Directed Search Decoding of Polar Codes with Reed-Solomon Kernel

Ruslan Morozov, Peter Trifonov
St. Petersburg Polytechnic University
Email: {rmorozov,petert}@dcn.icc.spbstu.ru

***Abstract*—The directed search algorithm, introduced originally for decoding of polar codes with Arıkan kernel, is generalized to the case of a Reed-Solomon kernel. The algorithm relies heavily on the knowledge of symbol subchannel error probabilities of the polarizing transformation. A method to approximate these probabilities is suggested. The proposed method results in almost the same complexity and performance as in the case of the decoder using exact values of error probabilities.**

I. Introduction

Polar codes are the first class of codes shown to achieve Shannon capacity of a wide class of channels with efficient encoding and decoding algorithms [1]. However, their finite length performance appears to be quite poor both due to suboptimality of the successive cancellation (SC) decoding algorithm and low minimum distance of polar codes. The first problem was addressed in [2], [3], where list and stack decoding algorithms were proposed. The second problem can be solved by employing polar subcodes [4] or using a kernel with higher polarization rate [5].

In this paper we present a generalization of directed search decoding algorithm to the case of polar codes with Reed-Solomon kernel, and a method to approximate error probabilities in the subchannels of the polarizing transformation, that are needed for directed search decoding. Furthermore, we show that polar codes with Reed-Solomon kernel and dynamic frozen symbols substantially outperform those with Arıkan kernel, as well as LDPC codes.

The paper is organized as follows. In section II polar codes with Reed-Solomon kernel are described, and directed search decoding algorithm for these codes is presented in section III. In section IV Gaussian approximation version for non-binary polar codes is shown, which is used by heuristic function of directed search algorithm. In section V non-binary polar subcodes are proposed. In sections VI and VII there are results and conclusions.

II. Background

A. Polar Codes with Reed-Solomon Kernel

Consider a $q\times q$ Reed-Solomon kernel K over $\mathbb{F}_q$ given by $K_{ij}=\alpha_j^{q-1-i}$, where $\alpha_j, j=0\dots q-1$ are distinct elements of $\mathbb{F}_q$. Let $G_{q,l}=B_{q,l}K^{\otimes l}$, where $B_{q,l}$ is a permutation matrix that reverses order of q-ary digits in the expansion $i=\sum_{j=0}^{l-1} i_j q^j, 0\leq i_j<q$. It was shown in [6] that the capacities of symbol subchannels $\widetilde{W}_n^{(i)}(y_0^{n-1},u_0^{i-1}|u_i)$ induced by $G_{q,l}$ converge with $l\to\infty$ to 0 or 1 symbols per channel use, where $n=q^l$ and a_s^t denotes $(a_s,a_{s+1},\dots,a_t)$. Polar code is defined as $\mathcal{C}=\{fG_{q,l}|f_i=0,i\in\mathcal{F}\}$, where $\mathcal{F}$ is the set of frozen subchannels and $\{0,\dots,n-1\}\setminus\mathcal{F}$ is the set of positions of information symbols. $\mathcal{F}$ can be defined either as a set of subchannels with low capacity or high error probability.

The successive cancellation decoding algorithm makes decisions $\widehat{f}_i=\begin{cases}\arg\max_{f_i\in\mathbb{F}_q}\widetilde{W}_n^{(i)}(f_0^i|y_0^{n-1}), & i\notin\mathcal{F}\\ 0, & i\in\mathcal{F},\end{cases}$ where

$$\widetilde{W}_n^{(qj+i)}(f_0^{qj+i}|y_0^{n-1})=\sum_{f_{qj+i+1}^{qj+q-1}\in\mathbb{F}_q^{q-i-1}}\prod_{s=0}^{q-1}\widetilde{W}_{\frac{n}{q}}^{(j)}((f_{qt}^{qt+q-1}K)_s,t\in N_0^j|y_{\frac{n}{q}s}^{\frac{n}{q}s+\frac{n}{q}-1}), \tag{1}$$

$\widetilde{W}_1^{(0)}(x|y)=W(x|y)$, $W(x|y)$ is the original channel, $N_a^b=\{a,a+1,\dots,b\}$. An efficient method for evaluation of this expression was suggested in [7]. Let $q=2^m$ and consider the Arıkan matrix $A_m=B_{2,m}\begin{pmatrix}1&0\\1&1\end{pmatrix}^{\otimes m}$. The output $c_0^{q-1}=f_0^{q-1}K$ of the transformation given by the Reed-Solomon kernel K can be represented as $c_0^{q-1}=u_0^{q-1}A_m$, where $u_0^{q-1}=f_0^{q-1}KA_m^{-1}$. Then the distributions $\widetilde{W}_q^{(i)}(f_0^i|y_0^{q-1})$ can be derived from distributions $W_q^{(j)}(u_0^j|y_0^{q-1}), i\leq j<q$, which can be computed using the Arıkan recursion

$$W_N^{(2i)}(u_0^{2i}|y_0^{N-1})=\sum_{u_{2i+1}\in\mathbb{F}_q}W_{\frac{N}{2}}^{(i)}(u_{0,e}^{2i+1}\oplus u_{0,o}^{2i+1}|y_0^{\frac{N}{2}-1})W_{\frac{N}{2}}^{(i)}(u_{0,o}^{2i+1}|y_{\frac{N}{2}}^{N-1})$$

$$W_N^{(2i+1)}(u_0^{2i+1}|y_0^{N-1})=W_{\frac{N}{2}}^{(i)}(u_{0,e}^{2i+1}\oplus u_{0,o}^{2i+1}|y_0^{\frac{N}{2}-1})W_{\frac{N}{2}}^{(i)}(u_{0,o}^{2i+1}|y_{\frac{N}{2}}^{N-1}),$$

where $a_{s,e}^t$ ($a_{s,o}^t$) is subvector of a_s^t with even (odd) indices. In particular, for the case of $q=4,\alpha_0=\alpha^2,\alpha_1=\alpha,\alpha_2=$

$1, \alpha_3 = 0$, where α is a primitive element of $\mathbb{F}_4$, one obtains

$$\widetilde{W}_4^{(0)}(f_0|y_0^3) = W_4^{(0)}(f_0|y_0^3) \quad (2)$$

$$\widetilde{W}_4^{(1)}(f_0^1|y_0^3) = \sum_{u_1 \in \mathbb{F}_4} W_4^{(2)}(f_0, u_1, \alpha^2 f_0 + \alpha u_1 + f_1|y_0^3) \quad (3)$$

$$\widetilde{W}_4^{(2)}(f_0^2|y_0^3) = W_4^{(2)}(f_0, \sum_{i=0}^{2} f_i, f_0 + \alpha^2 f_1 + \alpha f_2|y_0^3) \quad (4)$$

$$\widetilde{W}_4^{(3)}(f_0^3|y_0^3) = W_4^{(3)}(f_0, \sum_{i=0}^{2} f_i, f_0 + \alpha^2 f_1 + \alpha f_2, f_3|y_0^3). \quad (5)$$

III. Directed Search Decoding of Polar Codes with Reed-Solomon Kernel

The SC decoding algorithm is not able to correct errors which occur at early phases of the decoding process. Therefore, its performance is quite far from that of a maximum likelihood decoder. This problem can be avoided by employing the directed search decoding algorithm that was proposed for Arıkan kernel in [8]. In this paper it is generalized to the case of Reed-Solomon kernel.

A. High-Level Description

Paths $f_0^{\phi-1}$ together with their metrics $M(f_0^{\phi-1}|y_0^{n-1})$ are stored in a priority queue. At each iteration, the path $f_0^{\phi-1}$ with the highest metric is selected for continuation and removed from the queue. If $\phi \notin \mathcal{F}$, it is extended into q paths f_0^{ϕ} corresponding to distinct values of $f_\phi \in \mathbb{F}_q$. Otherwise, if $\phi \in \mathcal{F}$, it is extended into a single path f_0^{ϕ} by given frozen value of f_ϕ. For each newly obtained path f_0^{ϕ} the metric $M(f_0^{\phi}|y_0^{n-1})$ is calculated and the pair $\left[f_0^{\phi}, M(f_0^{\phi}|y_0^{n-1})\right]$ is inserted in the queue. The number of paths grows exponentially, so parameters L and Θ are introduced to restrict it. L is the maximum number of observed paths with given length ϕ. If this value is reached, all paths shorter than ϕ are killed and thereby no more paths of length ϕ are observed. Θ is the capacity of the queue: when there is no place for new paths, the worst paths are killed. The decoding terminates as soon as a path of length n is selected. The pseudo code for the high-level $Decode$ function is presented in Fig.1 (Alg. 1).

B. The Metric

Following [8], we propose to define for every $0 \leq \phi < n$

$$\widetilde{W}_n^{(n-1)}(f_0^{n-1}|y_0^{n-1}) = \\ \widetilde{W}_n^{(\phi-1)}(f_0^{\phi-1}|y_0^{n-1}) \prod_{j=\phi}^{n-1} \widetilde{W}_n^{(j)}(f_j|f_0^{j-1}, y_0^{n-1}).$$

At each iteration the first term $\widetilde{W}_n^{(\phi)}(f_0^{\phi-1}|y_0^{n-1})$ can be obtained from (1). The second term can be replaced by its expectation over all possible sequences y_0^{n-1}. Assuming that f_0^{n-1} is the correct path, one obtains

$$\Omega(\phi) = E_{y_0^{n-1}}\left[\prod_{j=\phi}^{n-1} \widetilde{W}_n^{(j)}(f_j|u_0^{j-1}, y_0^{n-1})\right] = \prod_{j=\phi}^{n-1}(1-\pi_j), \quad (6)$$

where π_j is the symbol error probability in subchannel $\widetilde{W}_n^{(j)}(f_j|y_0^{n-1}, f_0^{j-1})$. Hence, the path metric is defined as $M(f_0^{\phi-1}|y_0^{n-1}) = \widetilde{W}_n^{(\phi-1)}(f_0^{\phi-1}|y_0^{n-1})\Omega(\phi)$.

```
Algorithm 1: Decode
Input: $y_0^{q^l-1}, L, \Theta$
$Push(1, \emptyset);\ Size \leftarrow 1$
$P_0 \leftarrow GetArrayPointerP(0, j)$
for $\beta \in \{0, \dots, q^l - 1\}$ do
    for $a \in \mathbb{F}_q$ do $P_0[\beta][a] \leftarrow W(a|y_\beta)$
while true do
    $i \leftarrow PopMax()$
    if $q|\phi_i$ then return $GetArrayPointerC(0, i)$
    $N_{\phi_i} \leftarrow N_{\phi_i} + 1$ //count paths of given length
    $RecursivelyCalcP(i, l, \phi_i)$
    if $\phi \in \mathcal{F}$ then $ContinuePathFrozen(i, l, \phi_i)$
              else $ContinuePathsUnfrozen(i, l, \phi_i)$
    if $q|(\phi_i + 1)$ then $RecursivelyUpdateC(i, l, \phi_i)$
```

```
Algorithm 2: RecursivelyCalcP
Input: $i, \lambda, \phi$
if $\lambda = 0$ then return
if $q|\phi$ then $RecursivelyCalcP(i, \lambda - 1, \phi/q)$
$P_{\lambda-1} \leftarrow GetArrayPointerP(\lambda - 1, i)$
$P_\lambda \leftarrow GetArrayPointerP(\lambda, i)$
$C_\lambda \leftarrow GetArrayPointerC(\lambda, i)$
for $\beta \in \{0, \dots, q^{l-\lambda} - 1\}$ do
    $P_\lambda[\beta] \leftarrow KqDec(\phi \bmod q, C_\lambda[\beta], (P_{\lambda-1}[t])_{t=q\beta}^{q\beta+q-1})$
```

```
Algorithm 3: RecursivelyUpdateC
Input: $i, \lambda, \phi$
$\psi \longleftarrow \lfloor \phi/q \rfloor$
if $\lambda = 0$ then return
$C_\lambda \leftarrow GetArrayPointerC(\lambda, i)$
$C_{\lambda-1} \leftarrow GetArrayPointerC(\lambda - 1, i)$
for $\beta \in \{0, \dots, q^{l-\lambda}\}$ do
    $B_0^{q-1} \leftarrow C_\lambda[\beta]K_q$
    if $\lambda = 1$ then $C_{\lambda-1}[\beta] \leftarrow B_0^{q-1}$
    else for $j \in N_0^{q-1}$ do $C_{\lambda-1}[q\beta + j][\psi \bmod q] \leftarrow B_j$
if $q|(\psi + 1)$ then $RecursivelyUpdateC(i, \lambda - 1, \psi)$
```

Fig. 1. Pseudo code of the high-level functions

However, one should compute efficiently error probabilities in subchannels $\widetilde{W}_n^{(j)}(y_0^{n-1}, f_0^{j-1}|f_j)$. In the case of Arıkan kernel (i.e. channels $W_n^{(j)}(u_j|y_0^{n-1}, u_0^{j-1})$), the values π_j can be computed using density evolution [9], [10] or Gaussian approximation [11]. The latter approach is generalized to the case of Reed-Solomon kernel in section IV.

C. Paths Continuation

1) Probabilities calculation. For computing the metric $M(f_0^{\phi}|y_0^{n-1})$ one needs to obtain the values of (1). At the intermediate layers $0 < \lambda \leq l$ function $RecursivelyCalcP$ is called (see Fig.1, Alg. 2), which computes $\widetilde{W}_{\Lambda}^{(\phi)}(f_0^{\phi}|y_0^{\Lambda-1})$, that are stored in array $P_{\lambda}[\beta][f_{\phi}]$ of size q for each value of $f_{\phi} \in \mathbb{F}_q$, from $q(j+1)$ probabilities $W_{\Lambda/q}^{(j)}((f_{qt}^{qt+q-1}K)_s, t \in N_0^j|y_{s\Lambda/q}^{(s+1)\Lambda/q-1})$, $j = \phi - q\lfloor\phi/q\rfloor$, $\Lambda = q^{\lambda}$, $0 \leq s < q$, that are stored in array $P_{\lambda-1}[\beta]$ (q probabilities for each value of t), as described in [7] (for the case of $q = 2^2$ see (2)–(5)). Here, function $KqDecoder(\varphi, C, P)$ returns the distribution of φ-th information symbol of a Reed-Solomon kernel transformation given previous information symbols C and distributions of code symbols P, and i is input path index.

2) Decisions propagation. At phases $qj-1$ the vector $(f_{qj-q}, \ldots, f_{qj-1})$ consisting of the last q symbols of the corresponding path is multiplied by K_q and propagated to layer $\lambda - 1$, similarly to function $RecursivelyUpdateC$ introduced in [2]. Arrays $C_{\lambda}[\beta]$, $0 \leq \beta < q^{l-\lambda+1}$ contain symbols on the λ-th layer of polarization. The resulting function is shown in Alg. 3 in Fig.1.

3) Memory Management. The lazy copying approach is implemented to reduce memory copying. Described above arrays $P_{\lambda}[\beta]$ and $C_{\lambda}[\beta]$ are different for different path indices i. Function $GetArrayPointerC(\lambda, i)$ translates path index i to corresponding branch index β and returns pointer to $C_{\lambda}[\beta]$. Arrays $P_{\lambda}[\beta]$ are indexed in the same manner.

The worst-case complexity of this algorithm is $O(Ln \log n)$ operations that correspond to evaluation of (1).

IV. Gaussian Approximation

Assume that zero codeword is transmitted. In order to construct function $\Omega(\phi)$ defined in (6), one needs to calculate symbol subchannel error probabilities, i.e.

$$\pi_{\phi} = E_{y_0^{n-1}}\left[\widetilde{W}_n^{(\phi)}(f_{\phi} \neq 0|f_0^{\phi-1} = 0_0^{\phi-1}, y_0^{n-1})\right],$$

where $n = q^l$ is the code length. To the best of our knowledge, there are still no techniques for accurate evaluation of these values.

Figure 2 presents the distributions of log-likelihood ratios $LLR(f_i = x) = \log \frac{W_4^{(i)}\{f_i=0|f_0^{i-1}=0_0^{i-1}, y_0^3\}}{W_4^{(i)}\{f_i=x|f_0^{i-1}=0_0^{i-1}, y_0^3\}}, x \neq 0$, for the case of $q = 4$. It can be seen that these distributions are quite close to Gaussian ones. Hence, we suggest to extend Gaussian approximation techniques [12], [11] to the case of polar codes with Reed-Solomon kernel.

Let us consider transmission of the binary image of the codewords of the polar code over $\mathbb{F}_q, q = 2^m$, over an AWGN channel with BPSK modulation characterized by noise standard deviation σ. We propose to approximate the subchannels $\widetilde{W}_{q^s}^{(\phi)}(f_{\phi}|y_0^{q^s-1}, f_0^{\phi-1}), 1 \leq s \leq l$, also as BPSK-modulated AWGN channels with the same error probability. Hence, these subchannels can be characterized by the corresponding noise standard deviations $\sigma_s^{(\phi)}$. One needs to derive an efficient method for computing $\sigma_s^{(iq+j)}$ from $\sigma_{s-1}^{(i)}$, where $\sigma_0^{(0)} = \sigma, 0 \leq j < q$.

For a BPSK-modulated AWGN channel the bit error probability is given by $p(\sigma) = Q\left(1/\sigma\right)$, and the error probability in a binary image of a symbol from $\mathbb{F}_{2^m}$ is given by $P_m(\sigma) = 1 - (1 - p(\sigma))^m$. We propose to set $\sigma_s^{(iq+j)} = P_m^{-1}(\psi_j(\sigma_{s-1}^{(i)})), 1 \leq s \leq l$,where $\psi_j(\sigma)$ is the probability of a maximum-likelihood decoder for a binary image of $(q, q-j, j+1)$ Reed-Solomon code R_j returning a codeword $(0, \ldots, 0, f_j, \ldots, f_{q-1})K$ with $f_j \neq 0$ for AWGN channel with noise standard deviation σ.

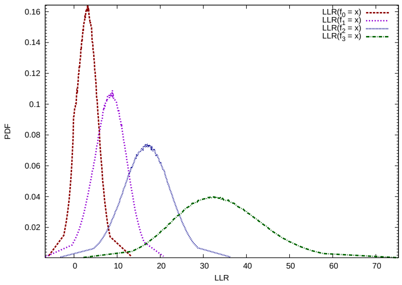

Fig. 2. Distribution of LLRs at the output of the kernel decoder

Furthermore, we propose to approximate $\psi_j(\sigma)$ using the tangential sphere bound (TSB) [13], [14]. Computing the TSB for the maximum-likelihood decoding error probability for a linear binary code requires knowledge of its weight distribution. More specifically, one needs to know the number of non-zero codewords of different weight, which correspond to possible decoder error events. In the considered case, one is interested only in such error events, which correspond to $f_j \neq 0$. Hence, one needs to use the weight distribution of the binary image of $R_j \setminus R_{j+1}$, where R_j is the code generated by the first i rows of matrix K_q.

The Hamming weight enumerator for R_j is given by $A^{(j)}(X) = 1 + \sum_{i=j+1}^{q} A_{ji}X^i$, where

$$A_{ji} = \binom{q}{i}(q-1)\sum_{t=0}^{i-j-1}(-1)^t\binom{i-1}{t}q^{i-t-j-1}. \quad (7)$$

Following [15], we consider a random mapping from $\mathbb{F}_{2^m} \setminus \{0\}$ to $\mathbb{F}_2^m \setminus \{0\}$. Then the average weight enumerator for a binary image of a symbol in $\mathbb{F}_{2^m} \setminus \{0\}$ is given by

$$B_m(X) = \frac{1}{2^m - 1}\sum_{i=1}^{m}\binom{m}{i}X^i,$$

and the weight enumerator of the binary image of $R_j \setminus R_{j+1}$ is given by

$$C^{(j)}(X) = A^{(j)}(B_m(X)) - A^{(j+1)}(B_m(X)),$$

where $A^{(q)}(X) = 1$. This weight enumerator should be used in the TSB (or some other bound) for computing an approximate value of $\psi_j(\sigma)$.

V. Codes with Dynamic Frozen Symbols

Similarly to the case of codes with Arıkan kernel, polar codes with Reed-Solomon kernel suffer from low minimum distance. In order to obtain codes which can provide good performance under the proposed decoding algorithm, we propose to extend the construction of polar codes with dynamic frozen

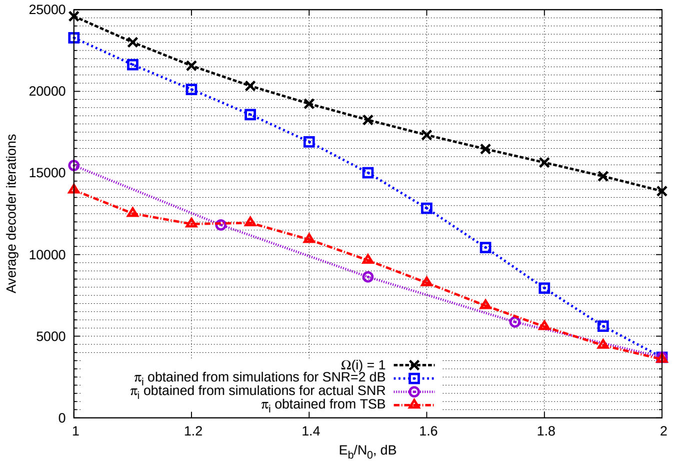

Fig. 3. Average number of iterations performed by the decoding algorithm ($L = 64, \Theta = 16384$, $n = 1024$, $L = 64$) for different $\Omega(i)$

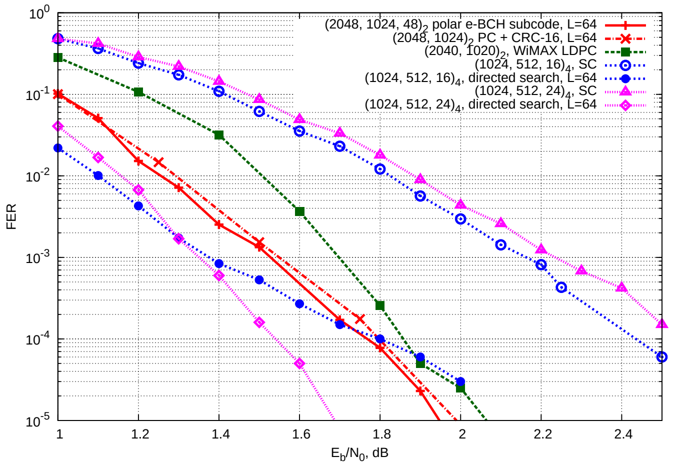

Fig. 4. Performance of polar codes with Reed-Solomon kernel and dynamic frozen channels

symbols introduced in [8] to the case of Reed-Solomon kernel. Namely, one can set the values of some frozen symbols not statically to zero, but to some linear combination of other non-frozen symbols. The particular expressions (*dynamic freezing constraints)* for these dynamic frozen symbols can be derived from the equation $fG_{q,l}H^T = 0$, where H is a check matrix of q-ary extended BCH (e-BCH) code with given minimum distance d. The constraints can be brought to the form

$$f_{t_i} = -\sum_{s=0}^{t_i-1} V_{i,s} f_s.$$

Then f_{t_i} are dynamic frozen symbols.

One can statically freeze (i.e. set $f_i = 0$) $q^l - \mathrm{rank}(H) - k$ additional subchannels with the highest error probability π_i, where k is the dimension of the polar code being constructed. Unfortunately, the method described in section IV does not provide sufficiently precise estimates of π_i for the purpose of code construction. Therefore, one needs to use simulations in order to find symbol subchannels with the highest error probability.

VI. Numerical Results

We consider transmission of a binary image of the code symbols of $(n = 1024, k = 512, d)_4$ codes, where n is the code length, k is the dimension, d is the minimum distance.

Fig. 3 presents the average number of iterations performed by the directed search decoding algorithm for the case of $\Omega(i) = 1$, as well as $\Omega(i)$ given by (6), where π_i are obtained by simulations either for fixed or actual SNR, as well as using the proposed TSB-based Gaussian approximation. Observe that the first case corresponds to the stack decoding algorithm [3]. It can be seen that the proposed method for computing π_i is sufficiently accurate, so that the average decoding complexity approaches that for the case of π_i obtained by simulations, and is substantially smaller compared to the case of $\Omega(i)$ constructed in a channel-independent way. At the same time on Fig. 4 one can see that the proposed approximation results in negligible performance degradation (not more than 0.05 dB) in the low-SNR region compared to the case of $\Omega(i)$ obtained from simulation data.

Fig. 4 presents the performance of a classical $(1024, 512, 16)_4$ polar code with Reed-Solomon kernel, and a $(1024, 512, 24)_4$ polar code with Reed-Solomon kernel and dynamic frozen symbols obtained as a subcode of $(1024, 913, 24)$ extended BCH code. It can be seen that the classical polar code exhibits an error floor under the directed search decoding algorithm. It can be also seen that $(1024, 512, 24)_4$ code outperforms the $(2048, 1024, 48)_4$ binary polar code with dynamic frozen symbols, polar code with CRC and $(2040, 1020)$ WiMAX LDPC code.

VII. Conclusion

In this paper an extension of the directed search decoding algorithm to the case of polar codes with Reed-Solomon kernel was proposed. The method to approximate error probabilities in symbol subchannels induced by the polarizing transformation was introduced. It is not sufficiently precise for the purpose of code construction, so that simulations still have to be used in order to find good codes. However, the accuracy is sufficient in order to allow efficient directed search decoding, where employing simulation data is not practical.

Similarly to the case of Arıkan kernel, polar codes with Reed-Solomon kernel exhibit an error floor due to poor minimum distance. In order to eliminate this effect, a construction of q-ary polar codes with Reed-Solomon kernel and dynamic frozen symbols was introduced. The obtained codes together with the proposed decoding algorithm were shown to substantially outperform binary polar codes with dynamic frozen symbols, binary LDPC codes, and classical polar codes with Reed-Solomon kernel.

References

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

[2] I. Tal and A. Vardy, “List decoding of polar codes,” in *Proceedings of IEEE International Symposium on Information Theory*, 2011, pp. 1–5.

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

[4] P. Trifonov and V. Miloslavskaya, “Polar subcodes,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 254–266, February 2016.

[5] S. B. Korada, E. Sasoglu, and R. Urbanke, “Polar codes: Characterization of exponent, bounds, and constructions,” *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6253–6264, December 2010.

[6] R. Mori and T. Tanaka, “Channel polarization on q-ary discrete memoryless channels by arbitrary kernels,” in *Proceedings of IEEE International Symposium on Information Theory*, 2010.

[7] P. Trifonov, “Binary successive cancellation decoding of polar codes with Reed-Solomon kernel,” in *Proceedings of IEEE International Symposium on Information Theory*, Honolulu, 2014, pp. 2972 – 2976.

[8] P. Trifonov and V. Miloslavskaya, “Polar codes with dynamic frozen symbols and their decoding by directed search,” in *Proceedings of IEEE Information Theory Workshop*, September 2013, pp. 1 – 5.

[9] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Comm. Letters*, vol. 13, no. 7, 2009.

[10] I. Tal and A. Vardy, “How to construct polar codes,” *IEEE Transactions on Information Theory*, vol. 59, no. 10, pp. 6562–6582, October 2013.

[11] P. Trifonov, “Efficient design and decoding of polar codes,” *IEEE Trans. on Communications*, vol. 60, no. 11, pp. 3221 – 3227, Nov. 2012.

[12] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, Feb. 2001.

[13] G. Poltyrev, “Bounds on the decoding error probability of binary linear codes via their spectra,” *IEEE Transactions on Information Theory*, vol. 40, no. 4, July 1994.

[14] I. Sason and S. Shamai, “Performance analysis of linear codes under maximum-likelihood decoding: A tutorial,” *Foundations and Trends in Communications and Information Theory*, vol. 3, no. 1–2, pp. 1–222, 2006. [Online]. Available: http://dx.doi.org/10.1561/0100000009

[15] Q. Zhuang, X. Ma, and A. Kavcic, “Bounds on the ML decoding error probability of RS-coded modulation over AWGN channels,” *CoRR*, vol. abs/1401.5305, October 2014. [Online]. Available: http://arxiv.org/abs/1401.5305