Polar Codes: Robustness of the Successive Cancellation Decoder with Respect to Quantization

S. Hamed Hassani and Rüdiger Urbanke

***Abstract*—Polar codes provably achieve the capacity of a wide array of channels under successive decoding. This assumes infinite precision arithmetic. Given the successive nature of the decoding algorithm, one might worry about the sensitivity of the performance to the precision of the computation.**

We show that even very coarsely quantized decoding algorithms can lead to excellent performance. More concretely, we show that under successive decoding with an alphabet of cardinality only three, the decoder still has a threshold and this threshold is a sizable fraction of capacity. More generally, we show that if we are willing to transmit at a rate δ below capacity, then we need only $c\log(1/\delta)$ bits of precision, where c is a universal constant.

I. Introduction

Polar coding schemes provably achieve the capacity of several classes of channels including binary memoryless symmetric (BMS) channels. Since the invention of polar codes by Arikan, [1], a large body of work has been done to investigate the pros and cons of polar codes in different practical scenarios. In [3], the authors propose methods to compute the compound capacity of polar codes, decoded under the successive cancellation (SC) decoder, over a given set of BMS channels and show that polar codes are not universal. In [5] and [6], given a desired probability of error, the trade-off between the maximum achievable rate and block-length is considered. In [7], [8] and [9], efficient constructions of polar codes are considered. Recently, in [11] the authors generalize the successive cancellation decoder to a proper successive list decoder and report that with such a decoder the error probability for short block-lengths is considerably improved (at the cost of an increase in complexity proportional to list size).

We address one further aspect of polar codes using successive decoding. We ask whether such a coding scheme is *robust*. More precisely, the standard analysis of polar codes under successive decoding assumes infinite precision arithmetic. Given the successive nature of the decoder, one might worry how well such a scheme performs under a finite precision decoder. A priori it is not clear whether such a coding scheme still shows any threshold behavior and, even if it does, how the behavior scales in the number of bits of the decoder.

We show that in fact polar coding is extremely robust with respect to the quantization of the decoder. In Figure 1, we show the achievable rate using a simple successive decoder with only three messages, called the decoder with erasures, when transmission takes place over several important channel families. As one can see from this figure, in particular for channels with high capacity, the fraction of the capacity that is achieved by this simple decoder is close to 1, i.e., even this extremely simple decoder almost achieves capacity. We further show that, more generally, if we want to achieve a rate δ below capacity ($\delta > 0$), then we need at most $c\log(1/\delta)$ bits of precision.

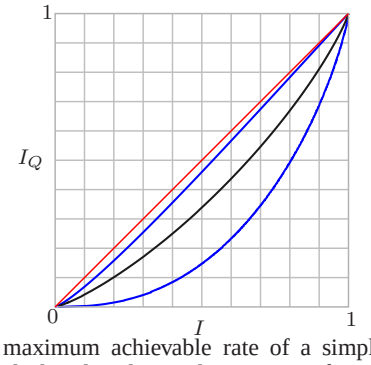

Fig. 1. The maximum achievable rate of a simple three message decoder, called the decoder with erasures, for different channel families. From top to bottom: the first curve corresponds to the family of binary erasure channels (BEC) where the decoder with erasures is equivalent to the original SC decoder and, hence, the maximum achievable rate is the capacity itself. The second curve corresponds to the family of binary symmetric channels (BSC). The third curve corresponds to the family of binary additive white Gaussian channels (BAWGN). The curve at the bottom corresponds to a universal lower bound on the achievable rate by the decoder with erasures.

The significance of our observations goes beyond the pure computational complexity which is required. The main bottleneck in the implementation of large high speed coding systems is typically memory. Therefore, if one can find decoders which work with only a few bits per message then this can make the difference whether a coding scheme can be implemented or not.

The outline of the paper is as follows. Section II gives a brief review of polar codes and successive decoding. In Section III we review an equivalent model which will form the basis for all of our analysis. Section IV contains the main statements of the paper which are proved in Section V and Section VI . Section VII concludes the paper.

II. Polar Codes

A. Basic setting and definitions

Let $W : \mathcal{X} \to \mathcal{Y}$ be a BMS channel, where $\mathcal{X} = \{0, 1\}$. Let $I(W) \in [0, 1]$ denote the mutual information between the input and output of W with uniform distribution on the inputs. We call this the symmetric mutual information. Since

EPFL, School of Computer & Communication Sciences, Lausanne, CH-1015, Switzerland, {seyedhamed.hassani, rudiger.urbanke}@epfl.ch.

we assumed W to be symmetric, $I(W)$ is in fact the capacity of W.

Let $G_2 = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. The generator matrix of polar codes is defined through the Kronecker powers of G_2, denoted by $G_N = G_2^{\otimes n}$. Throughout the paper, the variables N and n are related as $N = 2^n$. Let us quickly review how the generator matrix of polar codes is constructed. Consider the $N \times N$ matrix G_N and let us label the rows of the matrix G_N from top to bottom by $0, 1, \cdots, N-1$. Now assume that we desire to transmit binary data over the channel W at rate $R < I(W)$ with block-length N. One way to accomplish this is to choose a subset $\mathcal{I} \subseteq \{0, \cdots, N-1\}$ of size NR and to construct a vector $U_0^{N-1} = (U_0, \cdots, U_{N-1})$ in a way that it contains our NR bits of data at positions in $\mathcal{I}$ and contains, at positions not in $\mathcal{I}$, some fixed value (for example 0) which is known to both the encoder and decoder. We then send the codeword $X_0^{N-1} = U_0^{N-1} G_N$ through the channel W. We refer to the set $\mathcal{I}$ as the set of *chosen indices* or *information indices* and the set $\mathcal{I}^c$ is called the set of *frozen indices*. The choice of these indices is specific to the channel W and in general for two different channels it is different. ([3]). The chosen indices of the channel W are identified by using the following procedure on each of the indices $i \in \{0, \cdots, N-1\}$. Let u_0^{N-1} be a randomly and uniformly chosen vector from $\{0,1\}^N$ and let y_0^{N-1} be the result of transmitting the vector $x_0^{N-1} = u_0^{N-1} G_N$ through N parallel copies of W. Assume that we want to estimate the value of u_i (denoted by $\hat{u}_i$) given the received output y_0^{N-1} and the values of the previous bits $u_0, \cdots, u_{i-1}$. The optimal decision in this regard is to compute the probabilities $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 0)$ and $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 1)$ and to decide on the value of u_i by comparing the probabilities. These probabilities define a BMS channel between u_i and the "observation" (y_0^{N-1}, u_0^{i-1}) which is denoted by $W_N^{(i)} : \{0,1\} \to \mathcal{Y}^N \times \{0,1\}^{i-1}$ and whose law is given by

$$W_N^{(i)}(y_0^{N-1}, u_0^{i-1} \,|\, u_i) = \frac{1}{2^{N-1}} \sum_{u_{i+1}^{N-1}} \prod_{j=0}^{N-1} W(y_j \,|\, (u_0^{N-1} G_N)_j). \tag{1}$$

It is easy to see that given (y_0^{N-1}, u_0^{i-1}), we can decode u_i very reliably if and only if the channel $W_N^{(i)}$ is very close to being noise-less (i.e., its capacity is very close to 1). A crucial fact here is that the channels $\{W_N^{(i)}\}$ have the property that as n grows large, a fraction of $I(W)$ of them tend to become noise-less (i.e., have capacity close to 1) and a fraction of $1 - I(W)$ of them tend to become completely noisy (i.e., have capacity close to 0). As a result, given a rate $R < I(W)$, a natural way to choose the information indices is to choose the NR indices such as i that their corresponding channel $W_N^{(i)}$ has the largest capacity. At the decoder, the bits $u_0, \cdots, u_{N-1}$ are decoded one by one. That is, the bit u_i is decoded after $u_0, \cdots u_{i-1}$. If i is a frozen index, its value is known to the decoder. If not, using the output y_0^{N-1} and the estimates of $u_0, \cdots, u_{i-1}$, the decoder computes the log-likelihood ratio (llr) $\log \frac{p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 0)}{p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 1)}$ and decides the value of u_i hardly. It can be shown that by a clever exploitation of the structure of G_N, one can estimate the llr's for all the information bits in time $N(\log N + 1)$. For the sake of briefness, we do not fully describe the functionality of the SC decoder and refer to [1] for a detailed description.

III. Quantized SC Decoder

Let $\mathbb{R}^* = \mathbb{R} \cup \{\pm\infty\}$ and consider a function $Q(x) : \mathbb{R}^* \to \mathbb{R}^*$ that is *symmetric* (i.e., $Q(x) = Q(-x)$). We define the Q-quantized SC decoder as a decoder in which the function Q is applied to the output of any computation that the decoder does. We denote such a decoder by SCD_Q. More precisely, the decoder SCD_Q computes the log-likelihoods of the received symbols from the channel and applies the function Q to them. These new numbers are then fed into the SC algorithm to estimate further messages. However, after computing every new message, the function Q is applied and the new quantized message is used for further computations. Finally, the value of the i-th bit, if not frozen, is decided according to the sign of its corresponding computed message. If positive, $\hat{u}_i = 0$, if 0, the value of $\hat{u}_i$ is decided by flipping a fair coin, and if negative, $\hat{u}_i = 1$.

Typically, the purpose of the function Q is to model the case where we only have finite precision in our computations perhaps due to limited available memory or due to other hardware limitations. Hence, the computations are correct within a certain level of accuracy which the function Q models. Thus, let us assume that the range of Q is a finite set $\mathcal{Q}$ with cardinality $|\,\mathcal{Q}\,|$. As a result, all the messages passed through the decoder SCD_Q belong to the set $\mathcal{Q}$.

In this paper we consider a simple choice of the function Q that is specified by two parameters: The distance between levels Δ, and truncation threshold M. Given a specific choice of M and Δ, we define Q as follows:

$$Q(x) = \begin{cases} \lfloor \frac{x}{\Delta} + \frac{1}{2} \rfloor \Delta, & x \in (0, M], \\ \lceil \frac{x}{\Delta} - \frac{1}{2} \rceil \Delta, & x \in [-M, 0), \\ \text{sign}(x) M, & \text{otherwise.} \end{cases} \tag{2}$$

Note here that $|\,\mathcal{Q}\,| = 1 + \frac{2M}{\Delta}$. A graphical illustration of Q is given in Fig. 2.

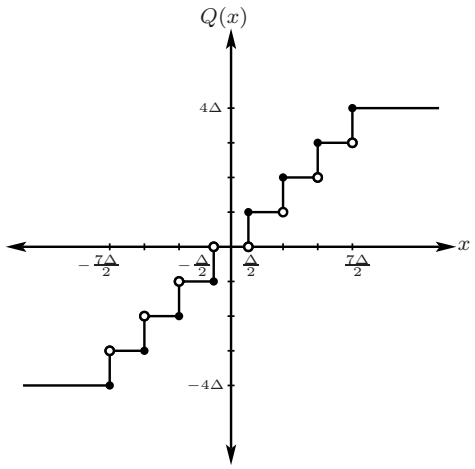

Fig. 2. The function $Q(x)$ for $|\,Q\,| = 9$ and $M = 4\Delta$.

IV. Main Statement

Theorem 1 (Main Statement): Consider transmission over a BMS channel W using polar codes and a SCD_Q with message alphabet $\mathcal{Q}$.

- For $|\mathcal{Q}| = 3$, we provide methods to precisely compute the maximum rate that can be achieved reliably when the transmission takes place over W and we use polar codes with the decoding algorithm SCD_Q. In particular, such maximum rates are plotted for different channel families in Figure 1. Also, in Figure 1 a universal lower bound for the maximum achievable rate is given. The methods used here are extendable to other quantized decoders.
- We can achieve up to an additive gap δ, $\delta > 0$, below the capacity $I(W)$ with $\log|\mathcal{Q}| \leq c\log(1/\delta)$.

Discussion: In short, polar codes are very robust to quantization within the decoder. In particular for BMS channels with capacity close to 1, very little is lost by quantizing. And as we discussed in the introduction, a reduction of the message alphabet can be crucial for the hardware implementation of such schemes.

Our proof strategy is the following. We describe a general framework of how to analyze the asymptotic performance of quantized decoders. We first apply our general framework to the so-called decoder with erasures. This decoder has a message alphabet of size 3. As we will see, this decoder achieves the fraction indicated in Figure 1. We then describe a general family of quantized decoders and prove how its performance scales.

V. General Framework for Analysis

A. Equivalent tree channel model and analysis of the probability of error

Since we are dealing with a linear code, a symmetric channel and symmetric decoders throughout this paper, without loss of generality we confine ourselves to the *all-zero codeword* (i.e., we assume that all the u_i's are equal to 0)[1]. In order to better visualize the decoding process, the following definition is handy.

Definition 2 (Tree Channels of Height n): For each $i \in \{0, 1, \cdots, N-1\}$, we introduce the notion of the i-th tree channel of height n which is denoted by $T(i)$. Let $b_1 \dots b_n$ be the n-bit binary expansion of i. E.g., we have for $n = 3$, $0 = 000$, $1 = 001$, …, $7 = 111$. With a slight abuse of notation we use i and $b_1 \cdots b_n$ interchangeably. Note that for our purpose it is slightly more convenient to denote the least (most) significant bit as b_n (b_1). Each tree channel consists of $n+1$ levels, namely $0, \dots, n$. It is a complete binary tree. The root is at level n. At level j we have 2^{n-j} nodes. For $1 \leq j \leq n$, if $b_j = 0$ then all nodes on level j are check nodes; if $b_j = 1$ then all nodes on level j are variable nodes. Finally, we give a label for each node in the tree $T(i)$: For each level j, we label the 2^{n-j} nodes at this level respectively from left to right by $(j,0), (j,1), \cdots, (j, 2^{n-j}-1)$.

[1]In terms of the analysis of the probability of error, it must be noted that since the we are dealing with a symmetric channel and a symmetric decoder, for any codeword the average error probability is the same as the average error probability for the all-zero error codeword ([12, Chapter 4])

All nodes at level 0 correspond to independent observations of the output of the channel W, assuming that the input is 0. In other words, assuming that the all-zero codeword has been transmitted, the N independent observations that result from passing each of the N codebits through W are fed into the bottom of $T(i)$ for further processing.

An example for $T(3)$ (that is $n = 3$, $b = 011$ and $i = 3$) is shown in Fig. 3.

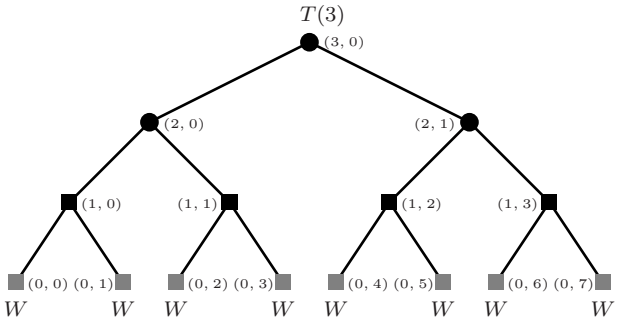

Fig. 3. Tree representation of the tree-channel $T(3)$ (W^{011}). The 3-bit binary expansion of 3 is $b_1 b_2 b_3 = 011$ (note that b_1 is the most significant bit). The pair beside each node is the label assigned to it.

Given the channel output vector y_0^{N-1} and assuming that the values of the bits prior to u_i are given, i.e., $u_0 = 0, \cdots, u_{i-1} = 0$, we now compute the probabilities $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 0)$ and $p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 1)$ via a simple message passing procedure on the equivalent tree channel $T(i)$. We attach to each node in $T(i)$ with label (j,k) a message[2] $m_{j,k}$ and we update the messages as we go up towards the root node. We start with initializing the messages at the leaf nodes of $T(i)$. For this purpose, it is convenient to represent the channel in the log-likelihood domain; i.e., for the node with label $(0,k)$ at the bottom of the tree which corresponds to an independent realization of W, we plug in the log-likelihood ratio (llr) $\log(\frac{W(y_k \,|\, 0)}{W(y_k \,|\, 1)})$ as the initial message $m_{0,k}$. That is,

$$m_{0,k} = \log(\frac{W(y_k \,|\, 0)}{W(y_k \,|\, 1)}). \tag{3}$$

Next, the SC decoder recursively computes the messages (llr's) at each level via the following operations: If the nodes at level j are variable nodes (i.e., $b_j = 1$), we have

$$m_{j,k} = m_{j-1,2k} + m_{j-1,2k+1}, \tag{4}$$

and if the nodes at level j are check nodes (i.e., $b_j = 0$), the message that is passed up is

$$m_{j,k} = 2\tanh^{-1}(\tanh(\frac{m_{j-1,2k}}{2})\tanh(\frac{m_{j-1,2k+1}}{2})). \tag{5}$$

In this way, it can be shown that ([1]) the message that we obtain at the root node is precisely the value

$$m_{n,0} = \log(\frac{p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 0)}{p(y_0^{N-1}, u_0^{i-1} \,|\, u_i = 1)}). \tag{6}$$

Given the description of $m_{n,0}$ in terms of a tree channel, it is now clear that we can use density evolution [12] to compute

[2]To simplify notation, we drop the dependency of the messages $m_{j,k}$ to the position i whenever it is clear from the context.

the the probability density function of $m_{n,0}$. In this regard, at each level j, the random variables $m_{j,k}$ are i.i.d. for $k \in \{0, 1, \cdots, 2^{n-j} - 1\}$. The distribution of the leaf messages $m_{0,k}$ is the distribution of the variable $\log(\frac{W(Y \mid 0)}{W(Y \mid 1)})$, where $Y \sim W(y \mid 0)$. One can recursively compute the distribution of $m_{j,k}$ in terms of the distribution of $m_{j-1,2k}, m_{j-1,2k+1}$ and the type of the nodes at level j (variable or check) by using the relations (4), (5) with the fact that the random variables $m_{j-1,2k}$ and $m_{j-1,2k+1}$ are i.i.d.

Finally, note that by the all-zero codeword assumption given the output y_0^{N-1} and the value of previous bits $u_0, \cdots, u_{i-1}$, the value of u_i is incorrectly decoded if either $m_{n,0} < 0$ or $m_{n,0} = 0$ and we choose the value of u_i to be 1 (This happens with probability $\frac{1}{2}$). Thus, denoting E_i as the event that we make and error on the i-th bit within the above setting, we obtain

$$\Pr(E_i) = \Pr(m_{n,0} < 0) + \frac{1}{2}\Pr(m_{n,0} = 0), \tag{7}$$

and the block error probability of polar codes using the information set $\mathcal{I}$ and SC decoder is upper bounded by

$$P_e \leq \sum_{i \in \mathcal{I}} \Pr(E_i). \tag{8}$$

B. Equivalent tree-channel model and quantized density evolution

An important point to note here is that with the decoder SCD_Q, the distribution of the messages in the trees $T(i)$ is different than the corresponding ones that result from the original SC decoder. Hence, the choice of the information indices is also specified by the choice of the function Q as well as the channel W. To be more precise, in order to analyze the error probability when we use the algorithm SCD_Q, one should note that since the function $Q(x)$ is a symmetric function around $x = 0$ and the channel W is also a BMS channel, the block error probability is equal to its value when we assume that the all-zero codeword has been sent. Similar to the analysis of the original SC decoder, we further assume that the codeword sent is the all-zero codeword and we fix the i-th bit and consider its equivalent tree-channel $T(i)$. Our objective is now to analyze the distribution of the messages in $T(i)$ assuming that the algorithm SCD_Q is performed and the previous bits $u_0, \cdots, u_{i-1}$ are decoded correctly (i.e., we know that all of them are 0).

For each label (j, k) in $T(i)$, let the random variable $\hat{m}_{j,k}$ represent the messages at this label. The messages $\hat{m}_{j,k}$ take their values in the discrete set $\mathcal{Q}$ (range of the function Q). At the leaf nodes of the tree we plug in the message

$$\hat{m}_{0,k} = Q(\log(\frac{W(y_k \mid 0)}{W(y_k \mid 1)})), \tag{9}$$

and the update equation for $\hat{m}_{(j,k)}$ is

$$\hat{m}_{j,k} = Q(\hat{m}_{j-1,2k} + \hat{m}_{j-1,2k+1}), \tag{10}$$

if the node (j, k) is a variable node and

$$\hat{m}_{j,k} = Q(2\tanh^{-1}(\tanh(\frac{\hat{m}_{j-1,2k}}{2})\tanh(\frac{\hat{m}_{j-1,2k+1}}{2}))), \tag{11}$$

if the node (j, k) is a check node. One can use the density evolution procedure to recursively obtain the densities of the messages $\hat{m}_{j,k}$.

Finally, let $\hat{E}_i$ denote the event that we make an error in decoding the i-th bit, with a further assumption that we have correctly decoded the previous bits $u_0, \cdots, u_{i-1}$. In a similar way as in the analysis of the original SC decoder, we get

$$\Pr(\hat{E}_i) = \Pr(\hat{m}_{n,0} < 0) + \frac{1}{2}\Pr(\hat{m}_{n,0} = 0). \tag{12}$$

Hence, one way to choose the information bits for the algorithm SCD_Q is to choose the bits u_i according to the least values of $\Pr(\hat{E}_i)$.

Note here that, since all of the densities takes their value in the finite alphabet $\mathcal{Q}$, the construction of such polar codes can be efficiently done in time $O(\mid \mathcal{Q} \mid^2 N \log N)$. We refer the reader to [1] to see how such a construction can be done.

C. Gallager Algorithm

Since our aim is to show that polar codes under successive decoding are robust against quantization, let us investigate an extreme case. The perhaps simplest message-passing type decoder one can envision is the Gallager algorithm. It works with single-bit messages. Does this simple decoder have a non-zero threshold? Unfortunately it does not, and this is easy to see.

We start with the equivalent tree-channel model. For each channel i of the polar code we have such a tree of height n and on each layer, nodes are either all check nodes or all variable nodes.

Since messages are only a single bit, the "state" of the decoder at level j can be described by a single non-negative number, namely the probability that the message at level j is incorrect. Assume that we transmit over a $\text{BSC}(p)$. Let $x_0 = p \in (0, \frac{1}{2})$. We are interested in the evolution of x_j. This evolution depends of course on the sequence of levels, i.e., it depends on which tree channel we are considering.

Assume that x_j is given and that the next level consists of check nodes. In this case the error probability increases. More precisely, $x_{j+1} = 2x_j(1 - x_j) > x_j$ when $x_j \in (0, \frac{1}{2})$. In other words, the state deteriorates. What happens if the next level consists of variable nodes instead? A little thought shows that in this case $x_{j+1} = x_j$, i.e., there is no change at all. This is true since if both incoming messages agree we can make a decision on the outgoing message, but if they differ we can only guess. This gives us $x_{j+1} = x_j^2 + x_j(1 - x_j) = x_j$.

Since in either case, the state either becomes worse or stays unchanged, no progress in the decoding is achieved, irrespective of the given tree. In other words, this decoder has a threshold of zero. As we have seen, the problem is the processing at the variable nodes since no progress is achieved there. But since we only have two incoming messages there is not much degree of freedom in the processing rules. It is doubtful if any message-passing decoder with only a single-bit message can do better.

D. 1-Bit Decoder with Erasures

Motivated by the previous example, let us now add one message to the alphabet of the Gallager decoder, i.e., we also add the possibility of having erasures to the above mentioned Gallager algorithm. In this regard, the function $Q(x)$ becomes the sign function, i.e.,

$$Q(x) = \begin{cases} \infty & x > 0, \\ 0 & x = 0, \\ -\infty & x < 0. \end{cases} \tag{13}$$

As a result, all messages passed by the algorithm SCD_Q take on only three possible values: $\{-\infty, 0, \infty\}$. In this regard, the decoding procedure takes a very simple form. The algorithm starts by quantizing the channel output to one of the three values in the set $\mathcal{Q} = \{-\infty, 0, \infty\}$. At a check node we take the product of the signs of the incoming messages and at a variable node we have the natural addition rule ($0 \leftarrow \infty + -\infty$, $0 \leftarrow 0 + 0$ and $\infty \leftarrow \infty + \infty, \infty \leftarrow \infty + 0$ and $-\infty \leftarrow -\infty + -\infty, -\infty \leftarrow -\infty + 0$). Note that on the binary erasure channel, this algorithm is equivalent to the original SC decoder.

Our objective is now to compute the maximum possible rate that the decoder SCD_Q can achieve reliably for a BMS channel W. We denote such quantity by $C(W, Q)$. The analysis is done in three steps:

1) The density evolution procedure: To analyze the performance of this algorithm, first note that since all our messages take their values in the set $\mathcal{Q}$, then all the random variables that we consider have the following form

$$D = \begin{cases} \infty & \text{w.p. } p, \\ 0 & \text{w.p. } e, \\ -\infty & \text{w.p. } m. \end{cases} \tag{14}$$

Here, the numbers p, e, m are probability values such that $p + e + m = 1$. Let us now see how the density evolves through the tree-channels. For this purpose, one should trace the output distribution of the relations (10) and (11) when the input messages are two i.i.d. copies of a r.v. D with pdf as in (14).

Lemma 3: Given two i.i.d. versions of a r.v. D with distribution as in (14), the output of a variable node operation (10), denoted by D^+, has the following form

$$D^+ = \begin{cases} \infty & \text{w.p. } p + 2pe, \\ 0 & \text{w.p. } e^2 + 2pm, \\ -\infty & \text{w.p. } m^2 + 2em. \end{cases} \tag{15}$$

Also, the check operation (11), yields D^- with the following law

$$D^- = \begin{cases} \infty & \text{w.p. } p^2 + m^2, \\ 0 & \text{w.p. } 1 - (1-e)^2, \\ -\infty & \text{w.p. } 2pm. \end{cases} \tag{16}$$

Proof: The proof follows by a straight forward computation of the corresponding probabilities for D^+ and D^-. As an example, let D_1, D_2 be two i.i.d. copies of D that are fed into the check operation (11). We know that with Q as in (13), the check operation is just multiplication of the signs. Hence, to have $D^- = \infty$, we should either have $D_1 = \infty, D_2 = \infty$ which occurs with probability p^2 or $D_1 = -\infty, D_2 = -\infty$ which occurs with probability m^2. Hence, D^- takes the value ∞ with probability $p^2 + m^2$. ∎

In order to compute the distribution of the messages $\hat{m}_{n,0}$ at a given level n, we use the method of [1] and define the polarization process D_n as follows. Consider the random variable $L(Y) = \log(\frac{W(Y\,|\,0)}{W(Y\,|\,1)})$, where $Y \sim W(y\,|\,0)$. The stochastic process D_n starts from the r.v. $D_0 = Q(L(Y))$ defined as

$$D_0 = \begin{cases} \infty & \text{w.p. } p = \Pr(L(Y) > 0), \\ 0 & \text{w.p. } e = \Pr(L(Y) = 0), \\ -\infty & \text{w.p. } m = \Pr(L(Y) < 0). \end{cases} \tag{17}$$

and for $n \geq 0$

$$D_{n+1} = \begin{cases} D_n^+ & ;\text{w.p. } \frac{1}{2}, \\ D_n^- & ;\text{w.p. } \frac{1}{2}, \end{cases} \tag{18}$$

where the plus and minus operations are given in (15), (16).

2) Analysis of the process D_n: Note that the output of process D_n is a itself a random variable of the form given in (14). Hence, we can equivalently represent the process D_n with a triple (m_n, e_n, p_n), where the coupled processes m_n, e_n and p_n are evolved using the relations (15) and (16) and we always have $m_n + e_n + p_n = 1$.

Following along the same lines as the analysis of the original SC decoder, we first claim that as n grows large, the process D_n will become polarized, i.e., the output of the process D_n will almost surely be a completely noiseless or a completely erasure channel.

Lemma 4: The sequence $\{D_n = (p_n, e_n, m_n), n \geq 0\}$ converges almost surely to a random variable D_∞ such that D_∞ takes its value in the set $\{(1,0,0), (0,1,0)\}$.

Proof: We first show that the process m_n is a super-martingale which converges a.s. to 0. From (15) and (16) we obtain,

$$\begin{aligned} \mathbb{E}[m_{n+1}\,|\,m_n] &= \frac{m_n^2 + 2m_n e_n + 2m_n p_n}{2} \\ &= m_n - \frac{m_n^2}{2} \leq m_n. \end{aligned}$$

As a result, since m_n is also bounded it converges a.s. to a limit r.v. m_∞. The a.s. convergence and boundedness of m_n also implies that

$$\mathbb{E}[m_{n+1} - m_n] = -\frac{1}{2}E[m_n^2] \to 0.$$

Hence, $m_n \to 0$ almost surely. In the same way, consider the process e_n. We have

$$\mathbb{E}[e_{n+1}\,|\,e_n] = e_n + 2p_n e_n. \tag{19}$$

Hence, the process e_n is a bounded sub-martingale which converges a.s. to a r.v. e_∞. This would imply that

$$\mathbb{E}[e_{n+1} - e_n] = 2\mathbb{E}[p_n e_n] \to 0.$$

Now, since $p_n = 1 - e_n - m_n$ and $m_n \to 0$, we get

$$\mathbb{E}[e_n(1 - e_n)] \to 0.$$

Thus, e_∞ ia either 0 or 1 and considering the fact that $m_\infty = 0$, the proof follows. ∎

We now aim to compute the value of $C(W,Q) = \Pr(D_\infty = (1,0,0))$, i.e., the ratio of the noiseless indices. The value of $\Pr(D_\infty = (1,0,0))$ is dependent on the starting channel D_0 that is given in (17) and is the highest rate that we can achieve with the 1-Bit Decoder with Erasures. In this regard, a convenient approach is to find a function $f : \mathcal{D} \to \mathbb{R}$ such that $f((0,1,0)) = 0$ and $f(0,0,1) = 1$ and for any $D \in \mathcal{D}$

$$\frac{1}{2}(f(D^+) + f(D^-)) = f(D).$$

With such a function f, the process $\{f(D_n)\}_{n\geq 0}$ is a martingale and consequently we have $\Pr(D_\infty = (1,0,0)) = f(D_0)$. Therefore, by computing the deterministic quantity $f(D_0)$ we obtain the value of $C(W,Q)$. However, finding a closed form for such a function seems to be a difficult task[3]. Instead, our idea is to look for alternative functions, denoted by $g : \mathcal{D} \to \mathbb{R}$, such that the process $g(D_n)$ is a super-martingale (sub-martingale) and hence we can get a sequence of upper (lower) bounds on the value of $\Pr(D_\infty = (1,0,0))$ as follows. Assume we have a function $g : \mathcal{D} \to \mathbb{R}$ such that $g((0,1,0)) = 0$ and $g(1,0,0) = 1$ and for any $D \in \mathcal{D}$,

$$\frac{1}{2}(g(D^+) + g(D^-)) \leq g(D). \tag{20}$$

Then, the process $\{g(D_n)\}_{n\geq 0}$ is a super-martingale and for $n \geq 0$ we have

$$\Pr(D_\infty = (1,0,0)) \leq \mathbb{E}[g(D_n)]. \tag{21}$$

The quantity $\mathbb{E}[g(D_n)]$ decreases by n and we have

$$\Pr(D_\infty = (1,0,0)) = \lim_{n\to\infty} \mathbb{E}[g(D_n)]. \tag{22}$$

In a similar way, on can search for a function $h : \mathcal{D} \to \mathbb{R}$ which $h((0,1,0)) = 0$ and $h(1,0,0) = 1$ and

$$\frac{1}{2}(h(D^+) + h(D^-)) \geq h(D). \tag{23}$$

Then $\{h(D_n)\}_{n\geq 0}$ is a sub-martingale, the quantities $\mathbb{E}[h(D_n)]$ are increasing with n, and

$$\Pr(D_\infty = (1,0,0)) = \lim_{n\to\infty} \mathbb{E}[h(D_n)]. \tag{24}$$

It remain to find some suitable candidates for g and h. It can be easily checked that one example for g is the function $g(D) = 1 - e$. To come up with more interesting examples, we first consider an equivalent representation of a generic density D that sometimes provides a good insight to choose candidates for g and h. A density D as in (14) can be equivalently represented as a simple BMS channel given in Fig. 4. This equivalence stems from the fact that for such a channel, conditioned on the event that the symbol $+1$ has been sent, the distribution of the output is precisely D. With a slight abuse of notation, we also denote the corresponding BMS channel by D. In particular, it is an easy exercise to show that the capacity ($I(D)$), Bhattacharyya parameter ($Z(D)$) and the error probability ($E(D)$) of the density D are given as

$$I(D) = (m+p)(1 - h_2(\frac{p}{p+m})), \tag{25}$$

[3]The function f clearly exists as one trivial candidate for it is $f(D) = \Pr(D_\infty = (1,0,0))$, where D_∞ is the limiting r.v. that the process $\{D_n\}_{n\geq 0}$ with starting value $D_0 = D$ converges to.

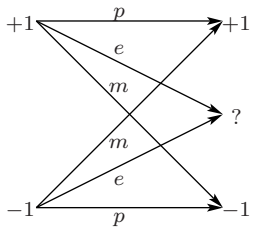

Fig. 4. The equivalent channel for the density D given in (14).

$$Z(D) = 2\sqrt{mp} + e, \tag{26}$$

$$E(D) = 1 - p - e/2, \tag{27}$$

where $h_2(\cdot)$ denotes the binary entropy function. We now show that another example of g is the function $g(D) = I(D)$, i.e., the capacity functional. We clearly have $I((1,0,0)) = 1$ and $I(0,1,0) = 0$. It is also easy to see that since the function Q is a not an injective function, we have

$$\frac{I(D^+) + I(D^-)}{2} \leq I(D).$$

We now find suitable candidates for the function h. We postpone the proof of the following lemma to the appendices.

Lemma 5: Define the function $h(D)$ as $h(D) = p - 4\sqrt{pm}$ for $D \in \mathcal{D}$. We have $h(D = (1,0,0)) = 1$, $h(D = (0,1,0)) = 0$ and

$$\frac{h(D^+) + h(D^-)}{2} \geq h(D). \tag{28}$$

Numerical experiments show that the functions $I(D)^2, (1 - Z(D))^2$ are also good candidates for h. However, an explicit proof of the fact that these functions satisfy the relation (23) may be a difficult task.

Given a BMS channel W, one can numerically compute $C(W,Q)$ with arbitrary accuracy δ: Consider the two functions $g(D) = I(D)$ and $h(D) = I(D)^2$. At time $n \in \mathbb{N}$, the process D_n given in (18) with staring density D_0 given in (17) has 2^n possible possible outputs of the form (14) with equal probability. Hence, the values $\mathbb{E}[g(D_n)]$ and $\mathbb{E}[h(D_n)]$ can be explicitly computed in time $O(2^n)$. Let $n \in \mathbb{N}$ be such that $\mathbb{E}[g(D_n)] - \mathbb{E}[h(D_n)] \leq \delta$. Since the value of $C(W,Q)$ is sandwiched between $\mathbb{E}[h(D_n)]$ and $\mathbb{E}[h(D_n)]$, then $\mathbb{E}[h(D_n)]$ provides a lower bound on $C(W,Q)$ which no further from it than δ. The curves in Figure 1 have been plotted with these considerations. Also, for a channel W with capacity $I(W)$ and error probability $E(W)$, we have

$$E(W) \leq \frac{1 - I(W)}{2}. \tag{29}$$

Therefore,

$$\inf_{D:E(D)=\frac{1-I(W)}{2}} C(D,Q) \leq C(W,Q), \tag{30}$$

and this leads to the universal lower bound obtained in Figure 1.

3) Scaling behavior and error exponent: In the last step, we need to show that for the rates below $C(W,Q)$ the block-error probability decays to 0 for large block-lengths.

Lemma 6: Let $D \in \mathcal{D}$. We have

$$Z(D^-) \leq 2Z(D), \tag{31}$$

$$Z(D^+) \leq 2(Z(D))^{\frac{3}{2}}. \tag{32}$$

Hence, for transmission rate $R < C(W,Q)$ and block-length $N = 2^n$, the probability of error of SCD_Q, denoted by $P_{e,Q}(N,R)$ satisfies $P_{e,Q}(N,R) = o(2^{-N^\beta})$ for $\beta < \frac{\log \frac{3}{2}}{2}$. Finally, we mention one major drawback of the 1-bit decoder with erasures and that is the fact the the speed of the polarization is further decreased compared to the original channel polarization process. As a result, by using the 1-bit decoder with erasures, we need to construct longer codes that polar codes with the original SC decoder. In Figure 5 we have plotted the block-error probability of polar codes of different block-lengths with the 1-bit decoder with erasures.

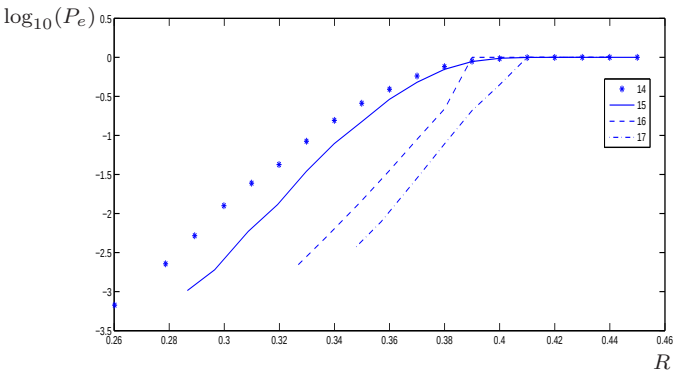

Fig. 5. Empirical value of the probability of error (P_e) in terms of rate (R) for the decoder with erasures. For top to bottom, the curves correspond to block-length 2^n with $n = 14, 15, 16, 17$. The transmission takes place over the BSC(0.11) which has capacity equal to $\frac{1}{2}$. For this channel, the decoder with erasures is capable of achieving at most the rate 0.46 with very large block-lengths (the value 0.46 has been computed by the methods given in Section V-D2).

We conclude this section by providing a lower bound on how fast the process $Z_{Q,n} = Z(D_n)$ polarizes.

Lemma 7: For $a, b \in (0,1)$, define the process

$$Y_n^{a,b} := Z_{Q,n}^a (1 - Z_{Q,n})^b. \tag{33}$$

We have

$$\mathbb{E}[Y_n^{a,b}] \leq \zeta_{a,b}^n, \tag{34}$$

where $\zeta_{a,b}$ is the given by,

$$\zeta_{a,b} = \sup_{D \in \mathcal{D}} \frac{Z(D^+)^a (1 - Z(D^+))^b + Z(D^-)^a (1 - Z(D^-))^b}{2Z(D)^a (1 - Z(D))^b}. \tag{35}$$

Here, $\mathcal{D}$ denotes the space of all the random variables that have the form as in (14). ■

Remark: Note that the optimization problem in (35) can be reformulated as a 2-dimensional optimization problem. Also, as an example for $a = b = \frac{3}{4}$ we have $\zeta_{a,b} = 0.9045$.

VI. Trade-off between the number of bits and the gap to capacity

In the previous section we have considered a particular family of decoders. We have seen that not only a small number of messages suffice to achieve a considerable fraction of capacity, but that by increasing the alphabet size this fraction quickly converges to capacity. Let us make this second observation precise now and prove the second part of Theorem 1. Consider a BMS channel W and assume that we need an algorithm SCD_Q such that is capable of achieving rates up to $I(W) - d$, where $d \leq \frac{1}{2}$ is a positive constant (for $d \geq \frac{1}{2}$ the 1-bit decoder with erasures is already a good choice). We first note that our ultimate goal is to find suitable parameters M and Δ so that the algorithms SCD_Q is capable of achieving a rate at least $I(W) - d$. We denote the maximum achievable rate of the algorithm SCD_Q by $C(W,Q)$. In order to compute $C(W,Q)$, we should precisely compute the ratio of the good indices among the set $\{0, 1, \cdots, N-1\}$ when N grows large. Here, we don't intend to compute the precise value of $C(W,Q)$ but to provide universal lower bound on $C(W,Q)$ that are already applicable for proving the theorem.

The proof consists of three steps. We first consider the original SC decoder and choose an integer n_d large enough so that for $n \geq n_d$, at least a fraction $I(W) - \frac{d}{2}$ of the sub-channels at level n have Bhattacharyya value less than e^{-2n}. More precisely, we have for $n \geq n_d$

$$\Pr(Z_n \leq e^{-2n}) \geq I(W) - \frac{d}{2}. \tag{36}$$

As a result, if we perform the original SC decoding, then at level n at least a fraction $I(W) - \frac{d}{2}$ of the sub-channels are very perfect. Let $\mathcal{I}_{n,d}$ denote the set of indices of these sub-channels. We now tune the parameters M and Δ for a decoder SCD_Q (with function Q given in (2)) in a way that the algorithm SCD_Q still decodes perfectly on the indices that belong to the set $\mathcal{I}_{n,d}$. Hence, in the first step, we fix $n \in \mathbb{N}$ and assume we are using a polar code of length 2^n. We intend to find candidates for M and Δ in terms of n so that the messages that we get by the algorithm SCD_Q with such candidates for M and Δ, are suitably close the their counterpart in the original SC decoder.

A. First step: How to choose M and Δ

Consider the i-th channel with its channel tree model. That is a binary tree with $n+1$ levels $0, 1, \cdots, n$ with 2^{n-j} nodes at the j-th level. The nodes are categorized into two types: variable nodes and check nodes. Also, depending on the value of i, all the nodes at each level are either variable of all are check nodes. Also, recall that for each node in $T(i)$ with label (j,k), we denote by $m_{j,k}$ the corresponding message that is passes by the original SC decoder and $\hat{m}_{j,k}$ denotes the corresponding messages of the SCD_Q algorithm. The primary problem that we consider here is as follows: Consider a specific realization of independent uses of the channel W at each of the leaves of the tree; By using the original SC decoder, this realization results in a specific value at the root node. Now, consider the same recursive computation process

with the following extra operations of the value that comes out of each computation:

1) After each of the computations we also perturb the resulting value by at most a fixed value Δ.
2) If the absolute value of the output is larger than a fixed value M, we replace the value by $\pm\infty$ according to its sign.

It is easy to see that the operations (1) and (2) are given to better analyze the algorithm SCD_Q. In this regard, how should we choose the values of M and Δ so that the final message that is computed at the top of the tree, i.e., $\hat{m}_{n,0}$ is not too far from its counterpart in the original SC decoder, i.e., $m_{n,0}$? First assume $M = \infty$. As a result, the operation (2) is not applied anymore. Straight forward computation shows that the partial derivatives of the functions $v(x, y)$ and $c(x, y)$, which correspond to (4) and (5) respectively, given by

$$v(x, y) := x + y, \tag{37}$$

$$c(x, y) := 2\tanh^{-1}(\tanh(\frac{x}{2})\tanh(\frac{y}{2})), \tag{38}$$

are always bounded above by 1. Hence, for $a, b \in \mathbb{R}$, we have

$$|\, v(x + a, x + b) - v(x, y)\,| \;\le\; |\, a\,| \;+\; |\, b\,|\,, \tag{39}$$

$$|\, c(x + a, x + b) - c(x, y)\,| \;\le\; |\, a\,| \;+\; |\, b\,|\,. \tag{40}$$

As a result, it is easy to see that assuming that only operation (1) is applied, the cumulative error that we get on the top of the tree $T(i)$ is upper bounded by $\Delta 2^{n+1}$. Hence, the following lemma follows.

Lemma 8: Consider a quantized SC algorithm in which $M = \infty$ (i.e., only operation (1) is applied). Also, consider the i-th position among the information bits with its corresponding binary tree $T(i)$. Then, for any realization of the channel outputs we have $|\, m_{j,k} - \hat{m}_{j,k}\,| \;\le\; 2^{j+1}\Delta$ for any label $(j, k) \in T(i)$. As a result, if we choose $\Delta \le 2^{-(n+1)}$, then $|\, m_{n,0} - \hat{m}_{n,0}\,| \le 1$.

Let us now assume that M is finite, hence the operation (2) is a non-trivial operation. Of course, depending on the value of M, the cumulative error varies in a large range. It seems that in this case providing worse case bounds as in Lemma 8 is a difficult task. Consequently, we seek for bounds that hold with high probability. We postpone the proof of the following lemma to the appendices.

Lemma 9: Let $M = 2n$ and $\Delta = 2^{-(n+1)}$. Then with probability at least $1 - 16(n + 1)(\frac{2}{e})^{2n}$, the following holds: If $\hat{m}_{n,0} \neq \infty$ then $|\, m_{n,0} - \hat{m}_{n,0}\,| \le 1$.

B. Second Step: What happens to the almost perfect channels

Let us now fix $n \ge n_d$ and consider the algorithm SCD_Q with parameters M and Δ as given in Lemma 9. In this step, we provide a lower bound on the value of $C(W, Q)$ which is equal to the final ratio of the good indices. In order to do this, we provide a lower bound only on the final ratio of the good indices that are branched out from the indices in the set $\mathcal{I}_{n,d}$. First, we consider the original SC decoder. By definition we have for each index $i \in \mathcal{I}_{n,d}$ that $Z(W_N^{(i)}) \le e^{-2n}$. Consider the tree-channel $T(i)$ and recall that the message that we get by the original SC decoder at its root note is denoted by $m_{n,0}$. Using the result of Lemma 11 in the appendices we obtain

$$\Pr(m_{n,0} \ge 2n + 1) \ge 1 - e^{1-n}. \tag{41}$$

Now, by using Lemma 9 and (41), at level n with probability at least $1 - e^{1-n} - 16(n + 1)(\frac{2}{e})^{2n} \ge 1 - 16(n + 2)(\frac{2}{e})^{2n}$, at an index $i \in \mathcal{I}_{n,d}$, the algorithm SCD_Q outputs the $+\infty$ message. This implies that at $i \in \mathcal{I}_{n,d}$ the distribution of the messages that we get by the algorithm SCD_Q stochastically dominates the following distribution

$$D = \begin{cases} \infty & \text{w.p. } 1 - 16(n + 2)(\frac{2}{e})^{2n}, \\ -\infty & \text{w.p. } 16(n + 2)(\frac{2}{e})^{2n}. \end{cases} \tag{42}$$

Now, let C_i be the final ratio of the perfect sub-channels that are branched from $i \in \mathcal{I}_{n,d}$. It is now easy to see that C_i is lower bounded by the ratio that we get by plugging the density D, given in (42), into the 1-bit decoder with erasures. In this way, by using Lemma 5 we obtain for $i \in \mathcal{I}_{n,d}$

$$C_i \ge p - 4\sqrt{pm} \ge 1 - 16(n+2)(\frac{2}{e})^{2n} - 16\sqrt{n+2}(\frac{2}{e})^{n}. \tag{43}$$

We thus obtain from (36) and (43)

$$C(W, Q) \ge (I(W) - \frac{d}{2})(1 - 16(n+2)(\frac{2}{e})^{2n} - 16\sqrt{n+2}(\frac{2}{e})^{n}). \tag{44}$$

C. Third Step: Putting things together

In the last step, we relate the values d, n_d and the lower bound (44) together. We first choose $n_1 \in \mathbb{N}$ such that for $n \ge n_1$ we have

$$16(n + 2)(\frac{2}{e})^{2n} + 16\sqrt{n + 2}(\frac{2}{e})^{n} \le \frac{d}{2}. \tag{45}$$

One can easily see that for small values of d, a suitable candidate for n_1 is $n_1 = \frac{1}{\log(\frac{e}{2})}\log(\frac{1}{d}) + o(\log(\frac{1}{d}))$. However, to have an explicit candidate for n_1 such that (45) holds for all values of d, one can fix

$$n_1 = 3\log(\frac{1}{d}) + 17. \tag{46}$$

Now, let $n = \max(n_1, n_d)$. From (44) and (45) it is easy to see that $C(W, Q) \ge I(W) - d$. In other words, by choosing $M = 2n$ and $\Delta = 2^{-(n+1)}$ for the function Q given in (2), the algorithm SCD_Q is capable of achieving rates that satisfy $C(W, Q) \ge I(W) - d$. Also, note that we have

$$|\, \mathcal{Q}\,| = 1 + \frac{2M}{\Delta} = 1 + n2^{n+2}.$$

As a result,

$$\log |\, \mathcal{Q}\,| \approx n + \log n + 2. \tag{47}$$

Finally, what remains to be done is to relate n_d to d.

Lemma 10: In order to have (36) for $n \ge n_d$, it is enough to let

$$n_d = 7\log(\frac{1}{d}) + \log(\log(\frac{1}{d}))^2 + 48. \tag{48}$$

With such a choice of n_d and n_1 as in (48) and (46), we have $n_d \ge n_1$ and $n = n_d$. Thus, we obtain from (47)

$$\log |\, \mathcal{Q}\,| \le 7\log(\frac{1}{d}) + O(\log(\log(\frac{1}{d}))).$$

VII. Conclusion and Open Problems

We have shown that polar codes are very robust with respect to quantization at the decoder – even very simple decoders with only a few messages achieve a high fraction of the capacity. This is good news if we are interested in a low-complexity implementation.

Not all news is good. Numerical calculations indicate that the speed of the polarization is in general further decreased by quantization. This means that we need to construct even longer codes.

A precise characterization of this trade-off, namely the trade-off between the polarization speed and the quantization would be of considerable practical value.

Acknowledgment

The work of Hamed Hassani has been supported by Swiss National Science Foundation Grant no 200021-121903.

References

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

[2] E. Arıkan and E. Telatar, "On the rate of channel polarization," in *Proc. 2009 IEEE Int. Symp. Info. Theory*, Seoul, South Korea, pp. 1493–1495, 2009.

[3] S. H. Hassani, S. B. Korada and R. Urbanke, "The compound capacity of polar codes," in *Proc. 47th Annual Allerton Conference on Communication, Control, and Computing*, pp. 16–21, 2009.

[4] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2009.

[5] S. B. Korada, A. Montanari, E. Telatar and R. Urbanke, "An empirical scaling law for polar codes," in *Proc. 2010 IEEE Int. Symp. Info. Theory*, Austin, Texas, USA, pp. 884-888, 2010.

[6] S. H. Hassani, K. Alishahi and R. Urbanke, "On the scaling of polar codes: II. The behavior of un-polarized channels," in *Proc. 2010 IEEE Int. Symp. Info. Theory*, Austin, Texas, USA, pp. 879–883, 2010.

[7] R. Mori and T. Tanaka, "Performance and construction of polar codes on symmetric binary-input memoryless channels," in *Proc. 2009 IEEE Int. Symp. Info. Theory*, Seoul, South Korea, pp. 1496–1500, 2009.

[8] I. Tal and A. Vardy, "How to construct polar codes," presented at 2010 IEEE Info. Theory Workshop, Dublin, Ireland, 2010. [online] Available: `arXiv:1105.6164v1 [cs.IT]`.

[9] R. Pedarsani, H. Hassani, I. Tal and E. Telatar, "On the construction of polar codes," in *Proc. 2011 IEEE Int. Symp. Info. Theory*, St. Petersberg, Russia, pp. 11–15, 2011.

[10] C. Leroux, I. Tal, A. Vardy and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *Proc. ICASSP 2011*, Prague, Czech Republic, pp. 1665-1668, 2011.

[11] I. Tal and A. Vardy, "List decoding of polar codes," in *Proc. 2011 IEEE Int. Symp. Info. Theory*, St. Petersberg, Russia, pp. 1–5, 2011.

[12] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

Appendix

A. Proof of Lemma 5

The fact that $h(D=(1,0,0))=1$, $h(D=(0,1,0))=0$ is very easy to check and thus it remains to prove (28). Using (15) and (16) we obtain

$$h(D^+) = p^2+2pe-4\sqrt{(p^2+2pe)(m^2+2pm)},$$
$$h(D^-) = p^2+m^2-4\sqrt{2pm(p^2+m^2)}.$$

After some straight forward simplifications, we get

$$\frac{h(D^+)+h(D^-)}{2}$$
$$= p+\frac{m^2}{2}$$
$$-2\sqrt{pm}\Big(\frac{\sqrt{pm}}{2}+\sqrt{(p+2e)(m+2e)}+\sqrt{2(p^2+m^2)}\Big).$$

Thus, in order to show (28), it is necessary that the right side of the above equality is less than $p-4pm$. We now prove a slightly stronger inequality: For $p+e+m=1$ we have

$$\frac{\sqrt{pm}}{2}+\sqrt{(p+2e)(m+2e)}+\sqrt{2(p^2+m^2)}\Big) \le 2. \quad (49)$$

It is easy to see that the above inequality results (23). To prove (49) we use the fact that

$$\sqrt{(p+2e)(m+2e)} \le \frac{p+2e+m+2e}{2} = 2-\frac{3}{2}(p+m),$$

and apply it to (49). Thus to have (49), it is sufficient to prove

$$\sqrt{pm}2+\sqrt{2(p^2+m^2)} \le \frac{3}{2}(p+m), \quad (50)$$

by squaring both sides of (50) and some further simplifications we get to

$$\sqrt{2pm(p^2+m^2)} \le \frac{1}{4}(p^2+m^2)+\frac{17}{4}pm.$$

The above inequality can easily proved by noting the fact that for $x,y\ge 0$ we have $x+y\ge 2\sqrt{xy}$, and hence

$$\frac{1}{4}(p^2+m^2)+\frac{17}{4}pm \ge 2\sqrt{\frac{17}{16}pm(p^2+m^2)} \ge \sqrt{2pm(p^2+m^2)}.$$

B. Proof of Lemma 6

Note that for $D\in\mathcal{D}$, the minus operation given in (16) is exactly the same as the original minus operation without any further quantization step, i.e., $D^- = D \circledast D$. We know from [1] that for any BMS channel we have $Z(W\circledast W)\le 2Z(W)$ and hence $Z(D^-)\le 2Z(D)$. To show (32), assuming $D = m\Delta_{-\infty}+e\Delta_0+p\Delta_\infty$. We have from (15),

$$\begin{aligned}
Z(D^+) &= 2\sqrt{(p^2+2pe)(m^2+2me)}+e^2+2pm\\
&= 2\sqrt{pm}\sqrt{(p+2e)(m+2e)}+e^2+2pm\\
&= 2\sqrt{pm}\sqrt{(pm+4e^2+2e(m+p)+e^2+2pm}\\
&\overset{(a)}{=} 2\sqrt{pm}\sqrt{(pm+2e(1+e))}+e^2+2pm\\
&\overset{(b)}{\le} 2\sqrt{pm}(\sqrt{pm}+\sqrt{2e(1+e)})+e^2+2pm\\
&= (2\sqrt{pm}+e)^2+2\sqrt{pm}(\sqrt{2e(1+e)}-e)\\
&= Z(D)^2+2\sqrt{pm}(\sqrt{2e(1+e)}-e),
\end{aligned}$$

where step (a) follows from the fact that $m+e+p=1$ and step (b) follows from the inequality $\sqrt{(a+b)}\le\sqrt{(a)}+\sqrt{(b)}$. Following the above lines, to get (32), it is enough to show that

$$\begin{aligned}
2\sqrt{pm}(\sqrt{2e(1+e)}-e) &\le 2Z(D)^{\frac{3}{2}}-Z(D)^2\\
&= Z(D)(2\sqrt{(Z(D))}-Z(D)).
\end{aligned}$$

Now, by noting that $Z(D)\ge 2\sqrt{pm}$, we only need to show the following,

$$\sqrt{2e(1+e)}-e \le 2\sqrt{(Z(D))}-Z(D)$$

$$= 2\sqrt{2\sqrt{pm}+e} - 2\sqrt{pm} - e.$$

Rearranging the terms, we should prove

$$\sqrt{2e(1+e)} + 2\sqrt{pm} \leq 2\sqrt{2\sqrt{pm}+e},$$

which by dividing both sides by 2 and then squaring both sides gives

$$\frac{e(1+e)}{2} + pm + \sqrt{2pme(1+e)} \leq 2\sqrt{pm} + e.$$

Now, since $e \leq 1$, we have $\frac{e(1+e)}{2} \leq 2$ and after further simplifications we finally get to the following relation to prove.

$$\sqrt{pm} + \sqrt{2e(1+e)} \leq 2,$$

which by noting that $\sqrt{pm} \leq \frac{p+m}{2} = \frac{1-e}{2}$, reduces to the following inequality

$$\frac{1-e}{2} + \sqrt{2e(1+e)} \leq 2.$$

It is straight forward to show that the above inequality holds for $e \in [0,1]$.

C. A lower bound on the tail probability for symmetric densities

Lemma 11: Let W be a BMS channels and let the r.v. L represent the log-likelihood value of its output, i.e., $L(Y) = \log(\frac{W(Y\,|\,0)}{W(Y\,|\,1)})$, where $Y \sim W(y\,|\,0)$. We have for $y \geq 0$

$$\Pr(L \leq y) \leq (1 + \frac{1}{2}e^{\frac{y}{2}})Z(W). \tag{51}$$

Proof: We have

$$\Pr(L \leq y) = \Pr(L \leq 0) + \Pr(0 < L \leq y). \tag{52}$$

Let $l(x)$ denote the pdf of the r.v. L. We first note that

$$Z(W) = \int_{-\infty}^{\infty} e^{-\frac{x}{2}} dl(x). \tag{53}$$

As a result,

$$\Pr(L \leq 0) \leq Z(W). \tag{54}$$

Also, since W is symmetric, we have for $x \geq 0$: $l(-x) = e^{-x}l(x)$ and hence from (53) we have, $Z(W) \geq 2\int_{0^+}^{\infty} e^{-\frac{x}{2}} dl(x)$. Consequently, we obtain

$$\Pr(0 < L \leq y) \leq e^{\frac{y}{2}} \int_{0^+}^{y} e^{-\frac{x}{2}} dl(x) \leq \frac{1}{2}e^{\frac{y}{2}} Z(W). \tag{55}$$

The proof now follows from (52), (54) and (55). ∎

D. Proof of Lemma 9

Throughout the proof we will frequently use the following definition.

Definition 12: Consider a path $P := (j_1,k_1),(j_2,k_2),\cdots,(j_l,k_l)$ in the graph $T(i)$, where we assume that $l \geq 2$ and $0 \leq j_l < j_{l-1} < \cdots < j_1 \leq n$. In other words, P is a path of length $l-1$ that starts from the node (j_1,k_1) and continues upwards through $T(i)$ by passing through $(j_2,k_2),(j_3,k_3),\cdots$ and finally reaches its endpoint (j_l,k_l). We call such a path an *upwards path* and denote the set of such paths by $\mathcal{P}$. For a path $P \in \mathcal{P}$, we define the set $S(P)$ as the set of nodes (j,k) such that (j,k) is a variable node and is adjacent to one of the nodes $(j_1,k_1),\cdots,(j_{l-1},k_{l-1})$. An example of a path P, consider the tree-channel in Figure 3 and let P be the path between labels $(0,4),(1,2),(2,1),(3,0)$. For this path we have $S(P) = \{(2,0),(1,3)\}$.

It is an easy exercise to show that the the number of downwards paths in a binary tree of height n is equal to

$$|\,\mathcal{P}\,| = (n-1)2^{n+1} + 2. \tag{56}$$

Now, recall that the messages $m_{j,k}$ correspond to the original SC decoder and the messages $\hat{m}_{j,k}$ correspond to the algorithm SCD_Q (with $M = 2n$ and $\Delta = 2^{-(n+1)}$). We know that by the all-zero codeword assumption, the messages $m_{j,k}$ have a symmetric density, i.e., for any real number $x \in \mathbb{R}$ we have

$$\Pr(m_{j,k} = x) = e^{-x}\Pr(m_{j,k} = -x). \tag{57}$$

As a result, we have for any label (j,k)

$$\mathbb{E}[e^{-m_{j,k}}] = 1. \tag{58}$$

Hence, by the Markov inequality we get for $x \geq 0$

$$\Pr(m_{j,k} \leq -x) \leq e^{-x}. \tag{59}$$

Define the event E_1 as

$$E_1 = \{\forall (j,k) : m_{j,k} > -2n+1\}. \tag{60}$$

Using (59) and applying the union bound we obtain

$$\Pr(E_1) \geq 1 - 2^{n+1}e^{1-2n}. \tag{61}$$

Also, define the event E_2 as

$$E_2 := \{\forall P \in \mathcal{P} : \sum_{(j,k)\in S(P)} m_{j,k} \geq (n+3)\ln 2 - 2n\}. \tag{62}$$

We now claim that

$$\Pr(E_2) \geq 1 - n2^{2n+4}e^{-2n}. \tag{63}$$

To show (63) note that for each specific path $P \in \mathcal{P}$ s.t. $S(P)$ is non-empty, the random variables $\{m_{j,k}\,|\,(j,k) \in S(P)\}$ are independent (This is due to the fact that we feed independent observations of the channel W into the leaf nodes of the tree $T(i)$). Hence, by using (58) we get

$$\begin{aligned}
&\Pr(\sum_{(j,k)\in S(P)} m_{j,k} < (n+3)\ln 2 - 2n) \\
&= \Pr(e^{-\sum_{(j,k)\in S(P)} m_{j,k}} > e^{2n}2^{-(n+3)}) \\
&\leq \frac{\mathbb{E}[e^{-\sum_{(j,k)\in S(P)} m_{j,k}}]}{e^{2n}2^{-(n+3)}} \\
&= \frac{\prod_{(j,k)\in S(P)} \mathbb{E}[e^{-m_{j,k}}]}{e^{2n}2^{-(n+3)}} \\
&= 2^{n+3}e^{-2n}
\end{aligned} \tag{64}$$

The claim (63) now follows by applying the union bound to all paths $P \in \mathcal{P}$ and by using (56) and (64). Finally, we claim that conditioned on the event $E = E_1 \cap E_2$, the following hold for each label (j,k):

1) If $\hat{m}_{j,k} = \infty$, then $m_{j,k} \geq (n+3)\ln 2$.

2) if $\hat{m}_{j,k} \neq \infty$, then $|\hat{m}_{j,k} - m_{j,k}| \leq (2^{j+1}-1)\delta$.
3) $\hat{m}_{j,k} \neq -\infty$.

Firstly, note that the proof of Lemma 9 follows from the claims 1-3 by inserting $(j,k) = (n,0)$ and noting that

$$\begin{aligned}\Pr(E_1 \cap E_2) &\geq \Pr(E_1) + \Pr(E_2) - 1 \\ &\overset{(61),(63)}{\geq} 1 - 2^{n+1}e^{1-2n} - n2^{2n+4}e^{-2n} \\ &\geq 1 - 16(n+1)(\frac{2}{e})^{2n}.\end{aligned}$$

Hence, it remains to show that conditioned on the event $E = E_1 \cap E_2$, the claims 1-3 hold. We show this by induction. We first show that the claims 1-3 hold for all the messages $m_{0,j}$, i.e., the messages that correspond to the leaf nodes of $T(i)$: Claim 1 follows the fact that $M = 2n$, claim 2 follows from the definition of the function Q given in (2) and claim 3 is due to the definition of E_1. Let $t \in \{1,2,\cdots,n\}$. We now assume that the claims 1-3 hold for all the messages $\hat{m}_{j,k}$ where $j \leq t-1$ and we show that these claims also hold for all the messages $\hat{m}_{t,k}$. We first prove claim 2. Consider a label (t,k). If $\hat{m}_{t,k} \neq \infty$, then either $\hat{m}_{t-1,2k-1}, \hat{m}_{t-1,2k} \neq \infty$ or one of the messages $\hat{m}_{t-1,2k-1}$ or $\hat{m}_{t-1,2k}$ is equal to ∞ and the other is finite and the node (t,k) is a check node (note that by the induction hypothesis we have $\hat{m}_{t-1,2k-1}, \hat{m}_{t-1,2k} \neq -\infty$). In the former case, by the induction hypothesis and claim 2 we have

$$\begin{aligned}&|\hat{m}_{t-1,2k-1} - m_{t-1,2k-1}| \leq (2^{(t-1)}-1)\delta, \\ &|\hat{m}_{t-1,2k} - m_{t-1,2k}| \leq (2^{(t-1)}-1)\delta,\end{aligned}$$

and by using (39) and (40) we get claim 2 for the message $\hat{m}_{t,k}$. In the latter case, assume w.l.o.g. that $\hat{m}_{t-1,2k-1} = \infty$ and $\hat{m}_{t-1,2k} \neq \infty$. In this way, by using claim 1 and 2 we get

$$\begin{aligned}&m_{t-1,2k-1} \geq (n+3)\ln 2, \\ &|\hat{m}_{t-1,2k} - m_{t-1,2k}| \leq (2^{t-1}-1)\delta.\end{aligned}$$

Since the node (t,k) is a check node we have $\hat{m}_{t,k} = \hat{m}_{t-1,2k}$. Hence, we can write

$$\begin{aligned}&|\hat{m}_{t,k} - m_{t,k}| \\ &= |\hat{m}_{t-1,2k} - 2\tanh^{-1}(\tanh(\frac{m_{t-1,2k-1}}{2})\tanh(\frac{m_{t-1,2k}}{2}))| \\ &= |\hat{m}_{t-1,2k} - 2\tanh^{-1}\big[\tanh(\frac{m_{t-1,2k}}{2}) \\ &\qquad + (\tanh(\frac{m_{t-1,2k-1}}{2})-1)\tanh(\frac{m_{t-1,2k}}{2})\big]| \\ &\overset{(a)}{\leq} |\hat{m}_{t-1,2k} - m_{t-1,2k}| + 2(1-\tanh(\frac{m_{t-1,2k-1}}{2})) \\ &\leq (2^{t-1}-1)\delta + 2(1 - \frac{1-e^{-(n+3)\ln 2}}{1+e^{-(n+3)\ln 2}}) \\ &\leq 2(2^{t-1}-1)\delta = (2^{n-t-1}-1)\delta.\end{aligned}$$

Here, the relation (a) follows from the fact that for $x, y \in \mathbb{R}$ we have $\tanh(x+y)$ The proof of claim 3 can be easily followed by a similar argument and hence we omit it here. Finally we prove claim 1. Consider a node (t,k) and assume that $\hat{m}_{t,k} = \infty$.

E. Proof of Lemma 10

Let $\{B_n\}_{n\in\mathbb{N}}$ be a sequence of iid Bernoulli($\frac{1}{2}$) random variables. Denote by $(\mathcal{F}, \Omega, \mathbb{P})$ the probability space generated by this sequence and let $(\mathcal{F}_n, \Omega_n, \mathbb{P}_n)$ be the probability space generated by $(B_1, \cdots, B_n)$. Also, denote by θ_n the natural embedding of $\mathcal{F}_n$ into $\mathcal{F}$, i.e., for every $F \in \mathcal{F}_n$

$$\theta_n(F) = \{(b_1, b_2, \cdots, b_n, b_{n+1}, \cdots) \in \Omega \,|\, (b_1, \cdots, b_n) \in F\}.$$

We have $\mathbb{P}_n(F) = \mathbb{P}(\theta_n(F))$. We now couple the process W_n with the sequence $\{B_i\}$:

$$W_n = \begin{cases} W_{n-1}^+ & ;\text{if } B_n = 1, \\ W_{n-1}^- & ;\text{if } B_n = 0. \end{cases} \tag{65}$$

As a result, $Z_n = Z(W_n)$ is coupled with the sequence $\{B_i\}$. By using the bounds given in [12, Chapter 4] we have the following relationship between the Bhattacharyya parameters of W^+, W^- and W:

$$\begin{aligned}&Z(W^+) = Z(W)^2, \\ &Z(W)\sqrt{2 - Z(W)^2} \leq Z(W^-) \leq 2Z(W) - Z(W)^2.\end{aligned}$$

As a result, for a BMS channel W, the process $Z_n = Z(W_n)$ satisfies ([4, Lemma 3.16])

$$Z_{n+1} \begin{cases} = {Z_{n-1}}^2 & ;\text{if } B_n = 1, \\ \in [Z_{n-1}\sqrt{2 - {Z_{n-1}}^2}, 2Z_n - {Z_{n-1}}^2] & ;\text{if } B_n = 0. \end{cases} \tag{66}$$

Lemma 13: Consider the process Z_n with the starting value $Z_0 = z_0$.

(i) For $a, b \in (0,1)$, define $\zeta(a,b)$ as

$$\zeta(a,b) \triangleq \sup_{\{x\in(0,1), y\in[x\sqrt{2-x^2}, x(2-x)]\}} \frac{x^{2a}(1-x^2)^{2b} + y^a(1-y)^b}{2x^a(1-x)^b}.$$

We have

$$E[Z_n^a(1-Z_n)^b] \leq z_0^a(1-z_0)^b\zeta(a,b)^n. \tag{67}$$

Furthermore, for $a = 0.82$ and $b = 0.60$ we have $\zeta_{a,b} \leq 0.89$.

(ii) We have

$$\Pr(Z_n \leq 2^{-2^{\sum_{i=1}^n B_i}}) \geq 1 - 6z_0(1 + \log(\frac{1}{z_0})). \tag{68}$$

(iii) We have

$$\Pr(Z_n^2 \geq 1 - 2^{-2^{n-\sum_{i=1}^n B_i}}) \geq 1 - 6(1-z_0^2)(1 + \log(\frac{1}{1-z_0^2})). \tag{69}$$

■

Before proving Lemma 13, let us show how the proof of Lemma 10 follows from it. Consider the first part of Lemma 13 with $a = 0.82$ and $b = 0.6$ and Let $n1 \in \mathbb{N}$ be such that

$$\mathbb{E}[Z_{n_1}^a(1-Z_{n_1})^b] \leq \frac{d}{24}. \tag{70}$$

By using part (i) of Lemma 13, if we let

$$n_1 = \frac{\log(\frac{d}{12})}{\log \zeta_{a,b}} \leq 6\log(\frac{1}{d}) + 22, \tag{71}$$

then the relation (70) holds universally for any channel W. We now search for an integer n_2 such that for the following two events

$$E_1 = \{\sum_{i=1}^{n_2} B_i \leq \log(3n_1 \log e)\},$$
$$E_2 = \{\sum_{i=1}^{n_2} B_i \geq n_2 - \log(3n_1 \log e)\},$$

we have

$$\Pr(E_1 \cup E_2) \leq \frac{d}{4}. \tag{72}$$

First, note that the two events E_1 and E_2 are equi-probable and hence by using the union bound we get $\Pr(E_1 \cap E_2) \leq 2\Pr(E_1)$. Thus, we desire a candidate for n_2 such that

$$\Pr(E_1) \leq \frac{d}{8}. \tag{73}$$

Now, since B_i's are i.i.d. random variables with distribution Bernoulli($\frac{1}{2}$), (73) becomes

$$\frac{\sum_{i=0}^{\lfloor \log(3n_1 \log e)\rfloor} \binom{n_2}{i}}{2^{n_2}} \leq \frac{d}{8},$$

and after a further simplification step, it is sufficient to have

$$\frac{(\lfloor \log(3n_1 \log e)\rfloor + 1)\binom{n_2}{\lfloor \log(3n_1 \log e)\rfloor}}{2^{n_2}} \leq \frac{d}{8}. \tag{74}$$

By looking more closely at (74) and (70), one can easily deduce that $n_2 = \log(\frac{1}{d}) + o(\log(\frac{1}{d}))$ is sufficient to fulfill (74). However, one precise candidate to fulfill (74) for all values of $d \leq \frac{1}{2}$ is

$$n_2 = \log(\frac{1}{d}) + (\log(\log(\frac{1}{d})))^2 + 26. \tag{75}$$

We now let

$$n_d = n_1 + n_2 = 7\log(\frac{1}{d}) + (\log(\log(\frac{1}{d})))^2 + 48, \tag{76}$$

and we show that for such a choice of n_d we have the statement of Lemma 10.

Proof of Lemma 13: For part (ii), Consider two processes Z_n^u given by $Z_0^u = Z(W)$,

$$Z_n^u = \begin{cases} (Z_{n-1}^u)^2 & ;\text{if } B_n = 1, \\ 2Z_{n-1}^u & ;\text{if } B_n = 0, \end{cases} \tag{77}$$

Clearly, Z_n is stochastically dominated by Z_n^u. The following lemma partially analyzes the behavior of Z_n^u.

Lemma 14: For the process Z_n^u (defined in (77)) starting at $Z_0^u = z_0^u \in (0,1)$ we have:

$$\mathbb{P}(Z_n^u \leq 2^{-\beta 2^{\sum_{i=1}^n B_i}}) \geq 1 - 2^{1+\beta}\sqrt{z_0^u}. \tag{78}$$

Proof: We analyze the process[4] $A_n = -\log(Z_n^u)$, i.e., $A_0 = -\log(z_0^u) \triangleq a_0$ and

$$A_{n+1} = \begin{cases} 2A_n & ;\text{if } B_n = 1, \\ A_n - 1 & ;\text{if } B_n = 0. \end{cases} \tag{79}$$

[4]In this paper, all the logarithms are in base 2.

Note that in terms of the process A_n, the statement of the lemma can be phrased as

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^n B_i}) \geq 1 - \frac{1}{2^{a_0 - \beta}}.$$

Associate to each $(b_1, \cdots, b_n) \triangleq \omega_n \in \Omega_n$ a sequence of "runs" $(r_1, \cdots, r_{k(\omega_n)})$. This sequence is constructed by the following procedure. We define r_1 as the smallest index $i \in \mathbb{N}$ so that $b_{i+1} \neq b_1$. In general, if $\sum_{j=1}^{k-1} r_j < n$ then

$$r_k = \min\{i \mid \sum_{j=1}^{k-1} r_j < i \leq n, b_{i+1} \neq b_{\sum_{j=1}^{k-1} r_j}\} - \sum_{j=1}^{k-1} r_j.$$

The process stops whenever the sum of the runs equals n. Denote the stopping time of the process by $k(\omega_n)$. In words, the sequence $(b_1, \cdots, b_n)$ starts with b_1. It then repeats b_1, r_1 times. Next follow r_2 instances of $\overline{b_1}$, followed again by r_3 instances of b_1, and so on. We see that b_1 and $(r_1, \cdots, r_{k(\omega_n)})$ fully describe $\omega_n = (b_1, \cdots, b_n)$. Therefore, there is a one-to-one map

$$(b_1, \cdots, b_n) \longleftrightarrow \{b_1, (r_1, \cdots, r_{k(\omega_n)})\}. \tag{80}$$

Note that we can either have $b_1 = 1$ or $b_1 = 0$. We start with the first case, i.e., we first assume $B_1 = 1$. We have:

$$\sum_{i=1}^{n} b_i = \sum_{j \text{ odd} \leq k(\omega_n)} r_j,$$

and

$$n = \sum_{j=1}^{k(\omega_n)} r_j.$$

Analogously, for a realization $(b_1, b_2, \cdots) \triangleq \omega \in \Omega$ of the infinite sequence of random variable $\{B_i\}_{i\in\mathbb{N}}$, we can associate a sequence of runs $(r_1, r_2, \cdots)$. In this regard, considering the infinite sequence of random variables $\{B_i\}_{i\in\mathbb{N}}$ (with the extra condition $B_1 = 1$), the corresponding sequence of runs, which we denote by $\{R_k\}_{k\in\mathbb{N}}$, is an iid sequence with $\mathbb{P}(R_i = j) = \frac{1}{2^j}$. Let us now see how we can express the A_n in terms of the $r_1, r_2, \cdots, r_{k(\omega_n)}$. We begin by a simple example: Consider the sequence $(b_1 = 1, b_2, \cdots, b_8)$ and the associated run sequence $(r_1, \cdots, r_5) = (1, 2, 1, 3, 1)$. We have

$$\begin{aligned}
A_1 &= a_0 2^{r_1}, \\
A_3 &= a_0 2^{r_1} - r_2, \\
A_4 &= (a_0 2^{r_1} - r_2)2^{r_3} = a_0 2^{r_1 + r_3} - r_2 2^{r_3}, \\
A_7 &= (a_0 2^{r_1} - r_2)2^{r_3} - r_4 = a_0 2^{r_1+r_3} - r_2 2^{r_3} - r_4, \\
A_8 &= ((a_0 \times 2^{r_1} - r_2) \times 2^{r_3} - r_4) \times 2^{r_5} \\
&= a_0 2^{r_1+r_3+r_5} - r_2 2^{r_3+r_5} - r_4 2^{r_5} \\
&= 2^{r_1+r_3+r_5}(a_0 - 2^{-r_1} r_2 - 2^{-(r_1+r_3)} r_4).
\end{aligned}$$

In general, for a sequence $(b_1, \cdots, b_n)$ with the associated run sequence $(r_1, \cdots, r_{k(\omega_n)})$ we can write:

$$\begin{aligned}
A_n &= a_0 2^{\sum_{i \text{ odd} \leq k(\omega_n)} r_i} - \sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{\sum_{i < j \text{ odd}} r_j} \\
&= a_0 2^{\sum_{i \text{ odd} \leq k(\omega_n)} r_i} - \sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{(-\sum_{j \text{ odd} < i} r_j + \sum_{i \text{ odd} \leq k(\omega_n)} r_i)}
\end{aligned}$$

$$= [2^{\sum_{i \text{ odd} \leq k(\omega_n)} r_i}][a_0 - (\sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j})]$$
$$= [2^{\sum_{i=1}^{n} B_i}][a_0 - (\sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j})].$$

Our aim is to lower-bound

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i}) = \mathbb{P}_n(a_0 - \sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j} \geq \beta),$$

or, equivalently, to upper-bound

$$\mathbb{P}_n(\sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j} \geq a_0 - \beta). \quad (81)$$

For $n \in \mathbb{N}$, define the set $U_n \in \mathcal{F}_n$ as

$$U_n = \{\omega_n \in \Omega_n \,|\, \exists l \leq k(\omega_n) : \sum_{i \text{ even} \leq l} r_i 2^{-\sum_{j \text{ odd} < i} r_j} \geq a_0 - \beta\}.$$

Clearly we have:

$$\mathbb{P}_n(\sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j} \geq a_0 - \beta) \leq \mathbb{P}_n(U_n).$$

In the following we show that if $(b_1, \cdots, b_n) \in U_n$, then for any choice of b_{n+1}, $(b_1, \cdots, b_n, b_{n+1}) \in U_{n+1}$. We will only consider the case when $b_n, b_{n+1} = 1$, the other three cases can be verified similarly. Let $\omega_n = (b_1, \cdots, b_{n-1}, b_n = 1) \in U_n$. Hence, $k(\omega_n)$ is an odd number (recall that $b_1 = 1$) and the quantity $\sum_{i \text{ even} \leq k(\omega_n)} r_i 2^{-\sum_{j \text{ odd} < i} r_j}$ does not depend on $r_{k(\omega_n)}$. Now consider the sequence $\omega_{n+1} = (b_1, \cdots, b_n = 1, 1)$. Since the last bit ($b_{n+1}$) equals 1, then $r_{k(\omega_{n+1})} = r_{k(\omega_n)}$ and the value of the sum remains unchanged. As a result $(b_1, \cdots, b_n, 1) \in U_{n+1}$. From above, we conclude that $\theta_i(U_i) \subseteq \theta_{i+1}(U_{i+1})$ and as a result

$$\mathbb{P}_i(U_i) = \mathbb{P}(\theta_i(U_i)) \leq \mathbb{P}(\theta_{i+1}(U_{i+1})) = \mathbb{P}_{i+1}(U_{i+1}).$$

Hence, the quantity $\lim_{n\to\infty} \mathbb{P}_n(U_n) = \lim_{n\to\infty} \mathbb{P}(\theta_n(U_n)) = \lim_{n\to\infty} \mathbb{P}(\cup_{i=1}^{n} \theta_i(U_i))$ is an upper bound on (81). On the other hand, consider the set

$$V = \{\omega \in \Omega \,|\, \exists l : \sum_{i \text{ even} \leq l} r_i 2^{-\sum_{j \text{ odd} < i} r_j} \geq a_0 - \beta\}.$$

By the definition of V we have $\cup_{i=1}^{\infty} \theta_i(U_i) \subseteq V$, and as a result, $\mathbb{P}(\cup_{i=1}^{\infty} \theta_i(U_i)) \leq \mathbb{P}(V)$. In order to bound the probability of the set V, note that assuming $B_1 = 1$, the sequence $\{R_k\}_{k\in\mathbb{N}}$ (i.e., the sequence of runs when associated with the sequence $\{B_i\}_{i\in\mathbb{N}}$) is an iid sequence with $\mathbb{P}(R_i = j) = \frac{1}{2^j}$. We also have

$$\mathbb{P}(a_0 - \sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j} \leq \beta) \quad (82)$$
$$= \mathbb{P}(\sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j} \geq a_0 - \beta)$$
$$= \mathbb{P}(2^{\sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j}} \geq 2^{a_0 - \beta})$$
$$\leq \frac{\mathbb{E}[2^{\sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j}}]}{2^{a_0-\beta}},$$

where the last step follows from the Markov inequality. The idea is now to provide an upper bound on the quantity $\mathbb{E}[2^{\sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j}}]$. Let $X = \sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j}$. We have

$$\mathbb{E}[2^X]$$
$$= \sum_{l=1}^{\infty} \mathbb{P}(R_2 = l)\mathbb{E}[2^X \,|\, R_2 = l]$$
$$\overset{a}{=} \sum_{l=1}^{\infty} \frac{1}{2^l}\mathbb{E}[2^X \,|\, R_2 = l]$$
$$= \sum_{l=1}^{\infty} \frac{1}{2^l}\mathbb{E}[2^{\frac{R_1}{2^l}}]\mathbb{E}[2^{\frac{X}{2^l}}]$$
$$= \sum_{l=1}^{\infty} \frac{1}{2^l(2^{1-\frac{1}{2^l}})}\mathbb{E}[2^{\frac{X}{2^l}}]$$
$$\overset{b}{\leq} \sum_{l=1}^{\infty} \frac{1}{2^l(2^{1-\frac{1}{2^l}})}(\mathbb{E}[2^X])^{\frac{1}{2^l}},$$

where (a) follows from the fact that R_is are iid and X is self-similar and (b) follows from Jensen inequality. As a result , an upper bound on the quantity $\mathbb{E}[2^X]$ can be derived as follows. We have

$$\mathbb{E}[2^X] \leq \frac{1}{2(2^{\frac{1}{2}}-1)}(\mathbb{E}[2^X])^{\frac{1}{2}} + \frac{1}{4(2^{\frac{3}{4}}-1)}(\mathbb{E}[2^X])^{\frac{1}{4}} + \frac{1}{4(2^{\frac{7}{8}}-1)}(\mathbb{E}[2^X])^{\frac{1}{8}}.$$

The equation $y = \frac{1}{2(2^{\frac{1}{2}}-1)}y^{\frac{1}{2}} + \frac{1}{4(2^{\frac{3}{4}}-1)}y^{\frac{1}{4}} + \frac{1}{4(2^{\frac{7}{8}}-1)}y^{\frac{1}{8}}$ has only one real valued solution $y^* \leq 2.87$. As a result we have $\mathbb{E}[2^X] \leq y^* \leq 2.87$. Thus by (82) we obtain

$$\mathbb{P}(a_0 - \sum_{i \text{ even} \leq m} R_i 2^{-\sum_{j \text{ odd} < i} R_j} \leq \beta) \leq \frac{2.87}{2^{a_0-\beta}}$$

Thus, given that $B_1 = 1$, we have:

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i}) \geq 1 - \frac{2.87}{2^{a_0-\beta}}.$$

Or more precisely we have

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i} \,|\, B_1 = 1) \geq 1 - \frac{2.87}{2^{a_0-\beta}}.$$

Now consider the case $B_1 = 0$. We show that a similar bound applies for A_n. Firstly note that, fixing the value of n, the distribution of R_1 is as follows: $\mathbb{P}(R_i) = \frac{1}{2^i}$ for $1 \leq i \leq n-1$ and $\mathbb{P}(R_1 = n) = \frac{1}{2^{n-1}}$. We have

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i} \,|\, B_1 = 0)$$
$$= \sum_{i=1}^{n} \mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i} \,|\, R_1 = i, B_1 = 0)\mathbb{P}(R_1 = i \,|\, B_1 = 0)$$
$$= \sum_{i \leq a_0-\beta, i \leq n} \mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i} \,|\, R_1 = i, B_1 = 0)\mathbb{P}(R_1 = i \,|\, B_1 = 0)$$
$$+ \sum_{i > a_0-\beta, i \leq n}^{n} \mathbb{P}(R_1 = i \,|\, B_1 = 0)$$
$$\leq \sum_{i \leq a_0-\beta, i \leq n} \frac{1}{2^i}\frac{2.87}{2^{a_0-\beta-i}} + \frac{2}{2^{a_0-\beta}}$$
$$\leq \frac{2.87(a_0 - \beta + 1)}{2^{a_0-\beta}}$$
$$\leq \frac{3}{2^{\frac{a_0-\beta}{2}}}.$$

Hence, considering the two cases together, we have:

$$\mathbb{P}(A_n \geq \beta 2^{\sum_{i=1}^{n} B_i}) \geq 1 - \frac{2}{2^{\frac{a_0-\beta}{2}}}.$$

■