Bounds on the List Size of Successive Cancellation List Decoding

Mustafa Cemil Coşkun*† and Henry D. Pfister‡
*German Aerospace Center (DLR) †Technical University of Munich (TUM) ‡Duke University
Email: mustafa.coskun@tum.de, henry.pfister@duke.edu

***Abstract*—Successive cancellation list decoding of polar codes provides very good performance for short to moderate block lengths. However, the list size required to approach the performance of maximum-likelihood decoding is still not well understood theoretically. This work identifies information-theoretic quantities that are closely related to this required list size. It also provides a natural approximation for these quantities that can be computed efficiently even for very long codes. Simulation results are provided for the binary erasure channel as well as the binary-input additive white Gaussian noise channel.**

I. Introduction

Polar codes constitute the first deterministic construction of capacity-achieving codes for binary memoryless symmetric (BMS) channels with an efficient decoder [1]. While proven to achieve capacity under successive cancellation (SC) decoding, their initial performance results were not competitive with low-density parity-check and Turbo codes in practice. This changed with the advent of SC list (SCL) decoding and the addition of cyclic redundancy-check (CRC) outer codes [2]. Due to their competitive performance for short block lengths [3], they have been adopted by the 5G standard. Now, there is a significant research effort into improving their performance. For example, many authors have optimized polar codes and their variants for the SCL decoder [4]–[10].

An important property of the SCL decoder is that its performance matches that of maximum-likelihood (ML) decoding if the list size increases without bound. In this work, we consider the theoretical question: "What list size is sufficient to achieve ML decoding performance for a given channel quality?". While it is possible to simulate the SCL with a large list size and compare the results with the simulation-based ML lower bound, that approach becomes infeasible for long codes and doesn't provide much insight into the question. Such an insight has potential to suggest a constructive way to design polar codes for SCL decoding while the current works rely mostly on heuristics, e.g., see [5]–[7], [9]. Our results identify information-theoretic quantities associated with the required list size and also provide a natural approximation that can be computed efficiently even for very long codes. Using these quantities, our analysis suggests new code design criteria for polar code variants. Simulations are provided for the binary erasure channel (BEC) and the binary-input additive white Gaussian noise channel (biAWGNC).

The work of M. C. Coşkun was supported by the Helmholtz Gemeinschaft through the HGF-Allianz DLR@Uni project Munich Aerospace via the research grant "Efficient Coding and Modulation for Satellite Links with Severe Delay Constraints". The work of H. Pfister was supported in part by the National Science Foundation under Grant No. 1718494. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these sponsors.

978-1-7281-8895-9/20/$31.00 © 2020 IEEE

II. Background

Random variables (RVs) are denoted by upper case letters, e.g., X, and their realizations by the lower case counterparts, e.g., x. Vectors are denoted by $x_i^j = (x_i, x_{i+1}, \dots, x_j)$. If $j < i$, then it is void. We use $[N]$ for the set $\{1, 2, \dots N\}$. Subvectors with indices in $\mathcal{A} \subseteq [N]$ are denoted by $x_{\mathcal{A}} = (x_{i_1}, \dots, x_{i_{|\mathcal{A}|}})$ where $i_1 < \cdots < i_{|\mathcal{A}|}$ is an enumeration of the elements in $\mathcal{A}$ with $|\mathcal{A}|$ being the cardinality of the set $\mathcal{A}$. The length-N all-zero vector is denoted as 0^N. Finally, bold capital letters are used for matrices, e.g., $\boldsymbol{X}$.

Consider a BMS channel, $W : X \to Y$, with binary input $X \in \{0, 1\}$ and general output $Y \in \mathcal{Y}$. The transition probabilities are given by $W(y|x) \triangleq \Pr(Y = y|X = x)$ and we assume w.l.o.g. that symmetry implies $W(y|1) = W(-y|0)$.

A. Polar Codes and Successive Cancellation Decoding

The polar transform of length $N = 2^n$ is denoted by $\boldsymbol{G}_N \triangleq \boldsymbol{G}_2^{\otimes n}$ and equals to the n-fold Kronecker product of the 2×2 Hadamard matrix, i.e.,

$$\boldsymbol{G}_2 \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

This is the key building block in Arıkan's polar codes [1].

To define a polar code, one needs to partition the input vector into bits that carry information and frozen bits whose values are known by the receiver (e.g., fixed to 0). The set of information and frozen indices are denoted, respectively, by $\mathcal{A} \subseteq [N]$ and $\mathcal{F} \triangleq [N] \setminus \mathcal{A}$. Thus, the input vector u_1^N can be split into information bits $u_{\mathcal{A}}$ and frozen bits $u_{\mathcal{F}}$. Then, the codeword $x = u\boldsymbol{G}_N$ is transmitted over the channel. This construction also enables efficient SC decoding [1].

Let y_1^N be the observations of the bits x_1^N through N copies of the BMS channel W. The SC decoder takes the following steps sequentially from $i = 1$ to $i = N$. If $i \in \mathcal{F}$, it sets $\hat{u}_i$ to its frozen value. If $i \in \mathcal{A}$, it computes the soft estimate $p_i(\hat{u}^{i-1}) \triangleq \Pr(U_i = 1|Y_1^N = y_1^n, U^{i-1} = \hat{u}^{i-1})$, and makes a hard decision accordingly as

$$\hat{u}_i = \begin{cases} 0 & \text{if } p_i(\hat{u}^{i-1}) < \frac{1}{2} \\ 1 & \text{otherwise.} \end{cases}$$

To understand the SC decoder, we focus now on the effective channels seen by each of the input bits in u_1^N [1]. The SC decoder uses the entire y_1^N vector and all past decisions $\hat{u}_1^{i-1}$ to generate the soft estimate $p_i(\hat{u}^{i-1})$ and the hard decision $\hat{u}_i$ for u_i. Let $W_N^{(i)}$ denote the effective (virtual) channel seen by u_i during the SC decoding [1]. If all past bits u_1^{i-1} are provided by a genie, then this channel is easier to analyze. Under that assumption, the effective channel $W_N^{(i)} : U_i \to (Y_1^N, U_1^{i-1})$ is defined by its transition probabilities

$$W_N^{(i)}\left(y_1^N, u_1^{i-1}|u_i\right) \triangleq \sum_{u_{i+1}^N \in \{0,1\}^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N \boldsymbol{G}_N)$$

where $W_N(y_1^N|x_1^N) \triangleq \prod_{i=1}^N W(y_i|x_i)$.

B. Dynamic Frozen Bits

An important observation in [4] is that the SC decoder still works (with a slight modification) if, for some $i \in \mathcal{F}$, the bit u_i is a function of a set of preceding information bits. A frozen bit whose value depends on past inputs is called dynamic.

A polar code with dynamic frozen bits is defined by its information indices $\mathcal{A}$ and a matrix that defines each frozen bit as a linear combination of preceding information bits. There are now a number of approaches for choosing these parameters [4], [5] and heuristic design methods [6]. In this work, after specifying $\mathcal{A}$, we define each frozen bit to be a uniform random linear combination of information bits preceding it.

C. Successive Cancellation List Decoding

SCL decoding of Reed–Muller (RM) codes (and related subcodes) was introduced in [11]. These ideas were extended to optimized constructions of generalized concatenated codes in [12]. But, these approaches became popular only after [2] applied them to polar codes combined with an outer CRC code to increase the minimum distance.

The basic idea of SCL decoding is to recursively compute the value of $Q_i(\tilde{u}_1^i, y_1^N) \propto \Pr\left(U_1^i = \tilde{u}_1^i, Y_1^N = y_1^N\right)$ for $i = 1, \ldots, N$ via the SC message passing rules for partial information sequences $\tilde{u}_1^i \in \mathcal{U}_i \subseteq \{0,1\}^i$. Dropping y_1^N for the ease of notation, we refer to the quantity $Q_i(\tilde{u}_1^i)$ as the *myopic likelihood* of the sequence $\tilde{u}_1^i$ as it does not use the receiver's knowledge of frozen bits after u_i.

Let $\mathcal{U}_{i-1} \subseteq \{0,1\}^{i-1}$ be a subset satisfying $|\mathcal{U}_{i-1}| = L$ and assume that $Q_{i-1}(\tilde{u}_1^{i-1})$ is known for some $\tilde{u}_1^{i-1} \in \mathcal{U}_{i-1}$. Then, for $\tilde{u}_i \in \{0,1\}$, one can write

$$\begin{aligned} Q_i(\tilde{u}_1^i) &\propto \Pr\left(U_1^i = \tilde{u}_1^i, Y_1^N = y_1^N\right) \\ &\propto Q_{i-1}(\tilde{u}_1^{i-1}) \Pr\left(U_i = \tilde{u}_i | Y_1^N = y_1^N, U_1^{i-1} = \tilde{u}_1^{i-1}\right), \quad (1) \end{aligned}$$

where the right-most term can be computed efficiently by the standard SC decoder starting from $Q_0(\tilde{u}_1^0) \triangleq 1$. This results in $Q_i(\tilde{u}_1^i)$ values for $2L$ partial sequences. Then, one prunes the list down to L sequences by keeping only most likely paths according to (1) for an SCL decoder with maximum list size L. Note that if u_i is frozen, then the decoder simply extends all paths with correct frozen bit. After the N-th decoding stage, the estimate $\hat{u}_1^N$ is chosen as the candidate maximizing the function $Q_N(\tilde{u}_1^N)$.

D. Successive Cancellation Inactivation Decoding

SC inactivation (SCI) decoding is a simplified version of SCL decoding for the BEC proposed in [13]. During SCL decoding, when an information bit is decoded to an erasure, it is replaced by both possible values (assuming the list size is large enough) and decoding proceeds separately under these two hypotheses. But, for linear codes on erasure channels, any uncertainty in the information bits takes the form of an affine subspace. For this reason, one can store a basis instead of listing all codewords, reducing the complexity significantly.

Thus, the SCI decoder follows the same decoding schedule as the SCL decoder but instead replaces an erased information bit by an unknown variable, i.e., the bit is *inactivated* [14]–[16]. Later, some inactivated bits may be resolved using linear equations derived from decoding frozen bits. This can be done at each stage of the decoding process or delayed until the end. When dummy variables are eliminated without delaying to the end of decoding, we refer to this as a *consolidation* event. In this work, our focus is on the SCI decoder with consolidations.

For the BEC, all partial information sequences $u_1^i \in \{0,1\}^i$ with $Q_i(u_1^i) > 0$ have the same probability. Hence, for a given u_1^{i-1}, we have $p_i(u_1^{i-1}) \in \{0, \frac{1}{2}, 1\}$. If $p_i(u_1^{i-1}) \in \{0,1\}$, then u_i is known perfectly at the receiver. However, if $p_i(u_1^{i-1}) = \frac{1}{2}$ and u_i is not frozen, then $\hat{u}_i$ can be seen as an erasure. Then, the SCI decoder inactivates u_i by introducing a dummy variable $\tilde{u}_i$ and storing the decision as $\hat{u}_i = \tilde{u}_i$. It then continues decoding using SC decoding for the BEC except that the message values are allowed to be a function of all inactivated variables. In the end, the inactivated bits are typically resolved using linear equations derived from decoding frozen bits.

The SCI decoder can inactivate multiple bits when required. If the maximum number of inactivations is not bounded, then this algorithm implements ML decoding [13].

III. Analysis of the List Decoders

An important property of list decoding is that, if the correct codeword is on the list at the end of decoding, then the error probability is upper bounded by that of the ML decoder. We focus on understanding how large the list should be at each stage so that the correct codeword is likely to be on it.

A. An Information-Theoretic Perspective

Consider a length-N polar code with SCL decoding after the first m input bits have been processed. Since SC decoding does not make use of future frozen bits, the idea is to focus on the subset of length-m input patterns that have significant conditional entropy given the channel observation. A straightforward but important insight is that, after observing Y_1^N, the uncertainty in U_1^m is quantified by the entropy

$$H\left(U_1^m | Y_1^N\right) = \sum_{i=1}^m H\left(U_i | U_1^{i-1}, Y_1^N\right) \quad (2)$$

where U_1^N is assumed to be uniform over $\{0,1\}^N$. This is exactly true if the first m bits are all information bits, i.e., if $[m] \subseteq \mathcal{A}$. If $[m]$ contains also frozen indices, however, then the situation is more complicated.

Let $\mathcal{A}^{(m)} \triangleq \mathcal{A} \cap [m]$ and $\mathcal{F}^{(m)} \triangleq \mathcal{F} \cap [m]$ be the sets containing information and frozen indices within the first m input bits, respectively. Now, consider an experiment where the frozen bits $U_{\mathcal{F}^{(m)}}$ are uniform and independent of U_1^{m-1}. Obviously, using (2) naively with the assumption that $U_{\mathcal{F}^{(m)}}$ is not known to the receiver would cause an overestimate of $H\left(U_1^m | Y_1^N\right)$ by an amount of at least $\sum_{i \in \mathcal{F}^{(m)}} H\left(U_i | U_1^{i-1}, Y_1^N\right)$. In addition to this, the frozen bits $U_{\mathcal{F}^{(m)}}$ may reveal additional information about the previous information bits.

To understand more about the uncertainty within the first m input bits during SCL decoding, we define the quantities $d_m \triangleq H\left(U_{\mathcal{A}^{(m)}} | Y_1^N, U_{\mathcal{F}^{(m)}}\right)$ and $\Delta_m \triangleq d_m - d_{m-1}$. Observe that, if U_m is an information bit, then we have

$$\begin{aligned}
\Delta_m &= H\left(U_{\mathcal{A}^{(m)}} | Y_1^N, U_{\mathcal{F}^{(m)}}\right) - H\left(U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&= H\left(U_{\mathcal{A}^{(m)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) - H\left(U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&= H\left(U_{\mathcal{A}^{(m)}}, U_{\mathcal{F}^{(m-1)}} | Y_1^N\right) - H\left(U_{\mathcal{A}^{(m-1)}}, U_{\mathcal{F}^{(m-1)}} | Y_1^N\right) \\
&= H\left(U_1^{m-1} | Y_1^N\right) + H\left(U_m | Y_1^N, U_1^{m-1}\right) - H\left(U_1^{m-1} | Y_1^N\right) \\
&= H(U_m; Y_1^N | U^{m-1}).
\end{aligned} \tag{3}$$

Notice that (3) is exactly what one would expect from the naive analysis given by (2).

If U_m is a frozen bit, then consider a model where it is not known to the receiver at that time of transmission.[1] Then, the act of revealing U_m to the receiver changes the conditional uncertainty about $U_{\mathcal{A}^{(m-1)}}$ by

$$\begin{aligned}
\Delta_m &= H\left(U_{\mathcal{A}^{(m)}} | Y_1^N, U_{\mathcal{F}^{(m)}}\right) - H\left(U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&= H\left(U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}, U_m\right) - H\left(U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&= -I\left(U_m; U_{\mathcal{A}^{(m-1)}} | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&= H\left(U_m | Y_1^N, U_1^{m-1}\right) - H\left(U_m | Y_1^N, U_{\mathcal{F}^{(m-1)}}\right) \\
&\geq H\left(U_m | Y_1^N, U_1^{m-1}\right) - 1.
\end{aligned} \tag{4}$$

This expression quantifies the effect of revealing the new frozen bit as a reduction in the conditional entropy of the information bits preceding it. A large reduction may occur when the channel $W_N^{(m)}$ has low entropy (i.e., a low-entropy effective channel is essentially frozen) and the reduction will be small if the channel entropy is high (i.e., the input is unpredictable from Y_1^N and U_1^{m-1}).

For BMS channels, we can combine (3) and (4) to understand the dynamics of d_m. This gives a proxy for the uncertainty in the SCL decoding after m steps. Thus, we have

$$\sum_{i \in \mathcal{A}^{(m)}} H\left(W_N^{(i)}\right) - \sum_{i \in \mathcal{F}^{(m)}} \left(1 - H\left(W_N^{(i)}\right)\right) \leq d_m \tag{5}$$

$$\leq \sum_{i \in \mathcal{A}^{(m)}} H\left(W_N^{(i)}\right). \tag{6}$$

We note that the lower bound neglects the possibility that (dynamic) frozen bits (even if perfectly observed) may not provide substantial information to reduce the entropy.

[1]This reflects how the SCL decoder operates, i.e., it does not use the knowledge of any frozen bit U_m until reaching the end of its decoding stage m. Then, the soft estimate $p_m(u_1^{m-1})$ provides an additional information to separate the hypotheses (i.e, paths) although the hard estimate is chosen as $\hat{u}_m = u_m$ independent of $p_m(u_1^{m-1})$.

Theorem 1. *Upon observing y_1^N when u_1^N is transmitted, the set of partial sequences $\tilde{u}_1^m$ more likely than the true sequence u_1^m after m stages of SCL decoding is given by $\mathcal{S}^{(m)}\left(u_1^m, y_1^N\right) \triangleq \{\tilde{u}_1^m : Q_m(\tilde{u}_1^m) \geq Q_m(u_1^m)\}$. On average, the logarithm of its cardinality is upper bounded by d_m, i.e,*

$$\mathrm{E}\left[\log_2 |\mathcal{S}^{(m)}|\right] \leq d_m = H\left(U_{\mathcal{A}^{(m)}} | Y_1^N, U_{\mathcal{F}^{(m)}}\right). \tag{7}$$

Proof. Assume, w.l.o.g., that u_1^N and y_1^N are transmitted and observed, respectively. Then, we have

$$\begin{aligned}
\log_2 |\mathcal{S}^{(m)}| &\overset{\text{(a)}}{=} \log_2 \sum_{\tilde{u}_1^m} \mathbb{1}_{\left(\Pr\left(\tilde{u}_{\mathcal{A}} | y_1^N, u_{\mathcal{F}}\right) \geq \Pr\left(u_{\mathcal{A}} | y_1^N, u_{\mathcal{F}}\right)\right)} \\
&\overset{\text{(b)}}{\leq} -\log_2 \Pr\left(u_{\mathcal{A}} | y_1^N, u_{\mathcal{F}}\right)
\end{aligned}$$

where (a) follows from $Q_m(u_1^m) \propto \Pr\left(U_1^i = \tilde{u}_1^i, Y_1^N = y_1^N\right)$ and Bayes' rule and (b) from the fact that if there are more than $\Pr\left(u_{\mathcal{A}} | y_1^N, u_{\mathcal{F}}\right)^{-1}$ sequences $\tilde{u}_{\mathcal{A}}$ with probability $\Pr\left(u_{\mathcal{A}} | y_1^N, u_{\mathcal{F}}\right)$, then the total probability exceeds 1. As the inequality is valid for any pair u_1^N and y_1^N, taking the expectation over all u_1^m and y_1^N yields the stated result. ☐

Using (7) and (6) yields an upper bound easier to calculate

$$\mathrm{E}\left[\log_2 |\mathcal{S}^{(m)}|\right] \leq \sum_{i \in \mathcal{A}^{(m)}} H\left(W_N^{(i)}\right). \tag{8}$$

Now, consider an SCL decoder whose list size is L_m during the m-th decoding step. Then, the decoder should satisfy $L_m \geq |\mathcal{S}^{(m)}|$ for the true u_1^m to be in the set $\mathcal{S}^{(m)}$.

Remark 1. *It is worth noting that the analysis in terms of $\log_2 L_m$ has two weaknesses. First, the entropy really only characterizes typical events (e.g., ensuring that the correct codeword stays on the list at least half of the time) whereas coding focuses on much rarer events (e.g., block error rates less than 10^{-2}). Second, even if entropy is the right quantity, the sequence d_m is averaged over Y_1^N but the actual decoder sees a random realization $H\left(U_{\mathcal{A}^{(m)}} | Y_1^N = y_1^N, U_{\mathcal{F}^{(m)}}\right)$. Regardless, we believe that these results provide an initial step towards a theoretical analysis of the SCL decoder. In addition, the numerical results illustrates the accuracy of the analysis.*

Remark 2. *These results also have some significance for code design. To achieve good performance with an SCL decoder whose list size is L_m during the m-th decoding step, a reasonable first-order design criterion is that $\log_2 L_m \geq d_m$. This observation implies, in principle, that frozen bits should be allocated to prevent d_m from exceeding $\log_2 L_m$.*

B. The Binary Erasure Channel

For the BEC, the SCI decoder provides a concrete example of this information-theoretic perspective. In this case, the set of valid information sequences after m decoding steps is an affine subspace of $\{0,1\}^K$. For a fixed realization y_1^N, the subspace dimension is $d_m(y_1^N) = H\left(U_{\mathcal{A}^{(m)}} | Y_1^N = y_1^N, U_{\mathcal{F}^{(m)}}\right)$. Let $D_m = d_m(Y_1^N)$ denote corresponding RV. Our goal is to understand the evolution of this random sequence.

Let $\epsilon_N^{(m)} \triangleq \Pr\left(p_i(u_1^{i-1}) = \frac{1}{2}\right)$, where the implied randomness is due to the received vector. Consider the decoding of information and frozen bits given the observed vector and preceding frozen bits. When an information bit u_m is decoded, one of following events occurs:

- The information bit is decoded as an erasure and the subspace dimension increases by one, i.e., $d_m(y_1^N) = d_{m-1}(y_1^N) + 1$. Averaged over all y_1^N, the probability of this event equals $\epsilon_N^{(m)}$.
- The information bit is decoded as an affine function of the previous information bits and the subspace dimension is unchanged, i.e., $d_m(y_1^N) = d_{m-1}(y_1^N)$. Averaged over all y_1^N, the probability of this event equals $1 - \epsilon_N^{(m)}$.

If a frozen u_m is decoded, one of following events occurs:

- The decoder returns an erasure for the frozen bit. In this case, revealing the true value of the frozen bit allows decoding to continue, but no new information is provided about preceding information bits. Thus, we have $d_m(y_1^N) = d_{m-1}(y_1^N)$. Averaged over all y_1^N, the probability of this event equals $1 - \epsilon_N^{(m)}$.
- The frozen bit is decoded as an affine function of the previous information bits. Averaged over all y_1^N, the probability of this event equals $1 - \epsilon_N^{(m)}$. In this case, revealing the true value of the frozen bit gives a linear equation for a subset of the preceding information bits. If the linear equation is informative, then the subspace dimension decreases by one via a consolidation event, i.e., we have $d_m(y_1^N) = d_{m-1}(y_1^N) - 1$. Otherwise, the dimension is unchanged, i.e., $d_m(y_1^N) = d_{m-1}(y_1^N)$.

At first glance, these rules might appear to tell the whole story. But, the erasure rate $\epsilon_N^{(m)}$ is averaged over all y_1^N whereas predicting the value of D_m requires knowing the conditional probability of erasure events given all past observations. More importantly, to understand consolidation events, one needs to compute the probability that the obtained equation is informative.

Since we do not have expressions for these quantities,[2] we use two simplifying approximations. First, we approximate the probability of decoding an erasure for a frozen bit as independent of all past events. Second, we approximate the probability that an informative equation obtained from consolidation by $1 - 2^{-D_{m-1}}$, independent of past events. This value comes from modeling the obtained equation and the subset using a uniform random model. Under these assumptions, the random sequence $D_1, \ldots, D_N$ can be approximated by an inhomogeneous Markov chain with transition probabilities $P_{i,j}^{(m)} \approx \Pr\left(D_m = j \,|\, D_{m-1} = i\right)$ where

$$P_{i,j}^{(m)} = \begin{cases} \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i+1 \\ 1 - \epsilon_N^{(m)} & \text{if } m \in \mathcal{A},\ j = i \\ \epsilon_N^{(m)} + \left(1 - \epsilon_N^{(m)}\right) 2^{-D_{m-1}} & \text{if } m \in \mathcal{F},\ j = i \\ \left(1 - \epsilon_N^{(m)}\right)\left(1 - 2^{-D_{m-1}}\right) & \text{if } m \in \mathcal{F},\ j = i-1. \end{cases} \tag{9}$$

Based on this Markov chain approximation, one can make a further approximation by computing the expectation

[2]Even if we had them exactly, they may be too complicated to be useful.

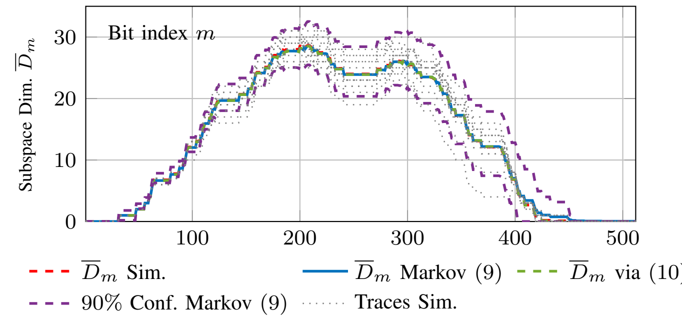

Fig. 1. $\overline{D}_m$ vs. m for randomly permuted 246 erasures.

$\overline{D}_m \triangleq \mathbb{E}\left[D_m\right]$ and approximating $\mathbb{E}\left[2^{-D_m}\right] \approx 2^{-\overline{D}_m}$. By setting $\overline{D}_0 \triangleq 0$, this gives the simple recursive approximation

$$\overline{D}_m \approx \begin{cases} \overline{D}_{m-1} + \epsilon_N^{(m)} & \text{if } m \in \mathcal{A} \\ \left[\overline{D}_{m-1} - \left(1 - 2^{-\overline{D}_{m-1}}\right)\left(1 - \epsilon_N^{(m)}\right)\right]^+ & \text{if } m \in \mathcal{F} \end{cases} \tag{10}$$

where $[\cdot]^+ \triangleq \max\{0, \cdot\}$.

IV. Simulation Results

In the following, the simulation results are provided for some constructions with dynamic frozen bits. In particular, we consider a modified RM code (called a d-RM code) [13], where each frozen bit after the first information bit is set to a random linear combination of preceding information bits.

Recently, Arıkan introduced polarization-adjusted convolutional (PAC) codes [10], which can be represented as a polar code with dynamic frozen bits [17], [18]. However, the rate-profiling choice of a PAC code is directly reflected in the frozen index set of its polar code representation [17]. Thus, if $\mathcal{A}$ of an RM code is chosen as the rate-profiling, then the frozen index set of the PAC code becomes the same as that of a d-RM code. They differ in the dynamic frozen bit constraints.

A. The Binary Erasure Channel

In order to understand the accuracy of the analysis and approximations presented above, we have simulated the SCI decoder with consolidation. The results of these simulations are realizations of the random process $D_1, \ldots, D_N$.

One weakness of these bounds is that the channel variation (e.g., in the number of erasures) significantly increases the variation in D_1^N. Thus, in order to highlight the similarity between the theory and simulation, we use a fixed-weight BEC that chooses a random pattern with exactly $\texttt{round}(N\epsilon)$ erasures. To motivate this, we note that density evolution naturally captures the typical behavior of the analyzed system [19]. Fig. 1 shows simulation results for realizations of D_1^N and compares these with their average and the theoretical predictions (9) and (10). These results show that, for a $(512, 256)$ d-RM code, the simulation mean is quite close to the analysis. The 15 random simulation traces also lie largely within the 90% confidence range of the Markov chain analysis.

B. The Binary-Input Additive White Gaussian Channel

Fig. 2 shows simulation results for a $(128, 64)$ d-RM code and a novel design (based on suggestions in Remark 2) under

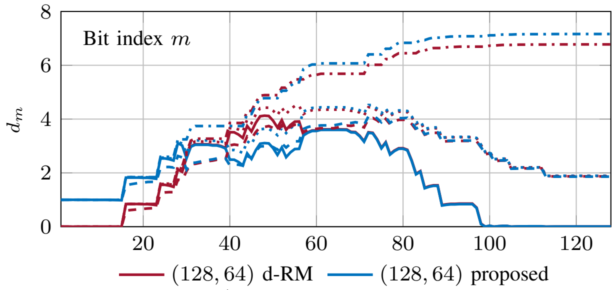

Fig. 2. d_m vs. m at $E_b/N_0 = 0.5$ dB (dash-dotted: upper bound (6), solid: lower bound (5), dotted: d_m via simulation, dashed: $\mathrm{E}[\log_2 |\mathcal{S}^{(m)}|]$ via simulation).

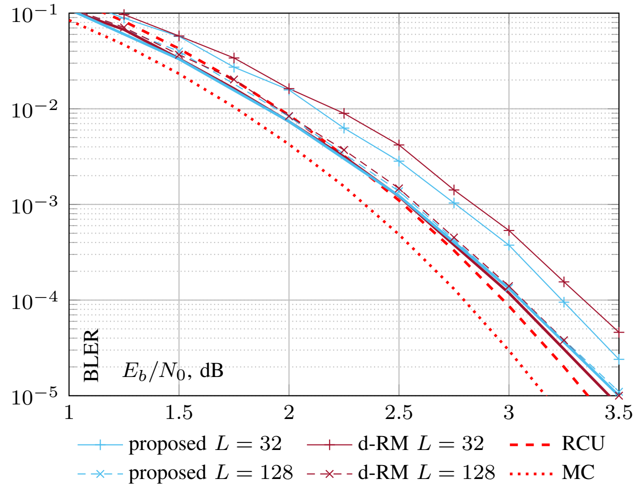

Fig. 3. BLER vs. SNR for (128, 64) codes (solid: ML LB).

SCL decoding with $L = 2^{14}$ and $E_b/N_0 = 0.5$ together with the upper and lower bounds (6) and (5) on d_m. The proposed code takes the set $\mathcal{A}$ of the $(128, 64)$ RM code and obtains a new set as $\mathcal{A}' = (\mathcal{A} \setminus \{30, 40\}) \cup \{1, 57\}$, i.e., $u_{\{30,40\}}$ are frozen and $u_{\{1,57\}}$ are unfrozen, where each frozen bit is still set to a random linear combination of preceding information bit(s). This helps especially for the considered list size $L = 32$. The reason is illustrated by the lower bounds on d_m in Fig. 2. In addition to having a smaller peak value, this peak occurs for the proposed design later than for the d-RM code. This helps for the proposed code to keep the correct path in the list towards the end for small list sizes, e.g., $L = 32$. If the list size is further decreased, then having u_1 as information bit can cause a degradation. Notice that there is no visible degradation in the ML performance. Fig. 2 validates the bounds (5), (6) and (8).[3] Also, (8) is tight for the entire range and (5) closely tracks the simulation for $m \leq 50$.

Fig. 3 compares the performance of the d-RM and proposed codes. When an SCL decoder with $L = 128$ is considered, both codes perform within 0.15 dB of the random coding union (RCU) bound [20, Thm. 16] at a block error rate of 10^{-5}. When a smaller list size, e.g., $L = 32$, is adopted, the proposed code outperforms the d-RM code especially at higher SNR values. This validates the analysis illustrated in Fig. 2. The metaconverse (MC) bound [20, Thm. 28] is also provided.

[3] To provide a robust estimate of $\mathrm{E}[\log_2 |\mathcal{S}^{(m)}|]$, the threshold for inclusion is reduced to $\alpha Q_m(u_1^m)$ where $\alpha < 1$ is a constant, e.g., $\alpha = 0.94$ in Fig. 2.

V. Conclusion

In this paper, we consider the theoretical question "What list size is sufficient to achieve maximum-likelihood (ML) decoding performance under an SCL decoder?". Our results identify information-theoretic quantities associated with the required list size and also lead to a natural approximation that can be computed efficiently even for very long codes.

Simulation results show that this approximation captures the dynamics of the required list size at each stage of decoding on the BEC. For general BMS channels, e.g., biAWGNC, the analysis identified the key quantity d_m as a proxy for the uncertainty in the SCL decoding. Insight from the analysis resulted in the proposed code with improved performance under SCL decoding with list size of 32.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051–3073, Jul. 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.

[3] M. C. Coşkun, G. Durisi, T. Jerkovits, G. Liva, W. Ryan, B. Stein, and F. Steiner, "Efficient error-correcting codes in the short blocklength regime," *Elsevier Phys. Commun.*, vol. 34, pp. 66–79, Jun. 2019.

[4] P. Trifonov and V. Miloslavskaya, "Polar subcodes," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 2, pp. 254–266, Feb. 2016.

[5] P. Trifonov and G. Trofimiuk, "A randomized construction of polar subcodes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2017, pp. 1863–1867.

[6] P. Yuan, T. Prinz, G. Böcherer, O. İşcan, R. Böhnke, and W. Xu, "Polar code construction for list decoding," in *Proc. 11th Int. ITG Conf. on Syst., Commun. and Coding (SCC)*, Feb. 2019, pp. 125–130.

[7] A. Elkelesh, M. Ebada, S. Cammerer, and S. ten Brink, "Decoder-tailored polar code design using the genetic algorithm," *IEEE Trans. Commun.*, vol. 67, no. 7, pp. 4521–4534, 2019.

[8] A. Fazeli, A. Vardy, and H. Yao, "Convolutional decoding of polar codes," in *IEEE Int. Symp. on Inf. Theory*, 2019, pp. 1397–1401.

[9] M. Rowshan and E. Viterbo, "How to modify polar codes for list decoding," in *IEEE Int. Symp. Inf. Theory*, Jul. 2019, pp. 1772–1776.

[10] E. Arıkan, "From sequential decoding to channel polarization and back again," *CoRR*, vol. abs/1908.09594, 2019. [Online]. Available: http://arxiv.org/abs/1908.09594

[11] I. Dumer and K. Shabunov, "Near-optimum decoding for subcodes of Reed-Muller codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2001, p. 329.

[12] N. Stolte, "Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung," Ph.D. dissertation, TU Darmstadt, 2002.

[13] M. C. Coşkun, J. Neu, and H. D. Pfister, "Successive cancellation inactivation decoding for modified Reed-Muller and eBCH codes," *IEEE Int. Symp. Inf. Theory, accepted*, 2020. [Online]. Available: http://arxiv.org/abs/2004.05969

[14] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.

[15] C. Measson, A. Montanari, and R. Urbanke, "Maxwell construction: The hidden bridge between iterative and maximum a posteriori decoding," *IEEE Trans. Inf. Theory*, vol. 54, no. 12, pp. 5277–5307, Dec. 2008.

[16] A. Eslami and H. Pishro-Nik, "On bit error rate performance of polar codes in finite regime," in *Proc. 48th Annu. Allerton Conf. on Commun., Control, and Comput.*, Sep. 2010, pp. 188–194.

[17] H. Yao, A. Fazeli, and A. Vardy, "List decoding of arıkan's PAC codes," *CoRR*, vol. abs/2005.13711, 2020. [Online]. Available: http://arxiv.org/abs/2005.13711

[18] M. Rowshan, A. Burg, and E. Viterbo, "Polarization-adjusted convolutional (PAC) codes: Fano decoding vs list decoding," *CoRR*, vol. abs/2002.06805, 2020. [Online]. Available: http://arxiv.org/abs/2002.06805

[19] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008.

[20] Y. Polyanskiy, V. Poor, and S. Verdù, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–235, May 2010.