Polar Codes for Partial Response Channels

Ubaid U. Fayyaz and John R. Barry
School of Electrical and
Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250
Email: ubaid@gatech.edu, john.barry@ece.gatech.edu

***Abstract*—We describe an error-correcting system that combines polar codes with turbo equalization for partial response channels. The successive cancellation decoder, originally proposed by Arikan for polar codes, does not produce the soft outputs needed for turbo processing. The belief propagation decoder, on the other hand, requires many iterations and has high computational complexity. In this paper, we propose a soft-input soft-output variant of the successive cancellation decoder that produces the soft information required for turbo architectures, while keeping the computational complexity low. Numerical results show that the proposed decoder performs better than the hard-output successive cancellation decoder and the belief propagation decoder in the context of turbo equalization. The proposed decoder achieves this performance gain with lower complexity compared to belief propagation and maximum-likelihood decoders. Additionally, we prove that Arikan's successive cancellation decoder is a fast-polarizing instance of our soft-input soft-output successive cancellation decoder.**

I. Introduction

Polar codes are recently discovered highly structured codes that provably achieve capacity for discrete memoryless channels (DMC) with an encoder and decoder of $O(N \log N)$ complexity [1], where N is the length of the code. Motivated by their low complexity as well as their extremely regular structure, universal rate adaptability and explicit construction, we investigate their feasibility on the magnetic recording channel. The error-correction processing in a typical magnetic recording application uses turbo equalization [2] [3], which involves the exchange of soft information between a soft-output partial-response detector and a soft-output error-control decoder. Therefore, in order to assess the feasibility of polar codes for the magnetic recording application (or more generally for any other system using concatenated coding and iterative turbo processing), we require a soft-output decoder for polar codes. In this paper, we describe a low-complexity soft-input soft-output decoder for polar codes.

In his seminal paper [1], Arikan proposed a successive cancellation (SC) decoder for polar codes. Polar codes with the SC decoder approach capacity of the DMC in the asymptotic region of large N. Since [1], many researchers have considered using other decoders to improve the performance of polar codes in the finite-length regime. In [4], the authors proposed a belief propagation (BP) decoder to compare the performance of polar codes with Reed-Muller codes, but the BP decoder has high complexity that makes it impractical for many applications. Later, the authors in [5] considered linear program decoding of polar codes, while in [6], the authors proposed a successive cancellation list (SCL) decoder that bridged the gap to state-of-the-art low density parity check (LDPC) codes to a greater extent at the expense of increased time- and space-complexity of $O(LN \log N)$ and $O(LN)$ respectively, where L is the list size. Recently, the authors in [7] described a stack decoder that approached maximum-likelihood performance with reduced time-complexity compared to SCL in the high-SNR regime.

Despite the wide variety of previously reported decoders for polar codes, ranging from SC variants and list decoders to belief propagation [4] and maximum likelihood decoders, none provide the soft outputs with low complexity that would be required by many practical applications of turbo processing. In this paper, we propose a soft-input soft-output (SISO) version of the SC decoder called the *soft cancellation (SCAN)* decoder that produces reliability information for the estimates of both the coded and message bits. The proposed SCAN decoder outperforms the best-available soft-output decoder for polar codes, namely the BP decoder, for a class of partial response channels while keeping the computational complexity low.

The paper is organized as follows. Section II describes the system model under consideration and Section III, reviews Arikan's SC decoder. In Section IV, we describe the limitations of the SC decoder as a soft-output decoder, and propose the SCAN decoder. In section V, we present numerical results using the SCAN decoder to support our claims.

II. System Model

Consider a polar code of length N and dimension K. Arikan showed in [1] that this code can be constructed using the generator matrix $\mathbf{G}_N = \mathbf{G}_2^{\otimes n}$, where $n = \log_2(N)$, $(.)^{\otimes n}$ denotes the nth Kronecker power, and

$$\mathbf{G}_2 := \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \tag{1}$$

To transmit a message vector $\mathbf{m} = [m_0, m_1, \ldots, m_{N-1}]$ of length K, we form a vector $\mathbf{u} = [u_0, u_1, \ldots, u_{N-1}]$ of length N such that $\mathbf{m}$ appears in $\mathbf{u}$ corresponding to the index set $\mathcal{I} \subseteq \{0, 1, 2, \ldots, N-1\}$ of cardinality K. We fix the bits in $\mathbf{u}$ on indices in the set $\mathcal{I}^c$ to 0 where c denotes complement of the set. We then reveal these fixed bits and their locations to both the encoder and decoder before transmission. In rest of the paper, we call the set $\mathcal{I}$ the set of '*free indices*' and the set $\mathcal{I}^c$ the set of '*frozen indices*'. The design of polar code consists of constructing the index set $\mathcal{I}$. For a list of these construction methods, see [1], [8], [9] and [10].

At the encoder, we compute $\mathbf{v} = \mathbf{u}\mathbf{G}_N$, map it to $\mathbf{w} \in \{1, -1\}^N$, interleave them using an S-random interleaver to $\mathbf{x}$ and pass the interleaved symbols $\mathbf{x}$ through a partial response channel of impulse response $\mathbf{h} = [h_0 h_1, \ldots, h_{\mu-1}]$ followed by an AWGN channel with noise variance $\sigma^2 = N_0/2$. The received codeword $\mathbf{r}$ at the output of the channel is

$$r_k = \sum_{i=0}^{\mu-1} h_i x_{k-i} + n_k, \tag{2}$$

where $n_k \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian random variable with mean zero and variance σ^2. The signal-to-noise ratio (SNR) is $E_b/N_0 = \sum_i {h_i}^2/(2R\sigma^2)$, where $R = K/N$.

The receiver uses a standard turbo equalization architecture. We first equalize the channel using the Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm [11]. We compute extrinsic log-likelihoods (LLRs) $\mathbf{e}_b$ using BCJR, deinterleave $\mathbf{e}_b$ to $\mathbf{a}_b$ and pass the result to the polar decoder that estimates the message vector $\hat{\mathbf{m}}$ as well as the extrinsic LLRs $\mathbf{e}_d$. We interleave $\mathbf{e}_d$ again and pass them to the BCJR equalizer that uses these LLRs as prior information to once again compute $\mathbf{e}_b$. We carry out this information/LLR exchange for I number of iterations and estimate the message vector $\hat{\mathbf{m}}$ using the results of the last iteration.

III. Successive Cancellation Decoder

In this section, we summarize the SC decoder proposed by Arikan [1]. We will follow the notation and description of [6]. Figure 1 shows the factor graph of the SC decoder. Let $L_\lambda(\phi, \omega)$ be the log-likelihood value and $B_\lambda(\phi, \omega)$ be the bit-decisions, computed from already detected/known bits, of the node ω in the node group ϕ at a depth λ in the graph. The SC decoder computes the message vector estimate using

$$\hat{m}_i = \begin{cases} 0 & \text{if } i \in \mathcal{I}^c \text{ or } L_n(i, 0) \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

by going from $i = 0$ to $N - 1$ in increasing order, where $L_n(i, 0)$ is computed using the recursion

$$L_\lambda(\phi, \omega) = L_{\lambda-1}(\psi, 2\omega) \boxplus L_{\lambda-1}(\psi, 2\omega + 1) \tag{4}$$

for ϕ even, and

$$L_\lambda(\phi, \omega) = \begin{cases} L_{\lambda-1}(\psi, 2\omega+1) + L_{\lambda-1}(\psi, 2\omega) & \text{if } B_\lambda(\phi - 1, \omega) = 0, \\ L_{\lambda-1}(\psi, 2\omega+1) - L_{\lambda-1}(\psi, 2\omega) & \text{if } B_\lambda(\phi - 1, \omega) = 1, \end{cases} \tag{5}$$

when ϕ is odd and $\boxplus$ is defined as

$$a \boxplus b \triangleq 2\tanh^{-1}\left[\tanh\left(\frac{a}{2}\right) \times \tanh\left(\frac{b}{2}\right)\right]. \tag{6}$$

Everytime we update $L_\lambda(\phi, \omega)$ for odd values of ϕ, we update the B matrix using

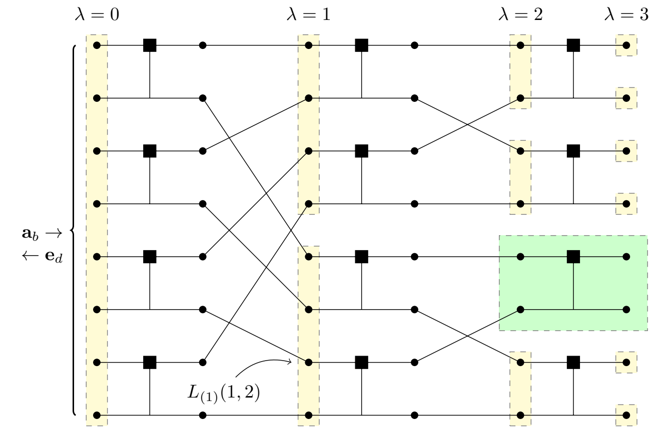

Fig. 1. Decoder factor graph and notation. The nodes in the greyed box at a particular depth λ belong to the same node group, indexed by ϕ, whereas the nodes within a group ϕ are indexed by ω. For example, the factor graph has two node groups under $\lambda = 1$, i.e., the upper and the lower node group represented by $\phi = 0$ and $\phi = 1$, respectively. The nodes within each of these node groups are indexed by $\omega \in \{0, 1, 2, 3\}$.

$$B_{\lambda-1}(\psi, 2\omega) = B_\lambda(\phi - 1, \omega) \oplus B_\lambda(\phi, \omega) \tag{7}$$

$$B_{\lambda-1}(\psi, 2\omega + 1) = B_\lambda(\phi, \omega), \tag{8}$$

where $\oplus$ is binary XOR operation and $\psi = \lfloor \frac{\phi}{2} \rfloor$. For a detailed description of the SC decoder with pseudocode, see [6].

IV. A Soft-Input Soft-Output Successive Cancellation Decoder

For iterative receiver architectures like a turbo equalizer, we need a decoder that outputs soft information about the reliability of bit estimates for both the message and encoded bits. One might be tempted to use a naive variant of the original SC decoder, in which the extrinsic LLRs $\mathbf{e}_d$ are calculated by first calculating $\{L_n(i, 0)\}_{i=0}^{N-1}$ and then traversing through the graph towards the left. However, for this or any other implementation that depends on $\{L_n(i, 0)\}_{i=0}^{N-1}$, the accuracy of these LLRs are of great importance. We observe that the LLRs that the SC decoder produces are distorted, as explained below.

Consider the basic decision element in the factor graph of the SC decoder, as indicated by the larger shaded rectangle in Figure 1. Suppose, bits u_0, u_1 are encoded using this polar code of length two, mapped to $x_0, x_1 \in \{+1, -1\}$ and are sent on a binary-input DMC W with transition probabilities $W(y|x)$. Suppose W_{SC}^+ and W_{SC}^- are the transformed channels defined by the transition probabilities $W_{SC}^+(y_0, y_1, u_0|u_1)$ and $W_{SC}^-(y_0, y_1|u_0)$ under SC decoder, respectively. In SC decoding, we assume, while computing the likelihood $W_{SC}^-(y_0, y_1, u_1|u_0)$ of the bit u_0, that u_1 is equally likely to be 0 or 1. On the other hand, while computing the likelihood for bit u_1, we assume that u_0 has been decoded with no error. Both of these assumptions introduce a distortion in the likelihood values and degrade performance in iterative schemes.

We can expect an improvement in the likelihood estimates if we can incorporate soft estimates in decoding instead of hard decision of u_0 and no information about u_1. We first show in the following lemma how the likelihood computation changes if we have access to such a soft estimate, and then we show how we provide this soft estimate in the SCAN decoder.

Lemma 1: Let $z_i : i \in \{0,1\}$ be the output of DMC's P_i, defined by the transition probabilities $P_i(z_i|u_i) : u_i \in \{0,1\}$ and conditionally independent of y_i. If we have access to z_i instead of perfect/no knowledge of u_i, the log-likelihood ratio of u_i under successive cancellation decoding is given by

$$L_1(0,0) = L_0(0,0) \boxplus (B_1(1,0) + L_0(0,1)), \tag{9}$$

$$L_1(1,0) = L_0(0,1) + (B_1(0,0) \boxplus L_0(0,0)). \tag{10}$$

The proof is provided in the Appendix.

Since all the processing on the factor graph of a polar code occurs locally on these basic decision elements, we can extend the same process to the complete factor graph of the code of any block length N. The only problem remains now is to show how can we provide these additional LLRs $B_1(0,0), B_1(1,0)$ in all decision elements in a factor graph.

In the start of a decoding cycle, we compute $\{L_0(0,k)\}_{k=0}^{(N-1)}$ as we receive symbols $\mathbf{r}$ from the channel. The decoder has a-priori information about the location of fixed bits that we use to initialize $\{B_n(k,0)\}_{k=0}^{(N-1)}$ using $\mathcal{I}$. Suppose, we are interested in finding the LLR $L_n(i,0)$ in (4) and (5) with $\{L_n(k,0)\}_{k=0}^{(i-1)}$ already computed and no information about $\{u_k\}_{k=(i+1)}^{N-1}$. Since there is no way we can have any information about $\{u_k\}_{k=(i+1)}^{N-1}$ in the first iteration, we will keep the assumption that they are equally likely, i.e., $B_n(k,0) = 0, \forall (i+1) \leq k \leq (N-1)$. It is noteworthy that we have already populated the L matrix partially from left to right while calculating $\{L_n(k,0)\}_{k=0}^{(i-1)}$. Therefore, as we calculate $\{L_n(k,0)\}_{k=0}^{(i-1)}$, we can use the partially calculated L matrix as a-priori information to calculate the B matrix from right to left using Eq. (9) and Eq. (10) on all the decision elements involved. When $i = N-1$, we have the B matrix with extrinsic LLRs corresponding to all the nodes in the decoder's factor graph. We once again start computing LLRs $\{L_n(i,0)\}_{i=1}^{N-1}$ but this time we have soft information in the B matrix for $\{u_k\}_{k=(i+1)}^{N-1}$ unlike the first iteration. Therefore, we can use the B matrix to supply a-priori information to all decision elements in subsequent iterations. We use this iterative process I times and in the last iteration we will have the extrinsic LLRs corresponding to message and coded bits in $\{L_n(i,0)\}_{i=0}^{N-1}$ and $\{B_0(0,i)\}_{i=0}^{N-1}$, respectively. We explain all the necessary implementation details in Algorithms 1, 3 and 2.

One of the important parameters of polar codes under a decoding scheme is the rate of channel polarization which describes how fast the capacity of the bit channels corresponding to the indices in $\mathcal{I}$ and $\mathcal{I}^c$ approaches 1 and 0, respectively as $N \to \infty$. We refer the interested readers to [1] and [12] for further details about this parameter and mention here the advantage of using the SC decoder in place

Algorithm 1: The SCAN decoder

```
Data: The input LLRs from BCJR a_b
for i = 0 → (N − 1) do
    L_0(0, i) ← a_i^b
    if i ∈ I^c then  B_n(i, 0) ← ∞
    else  B_n(i, 0) ← 0
end
begin
    for i = 1 → I do
        for φ = 0 → (N − 1) do
            updatellrmap(n, φ)
            if φ is odd then  updatebitmap(n, φ)
        end
    end
    for i = 0 → (N − 1) do
        e_i^d ← B_0(0, i)
        if (B_n(i, 0) + L_n(i, 0)) ≥ 0 then  m̂_i ← 0
        else  m̂_i ← 1
    end
end
```

Algorithm 2: updatebitmap(λ, ϕ)

```
Data: Depth λ and node group φ
if φ is odd then
    for ω = 0 → (2^(n−λ) − 1) do
        B_{λ−1}(ψ, 2ω) ← B_λ(φ − 1, ω) ⊞ [B_λ(φ, ω)
                                         + L_{λ−1}(ψ, 2ω + 1)]
        B_{λ−1}(ψ, 2ω + 1)
            ← B_λ(φ, ω) + B_λ(φ − 1, ω) ⊞ L_{λ−1}(ψ, 2ω)
    end
    if ψ is odd then updatebitmap(λ − 1, ψ)
end
```

Algorithm 3: updatellrmap(λ, ϕ)

```
Data: Depth λ and node group φ
if λ = 0 then return
ψ ← ⌊φ/2⌋
if φ is even then updatellrmap(λ − 1, ψ)
for ω = 0 → (2^(n−λ) − 1) do
    if φ is even then
        L_λ(φ, ω) ← L_{λ−1}(ψ, 2ω) ⊞ [L_{λ−1}(ψ, 2ω + 1)
                                      + B_λ(φ + 1, ω)]
    end
    else
        L_λ(φ, ω) ← L_{λ−1}(ψ, 2ω + 1)
                    + L_{λ−1}(ψ, 2ω) ⊞ B_λ(φ − 1, ω)
    end
end
```

of the SCAN decoder with $I = 1$ for AWGN channels. We observe that by clipping the LLR of already detected bits to $+\infty, -\infty$ we can increase the convergence and polarization rate. Zimmermann et al. [13] observed the same phenomenon in belief propagation decoder for LDPC codes and called it '*belief pushing*'. We also observe that one can convert the SCAN decoder with $I = 1$ into an SC decoder by assigning $B_n(k, 0) = \infty \times \mathrm{sgn}(B_n(k, 0) + L_n(k, 0))$ as we calculate $\{L_n(k, 0)\}_{k=0}^{N-1}$, where $\mathrm{sgn}(.)$ is the sign function. Therefore, we can consider the SC decoder as a particular instance of the more generalized SCAN decoder. We conclude this section by presenting the following proposition.

Proposition 1: The SC decoder is a fast-polarizing instance of the SCAN decoder with $I = 1$.

The proof is provided in the appendix.

V. Performance Results

A. AWGN Channel

We present Monte Carlo simulation results for both an AWGN and an ISI channel. Figure 2 shows the performance of the SCAN decoder on the AWGN channel, based on a maximum of 100,000 codewords for each value of SNR. We observe that the FER performance of Arikan's SC decoder is better than that of the SCAN decoder with $I = 1$, which is expected in light of Proposition 1. The performance of the SCAN decoder improves as we increase the number of iterations I. We observe a gain of 0.22 dB with the SCAN decoder, for both FER and BER, when the number of iteration $I = 8$. We also observe that the BER performance of the SCAN decoder is better than that of the SC decoder for all values of I. We attribute the relatively poor performance of the SC decoder to its more prominent error propagation phenomenon.

B. Partial Response Channels

Figure 3 shows the performance of the proposed SCAN decoder for both the dicode channel and the EPR4 channel. We compare the performance of the SCAN decoder with two existing decoders: an SC decoder with LLR clipping, and a BP decoder. For the SC decoder with LLR clipping, we clip the extrinsic LLR of coded bits to a fixed value γ according to the rule

$$e_d^{(i)} = (-2 \times B_0(0, i) + 1) \times \gamma. \tag{11}$$

Monte Carlo simulations show that $\gamma = 1$ to $\gamma = 3$ produce the best results with this decoder. These values of γ agree to the values reported in [14] in the context of MIMO detection.

From Figure 3 we observe that, for the dicode channel, the performance of the SCAN decoder with $I = 2$ is comparable to the BP decoder with $I = 60$. This highlights the huge complexity reduction of the SCAN decoder ($\sim 96\%$), as the complexity of one iteration of both decoders is the same. Furthermore, the performance of the SCAN decoder improves as we increase I to 4 and 8. Therefore, the SCAN decoder not only improves the performance but it does so with a lower complexity than the BP decoder. Additionally, the SCAN

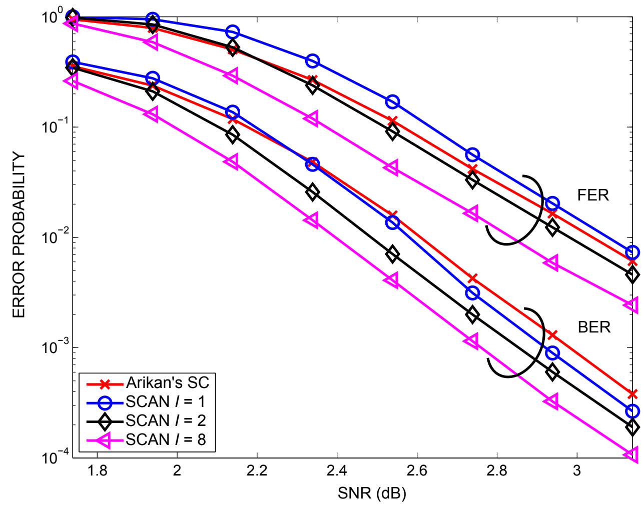

Fig. 2. Performance of the SCAN decoder in AWGN for a polar code of rate 0.7 and $N = 4096$. We have optimized the code using the method presented in [10] for an SNR=2.1387 dB.

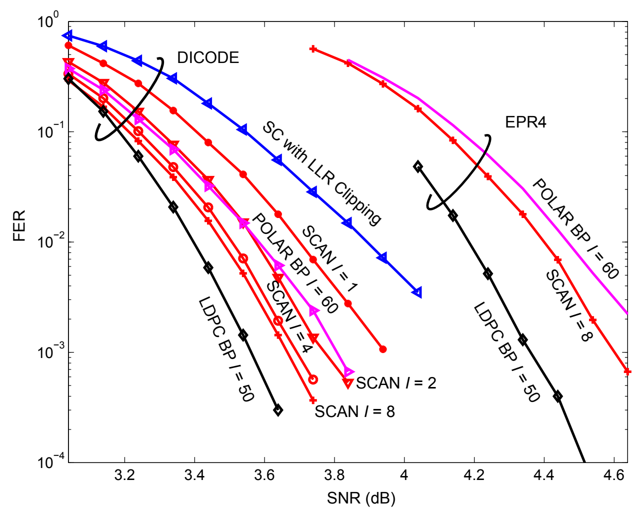

Fig. 3. FER performance of the SCAN decoder in the partial response channels for a polar code of rate 0.7 and $N = 4096$. We have optimized the code using the method in [10] for SNR=2.1387 dB (dicode) and SNR=2.2887 dB (EPR4).

decoder also restores the decoder's recursive nature that we observed in the SC decoder and lost in the BP decoder.

As a benchmark we also include in Figure 3 the performance of a (4096,2868) LDPC code with column weight 3, constructed using [15], with BP decoding. The performance of the LDPC code is comparable to that of the proposed SCAN decoder, especially at low SNR. The gap in performance remains small at high SNR; it is only 0.1 dB at FER = 10^{-3} for the dicode channel. We mention here that the performance gap between the LDPC code and the polar code is larger in terms of BER (not shown here) than FER.

VI. Conclusion

We presented a low-complexity soft-input soft-output successive cancellation decoder for polar codes, making polar codes a suitable candidate for a concatenated coding or turbo equalization scheme. Performance results indicate that the SCAN decoder outperforms the belief propagation decoder with low computational complexity in the context of turbo equalization for both the dicode and the EPR4 channel. In this setting, the FER performance of a polar code with SCAN decoding is comparable to that of a regular LDPC code.

APPENDIX

Proof of Lemma 1:

$$\begin{aligned} W^-(y_0, y_1, z_1|u_0) \\ &= \sum_{u_1} W(y_0, y_1, z_1, u_1|u_0) \\ &= \frac{1}{2}\sum_{u_1} W(y_0|u_0 \oplus u_1)W(y_1|u_1)P(z_1|u_1). \end{aligned} \quad (12)$$

$$\begin{aligned} W^+(y_0, y_1, z_0|u_1) \\ &= \sum_{u_0} W(y_0, y_1, z_0, u_0|u_1) \\ &= \frac{1}{2}W(y_1|u_1)\sum_{u_0} W(y_0|u_0 \oplus u_1)P(z_0|u_0), \end{aligned} \quad (13)$$

where we have used the fact that both the bits u_0, u_1 are equally likely to be 0 or 1. Using Eq. (12) and Eq. (13) with the definition of an LLR, we get Eq. (9) and Eq. (10). ■

Proof of Proposition 1: Consider the problem setup for Eq. (12) and (13). Recall for an SC decoder, we have from [16]

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

where $Z(W)$ is Bhattacharyya parameter of the DMC W defined as

$$Z(W) \triangleq \sum_{y} \sqrt{W(y|0)W(y|1)}. \quad (14)$$

Since, for the SCAN decoder with $I = 1$, the computation for the check-node doesn't change, the relationships for $Z(W^-)$ as described above hold. Therefore, we only need to prove $Z(W^+) \geq Z(W)^2$.

$$\begin{aligned} Z(W^+) &= \sum_{y_0,y_1,z_0} \sqrt{W^+(y_0, y_1, z_0|0)W^+(y_0, y_1, z_0|1)} \\ &= \frac{1}{2}Z(W) \times A(W, P), \end{aligned} \quad (15)$$

where

$$\begin{aligned} A(W, P) = \sum_{y_0,z_0} &\left(\sum_{u_0} W(y_0|u_0)P(z_0|u_0)\right) \\ &\times \left(\sum_{u_0'} W(y_0|u_0' \oplus 1)P(z_0|u_0')\right). \end{aligned}$$

From Lemma 3.15 in [16], we have

$$A(W, P) \geq 2\sqrt{Z(W)^2 + Z(P)^2 - Z(W)^2Z(P)^2}. \quad (16)$$

Using Eq. (16) in Eq.(15), we get

$$\begin{aligned} Z(W^+) &= Z(W)^2\sqrt{1 + Z(P)^2\left(\frac{1}{Z(W)^2} - 1\right)}, \\ &\geq Z(W)^2 \end{aligned}$$

as by definition $0 \leq Z(P), Z(W) \leq 1$. ■

REFERENCES

[1] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 3051 –3073, July 2009.

[2] C. Douillard, M. Jzquel, C. Berrou, D. Electronique, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, 1995. [Online]. Available: http://dx.doi.org/10.1002/ett.4460060506

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, May 1993, pp. 1064 –1070 vol.2.

[4] E. Arkan, "A performance comparison of polar codes and reed-muller codes," *Communications Letters, IEEE*, vol. 12, no. 6, pp. 447 –449, June 2008.

[5] N. Goela, S. Korada, and M. Gastpar, "On LP decoding of polar codes," in *Information Theory Workshop (ITW), 2010 IEEE*, 30 2010-Sept. 3 2010, pp. 1 –5.

[6] I. Tal and A. Vardy, "List decoding of polar codes," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, 31 2011-Aug. 5 2011, pp. 1 –5.

[7] K. Niu and K. Chen, "Stack decoding of polar codes," *Electronics Letters*, vol. 48, no. 12, pp. 695 –697, 7 2012.

[8] I. Tal and A. Vardy, "How to construct polar codes," *CoRR*, vol. abs/1105.6164, 2011.

[9] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *Communications Letters, IEEE*, vol. 13, no. 7, pp. 519 –521, July 2009.

[10] P. Trifonov and P. Semenov, "Generalized concatenated codes based on polar codes," in *Wireless Communication Systems (ISWCS), 2011 8th International Symposium on*, Nov. 2011, pp. 442 –446.

[11] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *Information Theory, IEEE Transactions on*, vol. 20, no. 2, pp. 284 – 287, March 1974.

[12] E. Arikan and E. Telatar, "On the rate of channel polarization," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, 28 2009-july 3 2009, pp. 1493 –1495.

[13] E. Zimmermann and G. Fettweis, "Reduced complexity ldpc decoding using forced convergence," in *In Proceedings of the 7th International Symposium on Wireless Personal Multimedia Communications (WPMC04), Abano*, 2004, p. 15.

[14] Y. de Jong and T. Willink, "Iterative tree search detection for MIMO wireless systems," *Communications, IEEE Transactions on*, vol. 53, no. 6, pp. 930 – 935, June 2005.

[15] R. M. Neal, "Software for Low Density Parity Check (LDPC) codes," Toronto, 2006. [Online]. Available: http://www.cs.utoronto.ca/~radford/ftp/LDPC-2006-02-08/index.html

[16] S. B. Korada, "Polar codes for channel and source coding," Ph.D. dissertation, Lausanne, 2009, prix Asea Brown Boveri Ltd (ABB) (2010). [Online]. Available: http://library.epfl.ch/theses/?nr=4461