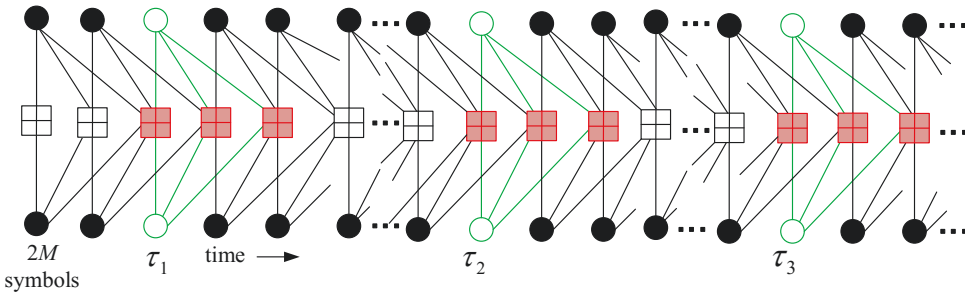


Fig. 6. VN doping for a (3,6)-regular SC-LDPC code with occasional fixed VNs spaced throughout the coupled chain.

In contrast to periodic doping, adaptive doping inserts doping positions into the coupled chain on an "as needed" basis, depending on the average LLR magnitudes in some number of recently decoded target blocks. In this case, (typically unequally spaced) doping positions at times $t = \tau_1, \tau_2, \tau_3, \ldots$ are inserted into the coupling chain in response to requests from the decoder. To trigger a doping request in the SWD process, after completing all the iterations necessary to decode the target block at time t, if the *average decoded LLR magnitude* $\mathcal{L}_t$ satisfies

$$\mathcal{L}_t \triangleq \frac{1}{2M} \sum_{i=0}^{2M-1} \left|\mathrm{LLR}_i^t\right| \leq \eta, \tag{13}$$

where $\left|\mathrm{LLR}_i^t\right|$ is the LLR of the ith VN at time t, $i = 0, 1, \ldots, 2M-1$, and η is some pre-determined *threshold*, we consider the target block at time t as *failed*. If we experience some preset number N_r of consecutive failed target blocks, a doping request is submitted and the next block of VNs entering the far end of the window is assumed to be doped.

D. BLER Analysis of Doping

We now make use of the analysis of the BLER of SWD of SC-LDPC codes presented in Sec. III to illustrate the advantages of doping. Let $\lambda_j, j \in [0, 1, \ldots, l-1]$ denote the number of blocks in each section of the graph (between doping positions), where l represents the number of sections in a frame. Referring to Fig. 5, for a frame of length L, it follows that $\sum_{j=0}^{l-1} \lambda_j = L$, where $\lambda_j = \tau_{j+1} - \tau_j$, $\tau_0 \triangleq 0$, and $\tau_l \triangleq L$. We now make the assumption that, if the decoder is in the burst error state S_{J} at the end of the ith section, it will leave state S_{J}, i.e., error propagation will be truncated, at the end of that section.[14] This allows us to apply the analysis of Sec. III independently to each section of length λ_j, $j \in [0, 1, \ldots, l-1]$, and we can write the overall average BLER as

$$P_{\mathrm{BL,doped}} = \frac{\sum_{j=0}^{l-1} \lambda_j \cdot P_{\mathrm{BL},\lambda_j}}{L}, \tag{14}$$

where $P_{\mathrm{BL},\lambda_j}$ represents the BLER of the jth section.

To determine the best doping points for a fixed number of sections l, (14) can in general be treated as an optimization problem with respect to the λ_j parameters. But, under error propagation conditions, we see from Fig. 3 that, for the same channel and code parameters, $P_{\mathrm{BL},\lambda_j}$ is an increasing function of λ_j, which implies that sections of equal length will minimize $P_{\mathrm{BL,doped}}$. Hence we consider only the special case of periodic doping, where the doping positions are equally spaced in the coupled chain (i.e., $\lambda_j = s$, $j \in [0, 1, \ldots, l-1]$), in which case (14) can be written as

$$P_{\mathrm{BL,doped}} = P_{\mathrm{BL},\lambda_j}|_{\lambda_j = s}, \quad j = 0, 1, \ldots, l-1, \tag{15}$$

where s is the doping period.

Before presenting examples of how to apply the analysis of Sec. III to doping, we distinguish two cases: ① a "strong code/decoder" case in which the protograph lifting factor M and the decoder window size W are both large, and ② a

[14]This assumption is supported by extensive simulations showing that doping is effective in limiting decoder error propagation, as will be demonstrated by the numerical results presented in Sec. IV.F.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

"weak code/decoder" case in which M and W are both small. As noted in Sec. III, the analysis model can be assumed to have $q_0 \ll 1$ and $q_J \approx 1$ for strong codes, while q_0 is larger and q_J is smaller for weak codes. In other words, strong codes have very low block error rates in state S_0 and a very small probability of reaching state S_J, but once they reach state S_J they typically experience unlimited error propagation. For this reason, large frame lengths can suffer significant performance degradation in this case. For weak codes, on the other hand, the block error rate is much higher in state S_0, making them unsuitable for capacity-approaching applications, but the decoder can sometimes recover from error propagation, so large frame lengths are not necessarily catastrophic.

The doping methods presented in this section typically result in much smaller values of q_J, thus allowing the decoder to escape more quickly from state S_J and significantly improving the decoded block error rate P_{BL}, particularly at SNR operating points near capacity. For periodic doping, the transition probability from state S_J to state S_0 can be approximated as $1-q_J = 2/s$, where we make the reasonable assumption that, on average, we enter state S_J in the middle of a doping period, i.e., $s/2$ time units from the next doping point. For adaptive doping, the transition probability from state S_J to state S_0 can be approximated as $1 - q_J = 1/(W + N_r - 1)$, since, once the decoder enters state S_J, it takes $W + N_r - 1$ time units before a retransmission requesting a doping position is sent over the feedback channel.[15]

We now calculate P_{BL}, using the asymptotic analysis model, for both strong code and weak code examples, where doping is accounted for by adjusting the value of q_J, as discussed above. Based on these calculations, we then compare the predicted P_{BL} performance of undoped, periodically doped, and adaptively doped codes. For simplicity, we choose the case $J = 2$, i.e., a 3-state model, but the same conclusions hold for any value of J.

Example 5 (strong code): Choose $q_0 = 0.01$, $q_1 = 0.1$, $q_2 = 0.999$.

For the undoped case ($1 - q_2 = 0.001$), the BLER is calculated using (8) as $P_{\mathrm{BL}}^{\mathrm{undoped}} = 0.5025$. For periodic doping ($s = 200$, $1 - q_2 = 2/s = 0.01$), $P_{\mathrm{BL}}^{\mathrm{periodic}} = 0.0991$, and for adaptive doping ($W = 18$, $N_r = 2$, $1 - q_2 = 0.0526$), $P_{\mathrm{BL}}^{\mathrm{adaptive}} = 0.0282$. □

For the chosen parameters, we see that both periodic and adaptive doping significantly reduce P_{BL} compared to the undoped case, which suffers from unlimited error propagation, with adaptive doping performing significantly better than periodic doping.

Remark: Here we chose $q_2 = 0.999$ since strong codes almost never recover from decoder error propagation, $s = 200$ since this limits the burst length to 100 on average, $W = 18 = 6(m+1)$ (a strong decoder - six times the decoding constraint length) since this gives near optimal SWD performance at moderate-to-high SNRs, and $N_r = 2$, since anything more than single isolated errors typically indicates decoder error propagation has begun.

[15]Here the time it takes to reach a doping position is fixed.

Example 6 (weak code): Choose $q_0 = 0.02$, $q_1 = 0.2$, $q_2 = 0.99$.

For the undoped case ($1 - q_2 = 0.01$), the BLER is calculated using (8) as $P_{\mathrm{BL}}^{\mathrm{undoped}} = 0.2968$. For periodic doping ($s = 100$, $1 - q_2 = 2/s = 0.02$), $P_{\mathrm{BL}}^{\mathrm{periodic}} = 0.1803$, and for adaptive doping ($W = 12$, $N_r = 4$, $1 - q_2 = 0.066$), $P_{\mathrm{BL}}^{\mathrm{adaptive}} = 0.0746$. □

For the chosen parameters, we again see that both periodic and adaptive doping significantly reduce P_{BL} compared to the undoped case, with adaptive doping performing much better than periodic doping.

Remark: Here we chose $q_2 = 0.99$ to reflect the fact that weak codes are less likely than strong codes to suffer from unlimited decoder error propagation, $s = 100$ since weak codes are more likely to reach state S_J and thus need more frequent doping, $W = 12 = 4(m + 1)$ (a weaker decoder than in Example 5), and $N_r = 4$ to reflect the fact that, for weak codes, we must wait longer before declaring that error propagation has begun and sending a doping request.

In general, either with or without doping, strong codes perform much better than weak codes due to the smaller value of q_0, since strong codes are much more resilient to channel errors. Also, for smaller values of L, the gains achieved by doping are expected to be less dramatic, since error propagation is not as damaging for small frame lengths. Finally, we note that the finite L analysis of Appendix A can also be used for periodic doping to draw similar conclusions by choosing $L = s$.

E. Rate Loss

The *design rate* of CN doped SC-LDPC codes with frame length L and d doping positions is given by

$$\begin{aligned} R_L^{\mathrm{CN}} &= 1 - \frac{n'_{\mathrm{c}}}{n'_{\mathrm{v}}} = 1 - \frac{(L+m+d)\,n_{\mathrm{c}}}{L n_{\mathrm{v}}} \\ &= 1 - \left(\frac{L+m+d}{L}\right)(1-R)\,, \end{aligned} \tag{16}$$

where $R = 1 - n_{\mathrm{c}}/n_{\mathrm{v}}$ is the design rate of the uncoupled LDPC-BC protograph [6], $d = L/s$ is fixed in the periodic case, and d is variable, depending on the frequency with which the threshold test of (15) fails N_r consecutive times, in the adaptive case.

Compared to the design rate $R_L = 1 - \left(\frac{L+m}{L}\right)(1-R)$ of undoped SC-LDPC codes [6], we see from (16) that the design rate R_L^{CN} of CN doped SC-LDPC codes is smaller, i.e., CN doping results in some *rate loss*. However, we note that *encoder termination* at the doping positions, which also truncates error propagation, would result in a larger rate loss. Below, the design rates of periodic CN doping for different values of d are calculated.

Example 7: Consider the (3,6)-regular SC-LDPC codes of Fig. 1(b) with frame length $L = 1000$.
Case 1: $d = 0$, $R_{1000} = 1 - \left(\frac{1000+2}{1000}\right)(1-0.5) = 0.499$.
Case 2: $d = 1$, $R_{1000}^{\mathrm{CN}} = 1 - \left(\frac{1000+2+1}{1000}\right)(1-0.5) = 0.4985 > R_{500} = 1 - \left(\frac{500+2}{500}\right)(1-0.5) = 0.498$ for termination.
Case 3: $d = 3$, $R_{1000}^{\mathrm{CN}} = 1 - \left(\frac{1000+2+3}{1000}\right)(1-0.5) = 0.4975 > R_{250} = 1 - \left(\frac{250+2}{250}\right)(1-0.5) = 0.496$ for termination.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

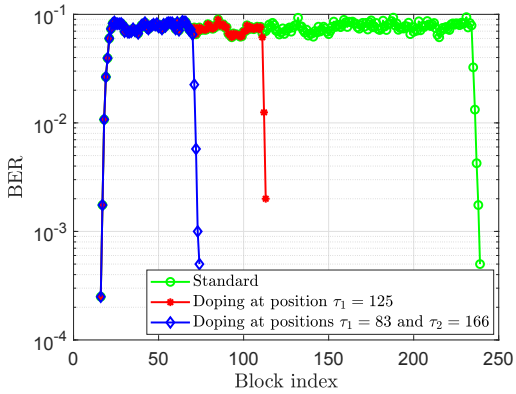


Fig. 7. BER distribution per block of CN doped and undoped (3,6)-regular SC-LDPC codes.

From these results, we note that, even though CN doping results in some rate loss, the rate loss of termination is greater. □

The *design rate* of VN doped SC-LDPC codes with frame length L and d doping positions is given by

$$R_L^{\mathrm{VN}} = 1 - \frac{n'_{\mathrm{c}}}{n'_{\mathrm{v}}} = 1 - \frac{(L+m)\,n_{\mathrm{c}}}{(L-d)\,n_{\mathrm{v}}} = 1 - [(L+m)/(L-d)]\,(1-R)\,. \tag{17}$$

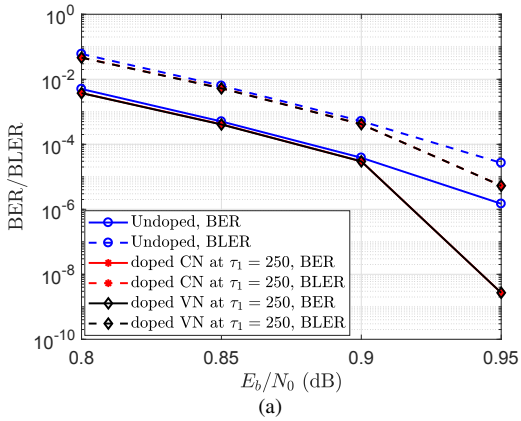
Similar to the CN doping case, VN doping results in some rate loss, but the rate loss of termination is greater. Also, since in general $(L+m)\,/\,(L-d) > (L+m+d)\,/L$, the rate loss of VN doping is always greater than the rate loss of CN doping, but the difference is very slight for large values of L. □

From this analysis and the results of Examples of 5, 6, and 7, we see that doped SC-LDPC codes have improved BLER performance compared to undoped SC-LDPC codes, at a cost of a slight rate loss due to the doping, and that adding more doping positions further improves the BLER performance, with some additional rate loss. Also, we note that the ability of SC-LDPC codes to interrupt decoder error propagation and improve BLER performance is achieved without having to frequently terminate the code graph into short frames.

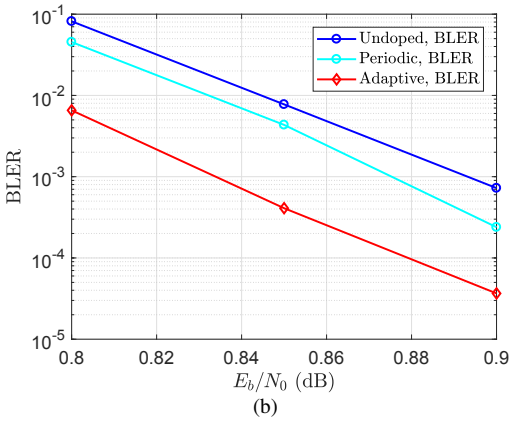
F. Numerical Results

In order to verify the effectiveness of code doping, the *bit error rate* (BER) distribution per block of a typical frame subject to error propagation in SWD of both CN doped and undoped (3,6)-regular SC-LDPC codes is plotted in Fig. 7, where $M = 2000$, $W = 18$, $I_{\max} = 50$, and $L = 250$. Two examples of CN doping are included, one with a single doping position at time $\tau_1 = 125$, and one with two doping positions at $\tau_1 = 83$ and $\tau_2 = 166$. From the figure, we can see that CN doping effectively truncates the error propagation and that adding more doped CNs truncates the error propagation earlier.

Next, for $M = 2000$, $W = 12$, $I_{\max} = 50$, and $L = 500$, Fig. 8(a) shows the BER and BLER performance comparison between an undoped (3,6)-regular SC-LDPC code with rate $R_{500} = 0.498$ and CN and VN doped (3,6)-regular SC-LDPC codes with a single doping position at time $\tau_1 = 250$ and rate



(a)



(b)

Fig. 8. Performance comparison of (a) CN doped, VN doped, and undoped and (b) undoped, periodically VN doped, and adaptively VN doped (3,6)-regular SC-LDPC codes.

$R_{500}^{\mathrm{CN}} = R_{500}^{\mathrm{VN}} = 0.497$.[16] We note that the doped codes gain more than two orders of magnitude in BER and nearly one order of magnitude in BLER compared to the undoped code at SNR operating points of interest (below the threshold of the underlying LDPC-BC).[17] Also, the fact that the performance of both doping methods is essentially equivalent corroborates our earlier observation that VN doping emulates the CN doping process while not requiring any alteration to the shape of the decoding window.

In order to verify the effectiveness of adaptive doping, the BLER performance of the undoped, periodically VN doped, and adaptively VN doped (3,6)-regular SC-LDPC codes of Fig. 1(b) with SWD is shown in Fig. 8(b) for $M = 2000$, $W = 12$, $N_r = 2$, $I_{\max} = 50$, $L = 1000$, 2 doping positions per frame for periodic doping, and a maximum of 2 doping positions per frame for adaptive doping.[18] The results confirm our analysis above that, when low latency operation is desired at the lower SNRs typically used in practice, code doping significantly improves the BLER performance, with adaptive doping outperforming periodic doping.

[16] In the case of VN doping, we set the LLRs of the doped symbols to $\Gamma = +10$.

[17] The performance gain affects only a limited range of SNRs, since doping makes little difference for either "very good" or "very bad" channels.

[18] We limit the number of doping points for adaptive doping so that its rate loss can never exceed that of periodic doping.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

V. Systematic VN Doping

In this section, we introduce the notion of *fractional VN doping*, in which only a portion of the VNs at a doping position are affected. Then we consider systematic encoding and only doping a fraction of the VNs, i.e., the systematic bits, at each doping position. This *systematic VN doping* allows the doping to be done prior to encoding, thus simplifying the encoding process, and it also results in a straightforward procedure for recovering the decoded information sequence. This is particularly advantageous in the case of adaptive doping, where these operations must be performed "on the fly".

A. Fractional VN Doping

Fractional VN doping is illustrated in Fig. 9 for the case of (3,6)-regular SC-LDPC codes, where the slashed circles represent the fractionally doped nodes and the solid circles represent undoped nodes.[19] At each time unit, the two protograph nodes represent a total of $2M$ symbols. Let $0 \leq \delta \leq 1$ represent the *doping fraction*, i.e., the fraction of doped bits at each doping position. Then, for example, at time τ_1, $2\delta M$ bits are doped, where we note that $\delta = 0$ corresponds to no doping and $\delta = 1$ corresponds to full doping. We classify the $2M$ symbols at each time unit into two sets: a doped set $\mathcal{D}$ and an undoped set $\bar{\mathcal{D}}$.

Now consider SWD of SC-LDPC codes. In the case of fractional doping, let L_i^t, $1 \leq i \leq 2M$, denote the channel LLR used for decoding the ith bit at time unit t. Then we have

$$L_i^t = \begin{cases} \Gamma, & i \in \mathcal{D} \\ L_i^{t,\mathrm{ch}}, & i \in \bar{\mathcal{D}} \end{cases}, \tag{18}$$

where $L_i^{t,\mathrm{ch}}$ denotes the received channel LLR of the ith bit at time unit t and, as in Sec. IV.F, $\Gamma = +10$ is chosen to denote the known LLR value corresponding to a doped bit 0. Note that (18) has the effect of assigning certainty to the doped bits during the decoding process.

In the case of fractional doping, we can choose which bits at a protograph node to dope and which to leave undoped. Two fractional doping patterns, *adjacent* doping and *periodic* doping, are illustrated in Fig. 10 for doping fraction $\delta = 0.5$, where the white circles represent doped bits and the black circles represent undoped bits. As can be seen from Fig. 10, in adjacent doping $2\delta M$ consecutive bits are doped, whereas in periodic doping, the $2\delta M$ doped bits are spaced uniformly across the $2M$ bits at a time unit. At a doping position, in the case of $\delta = 0.5$, we see that adjacent doping is equivalent to doping all the VNs in one protograph node and no VNs in the other, whereas periodic doping spreads the doped VNs evenly over both protograph nodes.

It is also possible to spread fractional doping over a *doping span* of σ consecutive positions, where σ is a positive integer, such that the equivalent of $\sigma\delta$ positions is fully doped. The design rate of fractionally VN doped SC-LDPC codes with frame length L is given by

$$R_L^{\mathrm{FR}} = 1 - \frac{n'_{\mathrm{c}}}{n'_{\mathrm{v}}} = 1 - \left(\frac{L+m}{L-\sigma\delta}\right)(1-R). \tag{19}$$

B. Systematic VN Doping

As noted above, VN doping involves fixing certain bits in the encoded sequence to have known values. This implies that an encoder for VN doped SC-LDPC codes must be designed to ensure that the value of the encoded bits in the doped positions remains constant for all possible information sequences. This requirement has the effect that certain information sequences are invalid inputs to the encoder, thus resulting in rate loss.

In order to see this, consider an example of a general (nonsystematic) convolutional encoder in which the length K information sequence $\mathbf{u} = (u_0, u_1, \ldots, u_{K-1})$ produces the length N encoded sequence $\mathbf{v} = (v_0, v_1, \ldots, v_{N-1})$ and a particular encoded bit, say v_j, must be a "0". Since every encoded bit is the sum of some subset of information bits, v_j can be expressed as $v_j = 0 = u_{j1} + u_{j2} + \cdots + u_{jk}$, where the indices $j_1, j_2, \ldots, j_k \in \{0, 1, \ldots, K-1\}$ represent the subset of k, $0 < k \leq K$, information bits that contributes to v_j. It follows that changing the value of any one of the bits in the subset, while leaving the others unchanged, will change the value of v_j from "0" to "1", which implies that all information sequences that contain this particular subset of information bits are invalid. Moreover, all information sequences containing any combination of these bits that gives odd parity are also invalid. As a consequence, not all 2^K possible information sequences are valid when code doping is used, which leads to rate loss.

Based on the above discussion, we conclude that designing a non-systematic encoder with doped code bits (or a systematic encoder with doped parity bits) in general leads to a highly complex encoding process. Moreover, in the decoding of LDPC codes, the decoding process results in an estimated code sequence, which must then be inverted according to the same highly complex encoding rule in order to recover the estimated information sequence. Furthermore, in the case of adaptive VN doping [16], these complex encoding and encoder inverse operations must be done "on the fly", whenever the feedback channel requests the insertion of doped bits into the encoded sequence. This difficulty motivates us to restrict our attention to systematic encoding rules and to limit doping to systematic bits only, which we refer to as systematic VN doping. This follows from the fact that ① doping can now be done directly on the information sequence, prior to encoding, thus greatly simplifying the encoding process, and ② the encoder inverse operation is trivial in this case, since all the information bits appear unchanged as code bits in the encoded sequence, and thus the estimated information sequence can be determined directly from the estimated code sequence.[20] Since only a fraction of the bits (depending on the design rate R

[19] Fractional doping can also be applied to CN doping, but we focus only on fractional VN doping here.

[20] We note that such a strategy can be implemented with only minor modifications to the encoding process by occasionally fixing input symbols to a standard systematic encoder and removing those symbols after recovering the decoded information sequence.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

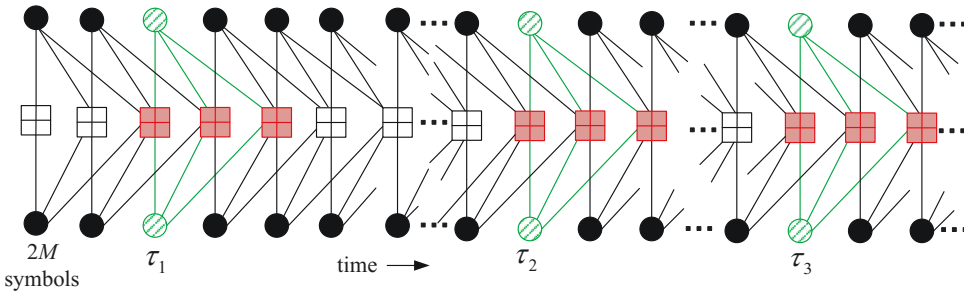


Fig. 9. Fractional VN doping for a (3,6)-regular SC-LDPC code with occasional partially doped VNs spaced throughout the coupled chain.

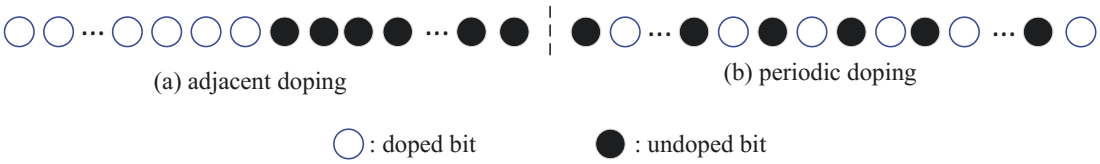


Fig. 10. Adjacent and periodic doping patterns for fractional VN doping with $\delta = 0.5$.

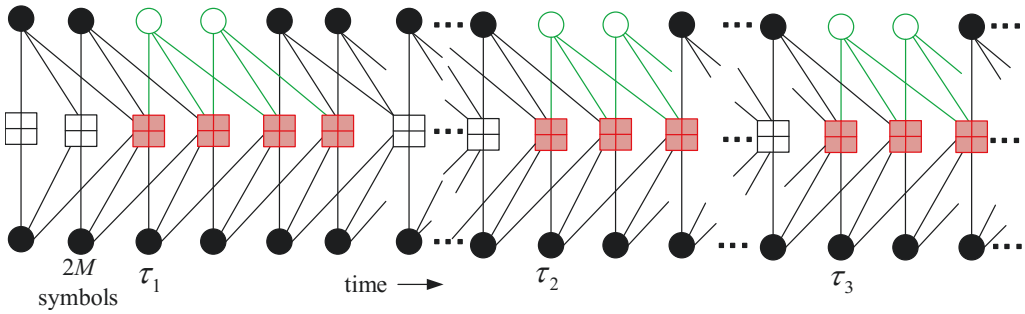


Fig. 11. Systematic VN doping with $\delta = R = 0.5$ and $\sigma = 1/R = 2$.

of the underlying LDPC-BC) at each position are systematic, a systematic doping strategy necessitates spreading fractional doping over a span of $\sigma > 1$ positions in order to dope the same number of bits as full doping of all the protograph nodes at any one position. However, as we will see in Sec. V.C, this can be achieved with essentially no loss in performance, and even fractional systematic doping at one position can sometimes perform as well as full doping.

To illustrate the procedure, we again use (3,6)-regular SC-LDPC codes as an example, for which $R = 1/2$. In order to dope the same number of bits as full doping of a single position, systematic VN doping requires doping a fraction $\delta \leq R = 1/2$ of bits over $\sigma \geq 1/R = 2$ consecutive doping positions such that $\sigma\delta = 1$, where only systematic VNs are doped. This is illustrated in Fig. 11 for $\delta = 0.5$ and $\sigma = 2$, where the white circles represent the doped systematic protograph nodes and we assume that the upper protograph node at each position contains systematic bits only, while the lower protograph node contains parity bits only, which corresponds to the adjacent doping pattern shown in Fig. 10(a). At each doping position, say $t = \tau_1$, only a fraction $\delta = R = 0.5$ of the protograph nodes are doped and a second systematic protograph node is doped at the next position $t = \tau_1 + 1$, making fractional ($\delta = 0.5$) systematic doping over a span of $\sigma = 2$ positions equivalent to full doping of a single position, which necessarily entails the doping of parity bits. Similarly, at time units $t = \tau_2, t = \tau_3, \ldots$, systematic doping covers a span of $\sigma = 2$ positions, such that $\sigma\delta = 1$.[21]

Systematic doping can be applied in the same manner to general $(d_\mathrm{v}, d_\mathrm{c})$-regular SC-LDPC codes with underlying LDPC-BC design rate $R = 1 - d_\mathrm{v}/d_\mathrm{c}$. Here, we consider (3,9)-regular SC-LDPC codes with rate $R = 2/3$ and (4,6)-regular SC-LDPC codes with $R = 1/3$ as examples. In the $R = 2/3$ (3,9)-regular case with coupling memory $m_s = 2$, derived from the $n_\mathrm{c} \times n_\mathrm{v} = 1 \times 3$ LDPC-BC base matrix $B = [3\ 3\ 3]$, the coupled chain formed by applying the edge-spreading technique to the uncoupled protograph is shown in Fig. 12, where $d_\mathrm{v} = 3n_\mathrm{c} = 3$, $d_\mathrm{c} = 3n_\mathrm{v} = 9$, and we see that there are three protograph VNs at each time unit. Therefore, in order to implement systematic doping, we can consider two options:

- doping span $\sigma = 2$. In this case, systematic doping operates over $\sigma = 2$ time units, as shown in Fig. 13(a), where two systematic protograph nodes are doped at time

[21] In general, we note that $\sigma\delta$, the equivalent number of fully doped positions, need not equal 1. In other words, as long as $\delta \leq R$, systematic VN doping can be spread over any number σ of consecutive positions.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

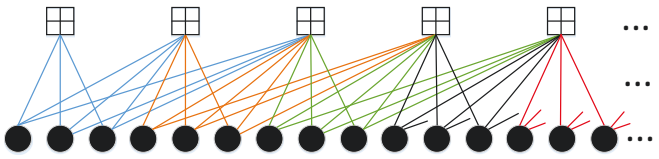


Fig. 12. Coupled chain for (3,9)-regular SC-LDPC codes.

τ_1 and one systematic protograph node is doped at time $\tau_1 + 1$.

- doping span $\sigma = 3$. In this case, systematic doping operates over $\sigma = 3$ time units, as shown in Fig. 13(b), where one systematic protograph node is doped at times $\tau_1, \tau_1 + 1$, and $\tau_1 + 2$.

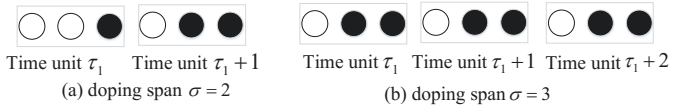


Fig. 13. Systematic doping options for (3,9)-regular SC-LDPC codes.

In the $R = 1/3$ (4,6)-regular case with coupling memory $m = 1$, formed from the $n_c \times n_v = 2 \times 3$ LDPC-BC base matrix

$$B = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix}, \tag{20}$$

applying edge spreading results in the coupled chain shown in Fig. 14, where $d_v = 2n_c = 4$ and $d_c = 2n_v = 6$. Since there

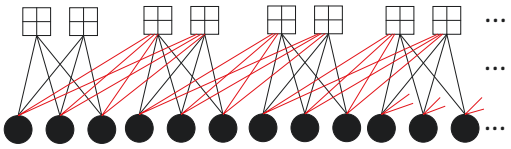


Fig. 14. Coupled chain for (4,6)-regular SC-LDPC codes.

are three protograph nodes at each time unit, one protograph node can be considered as the systematic node and the other two as the parity check nodes. Thus, for systematic doping, only one protograph node can be doped at each position, which results in the doping option shown in Fig. 13(b).

C. Numerical Results

In order to verify the effectiveness of systematic VN doping, we first consider a (3,9)-regular SC-LDPC code with design rate $R = 2/3$ and coupling memory $m = 2$ lifted from the coupled chain shown in Fig. 12, where we use the systematic doping option of Fig. 13(b). The simulated performance for $M = 1000$, $W = 12$, $I_{\max} = 50$, $L = 500$, and $R_L^{\mathrm{FR}} = 0.665$ is presented in Fig. 15(a). We see that systematic doping i.e., fractional doping of $\sigma = 3$ protograph nodes, achieves essentially the same BER and BLER performance as full doping of a single position, which involves a much more complex encoding process. We also note that the total number of doped VNs is $3\delta\sigma M = 3M$, both for systematic doping ($\delta = 1/3, \sigma = 3$) and full doping ($\delta = 1, \sigma = 1$).

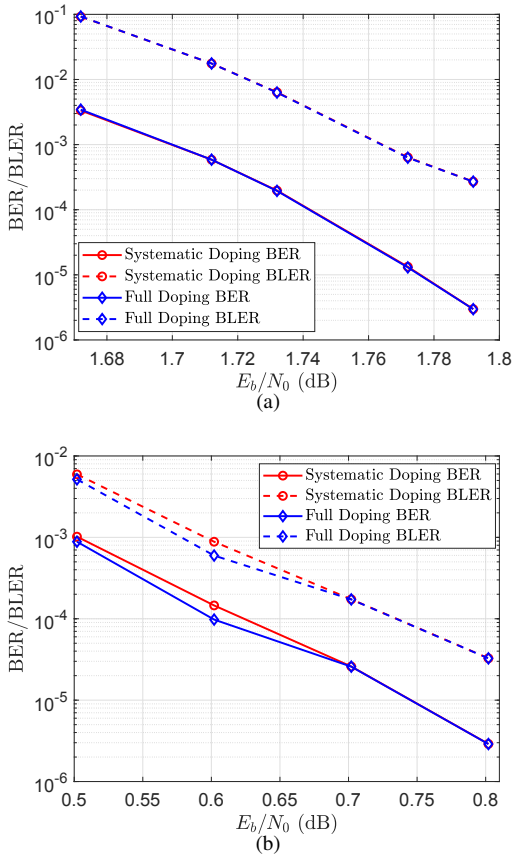


Fig. 15. Performance comparison between full doping and systematic doping for (a) a (3,9)-regular and (b) a (4,6)-regular SC-LDPC code.

We then consider a (4,6)-regular SC-LDPC code with design rate $R = 1/3$ and coupling memory $m = 1$ lifted from the coupled chain shown in Fig. 14, where again we use the systematic doping option of Fig. 13(b). For the same set of parameters as in Fig. 15(a)[22] and $R_L^{\mathrm{FR}} = 0.331$, the simulation results are shown in Fig. 15(b), where we see only a very slight (< 0.025 dB) performance loss for systematic doping at BER/BLER $> 10^{-5}/10^{-4}$ compared to the much more complex full doping of a single position, and we again note that the total number of doped VNs is $3M$ in both cases.

Finally, in order to test the effectiveness of general (systematic or nonsystematic) fractional VN doping at reducing rate loss, we simulated the (3,6)-regular SC-LDPC code of Fig. 9 with $M = 2000$, $W = 18$, $I_{\max} = 50$, and $L = 500$ at an SNR of $E_b/N_0 = 0.9$ dB. A single ($\sigma = 1$) doping position was placed in the middle of the coupled chain and the doping fraction δ was varied between 0 (no doping) and 1 (full doping). From Fig. 16(a), we see that even a small amount of fractional doping yields significant performance improvement, with $\delta = 0.2$ (20% doping, $R_L^{\mathrm{FR}} = 0.498$) giving essentially the same result as $\delta = 1.0$ (full doping, $R_L^{\mathrm{VN}} = 0.497$). This is consistent with the analytical results of [15] for a BEC, where the authors show that doping operates like a switch, i.e., a decoding wave is either initiated or it is not, and suggests

[22] In Fig. 15(b), we use $W = 8$, four decoding constraint lengths, the same as for Fig. 15(a)

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

that the systematic doping results presented above can also be achieved by doping only a single position, thus reducing the rate loss due to doping, as long as R exceeds some critical doping fraction. To test this hypothesis, we simulated the (3,6)-regular SC-LDPC code with $M = 2000$, $W = 12$, $I_{\max} = 50$, and $L = 250$. A single ($\sigma = 1$) doping position was placed in the middle of the coupled chain, with doping fraction $\delta = 0.5$. The results are shown in Fig. 16(b), where we see that fractional systematic doping with $\delta = 0.5$ achieves essentially the same the BER and BLER performance as full doping, and the rate in this case increases from $R_L^{\mathrm{VN}} = 0.494$ to $R_L^{\mathrm{FR}} = 0.495$. Finally, we note that these results suggest that fractional doping of multiple positions in a coupled chain will be more effective at mitigating decoder error propagation than fully doping a single position.

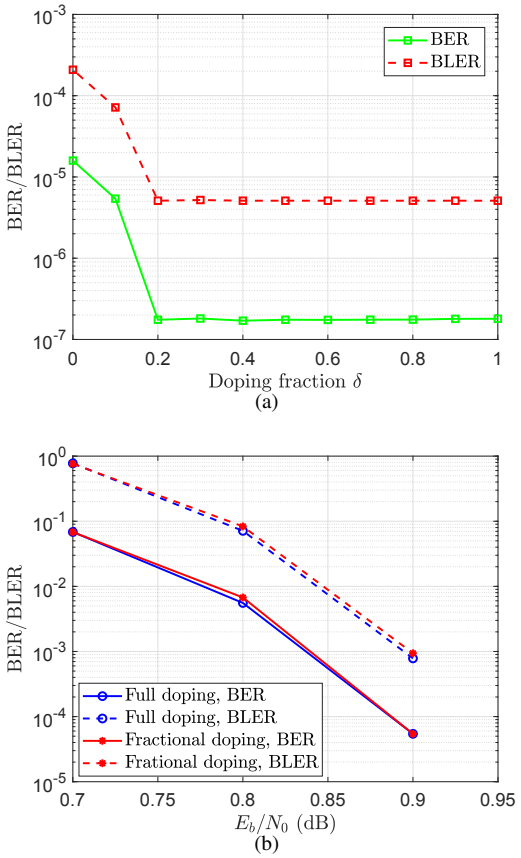


Fig. 16. Performance of a fractionally doped (3,6)-regular SC-LDPC code for (a) $M = 2000$, $W = 18$, $L = 500$, and $E_b/N_0 = 0.9$ dB and (b) $M = 2000$, $W = 12$, $L = 250$, and $\delta = 0.5$.

VI. Conclusion

In this paper we studied the use of code doping techniques for SC-LDPC codes with SWD to mitigate the decoder error propagation problem, which can severely degrade code performance when low latency operation at SNRs below the BP threshold of the underlying LDPC-BC is required. Doping was shown to effectively combat decoder error propagation without terminating encoding, thus enabling streaming transmission or large frame length applications. We began by introducing a multi-state decoder model to describe the behavior and analyze the BLER performance of SWD affected by error propagation. Then three doping methods (CN doping, VN doping, and adaptive doping) were presented to limit the effects of error propagation. An analysis of the BLER of doped SC-LDPC codes based on the multi-state decoder model indicated that doping results in substantially improved BLER performance under low latency operating conditions and that further improvement can be obtained by increasing the number of doping positions. This improved performance comes at a cost of some slight rate loss, which was shown to be always less than that resulting from encoder termination. Computer simulation results were also used to demonstrate the beneficial effects of code doping and the additional improvement that can be achieved by employing a feedback channel to adapt the doping to the observed decoder behavior. Finally, the encoding problem associated with doping was shown to be easily solved without any significant effect on performance by assuming a systematic encoder and restricting the doping to systematic bits only, i.e., by only doping a fraction of the bits at each position. As a result, systematic VN doping of SC-LDPC codes represents a practical approach to achieving near capacity performance with limited decoding latency.

Appendix A

Here we modify the analysis of Sec. III to cover the case of terminated (finite L) transmission.

Case A1: No intermediate states ($J = 1$), finite L.

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

$$x = d_0 + d_1 = \frac{1}{q_0} + \frac{1}{1-q_1} = \frac{1-q_1+q_0}{q_0(1-q_1)} = \frac{1-q_1+r_1}{q_0(1-q_1)}, \tag{21}$$

and the *average number of cycles* is $y = L/x$.[23]

Now 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

$$\bar{n}_0 = \begin{cases} \lfloor y \rfloor d_0 q_0 = \lfloor y \rfloor, \text{for } z < U_0 \text{ (since } d_0 q_0 = 1) \\ (\lfloor y \rfloor + 1) d_0 q_0 = \lfloor y \rfloor + 1, \text{for z} \geq U_0, \end{cases} \tag{22}$$

$$\bar{n}_1 = \begin{cases} \lfloor y \rfloor d_1 q_1 = \lfloor y \rfloor \frac{q_1}{1-q_1}, \text{for } z < U_0 \\ \left(\lfloor y \rfloor + \frac{(z-U_0)}{U_1} \right) \frac{q_1}{1-q_1}, \text{for z} \geq U_0, \end{cases} \tag{23}$$

and the *average BLER* can be expressed as

$$P_{\mathrm{BL}}^{(L)} = \frac{\bar{n}_0 + \bar{n}_1}{L}. \tag{24}$$

Example A1: Choose $L = 1000$, $q_0 = 0.01$, and $q_1 = 0.99$. Hence we have $d_0 = 100$, $d_1 = 100$, $x = 200$, $y = 5$, $z = 0$,

[23]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.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

and $U_0 = U_1 = 0.5$. Substituting these values into (22) and (23), 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 $\lim_{L\to\infty} P_{\rm BL}^{(L)} = \frac{1+(L-100)}{L} = 1$. Finally, we note that, for any L such that $z = 0$, i.e., when the decoder traverses an integral number of cycles, $P_{\rm BL}^{(L)} = P_{\rm BL}^{(\infty)}$. □

Case A2: One intermediate state ($J = 2$), finite L.

In this case, 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. Denote the average dwell times in each cycle as $d_i^{(k)}, i = 0, 1, 2, k = 1, 2$, where $d_0^{(1)} = d_0^{(2)} = \frac{1}{q_0}$, $d_1^{(1)} = d_1^{(2)} = 1$, 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

$$\begin{aligned} x &= \frac{1+q_0}{q_0}(1-q_1) + \frac{q_0 + (1+q_0)(1-q_2)}{q_0(1-q_2)} q_1 \\ &= \frac{1 - q_2 + q_0(1 - q_2 + q_1)}{q_0(1-q_2)} = \frac{1 - q_2 + r_2}{q_0(1-q_2)}. \end{aligned} \tag{25}$$

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

$$\begin{aligned} U_0 + U_1 + U_2 &= \frac{1}{q_0 x} + \frac{1}{x} + \frac{q_1}{(1-q_2)\, x} \\ &= \frac{(1-q_2) + q_0(1-q_2) + q_0 q_1}{q_0(1-q_2)\, x} \\ &= \frac{1 - q_2 + r_2}{q_0(1-q_2)\, x} = 1. \end{aligned} \tag{26}$$

Letting $\bar{n}_i$ be the average number of block errors in state $S_i, i = 0, 1, 2$, we then have[24]

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

$$\overline{n}_1 = \begin{cases} \lfloor y \rfloor\, \bar{d}_1 q_1 = \lfloor y \rfloor\, q_1 & \text{for } z < U_0 + U_1 \\ (\lfloor y \rfloor + 1)\, \bar{d}_1 q_1 = (\lfloor y \rfloor + 1)\, q_1 & \text{for } \ z \geq U_0 + U_1 \end{cases} \tag{28}$$

[24]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.

$$\overline{n}_2 = \begin{cases} \lfloor y \rfloor\, \bar{d}_2 q_2 = \lfloor y \rfloor\, \frac{q_1 q_2}{1-q_2} & \text{for } z < U_0 + U_1 \\ \left(\lfloor y \rfloor + \frac{z - U_0 - U_1}{U_2}\right) \frac{q_1 q_2}{1-q_2} & \text{for } z \geq U_0 + U_1, \end{cases} \tag{29}$$

and the average BLER can be written as

$$P_{\rm BL}^{(L)} = \frac{\overline{n}_0 + \overline{n}_1 + \overline{n}_2}{L}. \tag{30}$$

Example A2: Choose $L = 1000$, $q_0 = 0.01$, $q_1 = 0.1$, and $q_2 = 0.99$. Then $r_1 = 0.0011$, $\bar{d}_0 = 100$, $\bar{d}_1 = 1$, $\bar{d}_2 = 10$, $x = 111$, $y = 9.009$, $\lfloor y \rfloor = 9$, $z = 0.009$, $U_0 = 0.9009$, $U_1 = 0.009$, and $U_2 = 0.0901$ ($z < U_0 < U_0 + U_1$). Now using (27), (28), and (29), 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, $P_{\rm BL}^{(1000)} \approx P_{\rm BL}^{(\infty)}$ (from ***Ex. 2***), since the last cycle never reaches state S_2 on the average ($z < U_0 + U_1$). □

Case A3: $J \geq 3$, finite L.

From (21) and (25), it follows that $x = \frac{1 - q_J + r_J}{q_0(1-q_J)}$ is the general expression for the average cycle time and

$$U_i = \frac{\bar{d}_i}{x}, \quad i = 0, 1, \ldots, J. \tag{31}$$

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

$$\begin{cases} P\left(C^{(1)}\right) = 1 - q_1 \\ P\left(C^{(k)}\right) = q_1 q_2 \cdots q_{k-1}\,(1 - q_k)\,, k = 2, 3, \ldots, J-1 \\ P\left(C^{(J)}\right) = q_1 q_2 \cdots q_{J-1}, \end{cases} \tag{32}$$

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

$$\begin{cases} \bar{d}_0 = \dfrac{1}{q_0}, \quad \bar{d}_1 = 1, \\ \bar{d}_i = 1 - \displaystyle\sum_{k=1}^{i-1} P\left(C^{(k)}\right) = q_1 q_2 \cdots q_{i-1}, i = 2, 3, \ldots, J-1 \\ \bar{d}_J = \dfrac{q_1 q_2 \cdots q_{J-1}}{1 - q_J}, \end{cases} \tag{33}$$

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 is then given by

$$\bar{n}_0 = \begin{cases} \lfloor y \rfloor\, \bar{d}_0 q_0 = \lfloor y \rfloor & \text{for } z < U_0 \\ (\lfloor y \rfloor + 1)\, \bar{d}_0 q_0 = \lfloor y \rfloor + 1 & \text{for z} \geq U_0 \end{cases} \tag{34}$$

$$\bar{n}_j = \begin{cases} \lfloor y \rfloor\, \bar{d}_j q_j = \lfloor y \rfloor\, q_1 q_2 \cdots q_j, & \text{for } z < U_0 + U_1 + \cdots + U_j \\ (\lfloor y \rfloor + 1)\, \bar{d}_j q_j = (\lfloor y \rfloor + 1)\, q_1 q_2 \cdots q_j, & \\ \qquad j = 1, 2, \ldots, J-1, \text{for z} \geq U_0 + U_1 + \cdots + U_j & \end{cases} \tag{35}$$

$$\bar{n}_J = \begin{cases} \lfloor y \rfloor\, \bar{d}_J q_J = \lfloor y \rfloor\, \frac{q_1 q_2 \cdots q_J}{1-q_J}, & \text{for } z < U_0 + U_1 + \cdots + U_{J-1} \\ \left(\lfloor y \rfloor + \frac{z - U_0 - U_1 - \cdots - U_{J-1}}{U_J}\right) \frac{q_1 q_2 \cdots q_J}{1-q_J}, & \\ \qquad \text{for z} \geq U_0 + U_1 + \cdots + U_{J-1}, & \end{cases} \tag{36}$$

and the average BLER is given by

$$P_{\rm BL}^{(L)} = \sum_{i=0}^{J} \bar{n}_i / L, \tag{37}$$

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

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

APPENDIX B

We now describe how the model parameters can be estimated from the results of a single simulation run. First, choose an operating channel SNR of interest, typically below the iterative decoding threshold of the underlying LDPC-BC. Then 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 a total of N frames have been simulated, each of length L, for a total of LN simulated blocks[25], this data file gives the total number of *finite-length error bursts* of each length contained in all N frames. Then 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 decide the number of states to be included in the model, i.e., 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, normally J is chosen just large enough to include those burst lengths that occur most often, while combining the occasional longer finite-length bursts with the EOF bursts.[26]

Once J has been set, it is straightforward to determine the model parameters. Begin by letting λ_j be the number of burst errors of length j, $j = 1, 2, \ldots, J$, where $\lambda_J = \lambda_{\rm FL} + \lambda_{\rm EOF}$, $\lambda_{\rm FL}$ is the number of finite-length error bursts of length J or greater (if any), and $\lambda_{\rm EOF}$ is the number of EOF bursts. Also let $\delta_J = \delta_{\rm FL} + \delta_{\rm EOF}$ 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

$$T_j = \sum_{i=j}^{J} \lambda_i, \qquad j = 1, 2, \ldots, J-1, \tag{38}$$

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

$$E_j = \sum_{i=j+1}^{J} \lambda_i, \qquad j = 1, 2, \ldots, J-1, \tag{39}$$

and it follows that

$$q_j = E_j / T_j, \qquad j = 1, 2, \ldots, J-1. \tag{40}$$

To find q_0, 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

$$T_0 = LN - \sum_{i=1}^{J-1} i\lambda_i - \delta_J, \tag{41}$$

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

$$E_0 = \sum_{i=1}^{J} \lambda_i = T_1, \tag{42}$$

and it follows that $q_0 = E_0/T_0$. Finally, to compute q_J, 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 $T_J = \delta_J - (J-1)\lambda_J$, the total number of block errors E_J made in state S_J[27] is $E_J = T_J - \lambda_J$, and $q_J = E_J/T_J$.

Example B1: $N = 20,000$ frames of length $L = 5000$ were simulated for the SC-LDPC code of Fig. 1(b) with $M = 1000$, $I_{\max} = 50$, $W = 12$, and $E_b/N_0 = 0.9$ dB, which is 0.2 dB below the BP threshold of the underlying (3,6)-regular LDPC-BC, resulting in a decoded BLER = 0.4670.[28] 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 $\lambda_1 = 7957$, $\lambda_2 = 1762$, $\lambda_3 = 666$, $\lambda_4 = 261$, $\lambda_{\rm FL} = 178$, $\lambda_{\rm EOF} = 15,104$, $\delta_{\rm FL} = 118,157$, and $\delta_{\rm EOF} = 46,557,890$. 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 the above procedure. Then, from (12), the asymptotic average BLER is given by $P_{\rm BL}^{(\infty)} = 0.4670$, where we note that the agreement with the simulated result in this case is due to the fact that the simulated frames were quite long ($L = 5000$).

Now considering the finite length analysis, (12) and (31) - (36) can be used 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 see that almost all the block decoding errors occur in state S_5, the burst error state. It follows from (37) that the average simulated BLER is given by $P_{\rm BL}^{(5000)} = 0.3602$.[29] □

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 the code doping techniques presented in this paper, 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.[30] 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 position.

Example B1 (Cont.): For $L = 2500$, we left the values of

[25]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.

[26]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.

[27]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.

[28]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.

[29]The analysis here assumes that the dwell times in states S_0 and S_J are always equal to their average values. It is more realistic to assume that these dwell times can be modeled by a binomial probability distribution, which improves the accuracy of the results at a cost of some added complexity.

[30]For the same reason noted in Footnote 4, this statement is not exact.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

$q_0, q_1, \ldots, q_4$ 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 $q_5 = 0.9996$.[31] Again using (12) and (31) - (37), we obtain $P_{\mathrm{BL}}^{(2500)} = 0.1781$,[32] a roughly 50% reduction in the estimated BLER compared to $L = 5000$, which reflects the expected performance gain that can be achieved with doping.[33] 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 $L = 5000$. □

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] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 7761-7813, Dec. 2013.

[5] S. Kumar, A. J. Young, N. Macris, and H. D. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7389-7415, Dec. 2014.

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

[7] A. R. Iyengar, M. Papaleo, P. H. Siegel, J. K. Wolf, A. Vanelli-Corolli, 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.

[8] K. Huang, D. G. M. Mitchell, L. Wei, X. Ma, and D. J. Costello, Jr., "Performance comparison of LDPC block and spatially coupled codes over GF(q)," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 592-604, Mar. 2015.

[9] V. Aref, N. Rengaswamy, and L. Schmalen, "Finite-length analysis of spatially-coupled regular LDPC ensembles on burst-erasure channels," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3431-3449, May 2018.

[10] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, Jr., and B. Bai, "Error propagation mitigation in sliding window decoding of braided convolutional codes," *IEEE Trans. Commun.*, vol. 68, no. 11, pp. 6683-6698, Nov. 2020.

[11] M. Zhu, D. G. M. Mitchell, M. Lentmaier, D. J. Costello, Jr., and B. Bai, "Combating error propagation in window decoding of braided convolutional codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, June 17-22, 2018, pp. 1380-1384.

[12] 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.

[13] 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. Inf. Theory*, Los Angeles, CA, USA, June 21-26, 2020, pp. 473-478.

[14] 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* (ISITA), Kapolei, Hawai'i, USA, October 24-27, 2020, pp. 175-179.

[15] R. Sokolovskii, A. Graell i Amat, and F. Brännström, "On doped SC-LDPC codes for streaming," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2123-2127, July 2021.

[16] M. Zhu, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "Adaptive doping of spatially coupled LDPC codes," in *IEEE Information Theory Workshop* (ITW), Riva del Garda, Italy, April 11-15, 2021, pp. 1-5.

[17] M. Zhu, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "Modeling a sliding window decoder for spatially coupled LDPC codes," in *Proc. IEEE Global Commun. Conf. Workshops*, Madrid, Spain, December 7-11, 2021, pp. 1-6.

[18] M. Zhu, D. G. M. Mitchell, M. Lentmaier, and D. J. Costello, Jr., "Systematic doping of SC-LDPC codes," in *Proc. IEEE Int. Symp. Information Theory*, Espoo, Finland, June 26-July 01, 2022, pp. 536-541.

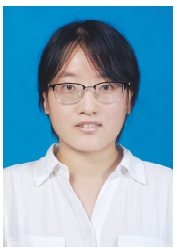
[19] S. Cammerer, V. Aref, L. Schmalen, and S. ten Brink, "Triggering wave-like convergence of tail-biting spatially coupled LDPC codes," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2016, pp. 93-98.

[20] S. ten Brink, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *Electronics Letters*, vol. 36, no. 15, pp. 1-2, Jul. 20, 2000.

[31] 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.

[32] Assuming a simple 3-valued probability distribution for the dwell time in state S_0 improves the result to $P_{\mathrm{BL}}^{(2500)} = 0.2331$.

[33] The expected doping gain depends on the values of M, W, and E_b/N_0. As noted in Footnote 17, doping only improves performance under conditions for which the decoder suffers from error propagation.



Min Zhu (S'15-M'17) received the B.S., the M.S. and the Ph.D. degrees in communication and information system from Xidian University, China, in 2006, 2009, and 2016 respectively. From Sept. 2014 to Sept. 2015, she was with the Department of Electrical Engineering, University of Notre Dame, USA, as a visiting Ph. D student. She is currently with State Key Lab. of ISN, Xidian University. Her research interests include channel coding and their applications to communication systems.

David G. M. Mitchell (M'10-SM'16) received the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 2009. From 2009 to 2015, he held Post-Doctoral Research Associate and Visiting Assistant Professor positions with the Department of Electrical Engineering, University of Notre Dame, USA. Since 2015, he has been an Assistant Professor with the Klipsch School of Electrical and Computer Engineering, New Mexico State University, USA. His research interests lie in the area of digital communications, with emphasis on error control coding and information theory. He currently serves as an Associate Editor for the IEEE Transactions on Information Theory.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Journal on Selected Areas in Information Theory. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSAIT.2023.3312656

Michael Lentmaier (S'98-M'03-SM'11) is an Associate Professor at the Department of Electrical and Information Technology at Lund University, which he joined in January 2013. His research interests include design and analysis of coding systems, graph based iterative algorithms and Bayesian methods applied to decoding, detection and estimation in communication systems. He received the Dipl.-Ing. degree in electrical engineering from University of Ulm, Germany in 1998, and the Ph.D. degree in telecommunication theory from Lund University, Sweden in 2003. He then worked as a Post-Doctoral Research Associate at University of Notre Dame, Indiana and at University of Ulm. From 2005 to 2007 he was with the Institute of Communications and Navigation of the German Aerospace Center (DLR) in Oberpfaffenhofen, where he worked on signal processing techniques in satellite navigation receivers. From 2008 to 2012 he was a senior researcher and lecturer at the Vodafone Chair Mobile Communications Systems at TU Dresden, where he was heading the Algorithms and Coding research group. He is a senior member of the IEEE and served as an editor for IEEE Communications Letters (2010-2013), IEEE Transactions on Communications (2014-2017), and IEEE Transactions on Information Theory (2017-2020). He was awarded the Communications Society & Information Theory Society Joint Paper Award (2012) for his paper "Iterative Decoding Threshold Analysis for LDPC Convolutional Codes".

Daniel J. Costello, Jr. (S'62-M'69-SM'78-F'85-LF'08) received the M.S. and Ph.D. degrees in Electrical Engineering from the University of Notre Dame, Notre Dame, IN, in 1966 and 1969, respectively. Dr. Costello joined the faculty of the Illinois Institute of Technology, Chicago, IL, in 1969. In 1985 he became Professor of Electrical Engineering at the University of Notre Dame, Notre Dame, IN, and from 1989 to 1998 served as Chair of the Department of Electrical Engineering. In 2000, he was named the Leonard Bettex Professor of Electrical Engineering at Notre Dame, and in 2009 he became Bettex Professor Emeritus.

Dr. Costello has been a member of IEEE since 1969 and was elected Fellow in 1985. In 2009, he was co-recipient of the IEEE Donald G. Fink Prize Paper Award, which recognizes an outstanding survey, review, or tutorial paper in any IEEE publication issued during the previous calendar year. In 2012, he was a co-recipient of the joint IEEE Information Theory Society/Communications Society Prize Paper Award, which recognizes an outstanding research paper in the IT or COM Transactions during the previous two calendar years. In 2013, he received the Aaron D. Wyner Distinguished Service Award from the IEEE Information Theory Society, which recognizes outstanding leadership in and long standing exceptional service to the Information Theory community. In 2015 he received the IEEE Leon K. Kirchner Graduate Teaching Award, which recognizes inspirational teaching of graduate students in the IEEE fields of interest.

Dr. Costello's research interests are in digital communications, with special emphasis on error control coding and coded modulation. He has numerous technical publications in his field, and in 1983 he co-authored a textbook entitled "Error Control Coding: Fundamentals and Applications", the 2nd edition of which was published in 2004.

Authorized licensed use limited to: UNIVERSITY OF NEW MEXICO. Downloaded on September 12,2023 at 19:37:35 UTC from IEEE Xplore. Restrictions apply.
© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.