# Sequential Decoding of Reed-Solomon Codes

Vera Miloslavskaya, Peter Trifonov
Saint-Petersburg State Polytechnic University
Email: {veram,petert}@dcn.icc.spbstu.ru

***Abstract*—The problem of efficient soft-decision decoding of Reed-Solomon codes is considered. Low-complexity sequential algorithm was recently proposed for decoding of polar codes. A generalization of this algorithm to the case of Reed-Solomon codes, represented as polar codes with dynamic frozen symbols, is proposed. Simplification of the proposed decoding algorithm to the case of transmission of binary image of Reed-Solomon code is derived.**

## I. Introduction

Soft-decision decoding of Reed-Solomon codes is a long-standing problem, which is still far from being completely solved. On the other hand, polar codes were recently shown to be able to achieve the capacity of a wide class of communication channels [1]. Decoding algorithms for polar codes with very low complexity are available.

It was shown in [2] that any linear code of length $2^m$ can be represented as a polar one with dynamic frozen symbols. This enables application of polar code decoding techniques to other classes of codes. The successive cancellation algorithm, which is the classical decoding method for polar codes, provides quite poor performance. This problem was addressed in [3], where a list decoding algorithm for polar codes was introduced. It was shown in [4], [2] that the same performance can be achieved with much smaller complexity by employing a stack decoding algorithm.

In this paper a generalization of the decoding algorithm suggested in [5] for polar codes to the case of extended Reed-Solomon codes is introduced. The algorithm becomes particularly efficient in the case of transmission of a binary image of the Reed-Solomon code.

## II. Background

### A. Reed-Solomon codes as polar codes

$(n = 2^m, k)$ polar code is a linear block code generated by $k$ rows of matrix $G_n = B_n A^{\otimes m}$, where $A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $\otimes m$ denotes $m$-times Kronecker product of a matrix with itself, $B_n$ is $n \times n$ bit reversal permutation matrix. By $a_i^j$ we will denote sequence $(a_i, a_{i+1}, \ldots, a_j)$. Any codeword of a polar code can be represented as $c_0^{n-1} = u_0^{n-1} G_n$, where $u_0^{n-1}$ is input sequence, such that $u_i = 0$, $i \in \mathcal{F}$, where $\mathcal{F} \subset \{0, \ldots, n-1\}$ is the set of $n-k$ indices of frozen bit subchannels. The remaining $k$ elements of $u_0^{n-1}$ are set to the information symbols.

Polar codes are not subject of this paper. Instead, we consider application of the techniques developed in the area of polar coding to the problem of decoding Reed-Solomon codes over $\mathbb{F}_{2^m}$. It was suggested in [2] to set $u_i$, $i \in \mathcal{F}$, to some linear function of $u_0^{i-1}$ (dynamic frozen symbols). This enables one to represent any linear code of length $n = 2^m$ as polar one.

Indeed, since $G_n$ is invertible matrix, any vector of length $n$ can be represented as $c_0^{n-1} = u_0^{n-1} G_n$. Let $H$ be a check matrix of some $(n, k)$ linear code $C$ over $\mathbb{F}_q$. In order to obtain $c_0^{n-1} \in C$ one needs to ensure that $u_0^{n-1} G_n H^T = 0$. By applying elementary row operations to matrix $H$, one can obtain matrix $V = QHG_n^T$, such that its rows end[1] in distinct columns. Let $\tau(s)$ be the index of column in which the $s$-th row ends. Let us assume without loss of generality that $V_{s,\tau(s)} = -1$. Therefore, one obtains dynamic freezing constraints

$$u_{\tau(s)} = \sum_{0 \le j < \tau(s)} V_{s,j} u_j, \quad 0 \le s < n-k. \tag{1}$$

This enables successive cancellation (SC) decoding of arbitrary linear block codes. However, the set of frozen symbols may not coincide with the set of bit subchannels with smallest capacities, induced by polarizing transformation $G_n$. This results in rather poor performance of SC decoding.

$(n = 2^m, k, 2^m - k + 1)$ extended Reed-Solomon code is defined as the set of vectors $(c_0, \ldots, c_{n-1})$, where $c_i = f(a_i)$, $a_i$ are distinct elements of $\mathbb{F}_{2^m}$, and $\deg f(x) < k$. The above described approach can be used to represent extended Reed-Solomon codes as polar codes with dynamic frozen symbols. It was shown in [6] that in the case of $a_i$ arranged in their vector representation in the standard bit order, the set of dynamic frozen symbols for $(n, k)$ Reed-Solomon code is given by $\mathcal{F} = \{0, \ldots, 2^m - 1\} \setminus \{2^m - 1 - r_m(i) | 0 \le i < k\}$, where $r_m(i)$ is a $m$-bit integer obtained by reversing the bits of integer $i$. Observe that this set does not depend on the basis of $\mathbb{F}_{2^m}$ being used.

### B. Decoding of polar codes

The decoding problem for polar codes consists in finding $\widehat{u}_0^{n-1} = \arg\max_{u_0^{n-1}} P(u_0^{n-1} | y_0^{n-1})$, where maximization is performed over the set of vectors $u_0^{n-1}$ satisfying freezing constraints. The SC decoding algorithm successively computes estimates

$$\widehat{u}_i = \begin{cases} \arg\max_{u_i} P(\widehat{u}_0^{i-1}, u_i | y_0^{n-1}), & i \notin \mathcal{F}, \\ \sum_{0 \le j < i} V_{s,j} \widehat{u}_j, & \text{otherwise}, \end{cases} \tag{2}$$

[1]The $s$-th row ends in the $j$-th column if $V_{s,j} \neq 0$ and $V_{s,j'} = 0$ for all $j' > j$.

where $\tau(s) = i$, $i$ is decoding phase, and

$$P(u_0^{2i}|y_0^{n-1}) = \sum_{u_{2i+1}} P(u_{0,e}^{2i+1} \oplus u_{0,o}^{2i+1}|y_0^{\frac{n}{2}-1}) P(u_{0,o}^{2i+1}|y_{\frac{n}{2}}^{n-1}), \tag{3}$$

$$P(u_0^{2i+1}|y_0^{n-1}) = P(u_{0,e}^{2i+1} \oplus u_{0,o}^{2i+1}|y_0^{\frac{n}{2}-1}) P(u_{0,o}^{2i+1}|y_{\frac{n}{2}}^{n-1}), \tag{4}$$

where $u_{0,e}^i$ and $u_{0,o}^i$ are subsequences of $u_0^i$ consisting of elements with even and odd indices, respectively. This method was introduced originally in the context of binary polar codes. However, it can be employed in the case of codes over $\mathbb{F}_{2^m}$ too. In this case maximization over $u_i$ in (2) and summation over $u_{2i+1}$ in (3) should be performed over $\mathbb{F}_{2^m}$. An efficient implementation of this approach can be obtained using techniques introduced in [7]. Namely, given probability distributions represented as arrays $\chi^{(0)}$: $\chi_r^{(0)} = P\left(u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1}, r|y_0^{n/2-1}\right)$ and $\chi^{(1)}$: $\chi_r^{(1)} = P\left(u_{0,o}^{2i-1}, r|y_{n/2}^{n-1}\right), r \in \mathbb{F}_{2^m}$, one can compute

$$\chi = \mathcal{H}^{-1}\left(\mathcal{H}(\chi^{(0)}) \odot \mathcal{H}(\chi^{(1)})\right), \tag{5}$$

where $\chi_r = P(u_0^{2i-1}, r|y_0^{n-1})$, $\odot$ denotes component-wise product of two vectors. Here $f = \mathcal{H}(\chi)$ denotes Hadamard transform, which is given by

$$f_r = \sum_{j \in \mathbb{F}_{2^m}} (-1)^{r \boxdot j} \chi_j, r \in \mathbb{F}_{2^m}, \tag{6}$$

where $r \boxdot j$ denotes dot product of two elements of $\mathbb{F}_{2^m}$ represented as binary vectors. It was shown in [1] that the SC algorithm for binary polar codes can be implemented with complexity $O(n \log n)$. In the case of codes over $\mathbb{F}_{2^m}$ the complexity becomes $O(n \log n 2^m m)$.

### C. Stack decoding algorithms for binary polar codes

A major drawback of the SC algorithm is that it cannot correct errors which may occur at early phases of the decoding process. This problem is solved in stack/list algorithms by keeping a list of the most probable paths within code tree [3], [4], [2]. Path of length $i$ is identified by values $u_0^{i-1} \in \{0,1\}^i$. Each path is associated with its score, which depends on its probability. Stack algorithms [4], [2] keep the paths in a stack (priority queue). At each iteration the decoder selects for extension path $u_0^{i-1}$ with the largest score, and performs the $i$-th phase of SC decoding. That is, if $i \in \mathcal{F}$ the path is extended to obtain $u_0^i$, where $u_i$ is calculated according to (1), and the extended path is stored in the stack together with its score. Otherwise, the path is cloned to obtain new paths $(u_0, \ldots, u_{i-1}, 0)$ and $(u_0, \ldots, u_{i-1}, 1)$, which are stored in the stack together with their scores. In order to keep the size of the stack limited, paths with low scores can be purged from the stack. Furthermore, if the decoder returns to phase $i$ more than $L$ times, all paths shorter than $i+1$ are eliminated. Decoding terminates as soon a path of length $n$ appears at the top of the stack, or the stack becomes empty. Hence, the worst case complexity of stack decoding is given by $O(Ln \log n)$. Average decoding complexity depends on how path scores are defined.

In [5] a low-complexity version of such stack algorithm for binary polar codes was introduced. A path $u_0^i$ is associated with the following score

$$\hat{T}(u_0^i, y_0^{n-1}) = R_2(u_0^i, y_0^{n-1}) \hat{\Omega}(i), \tag{7}$$

where

$$R_2(u_0^i, y_0^{n-1}) = \max_{u_{i+1}^{n-1} \in \mathbb{F}_2^{n-i-1}} P(u_0^{n-1}|y_0^{n-1}), \tag{8}$$

$$\hat{\Omega}(i) = \prod_{j \in \mathcal{F}, j > i} (1 - P_j), \tag{9}$$

where $P_j$ is the $j$-th subchannel error probability, provided that exact values of all previous bits $u_{j'}$, $j' < j$, are available. Heuristic function $\hat{\Omega}(i)$ is equal to the expected value of probability that frozen symbols in the remaining part of input sequence $u_{i+1}^{n-1}$ are equal to $\mathbf{0}$. It depends only on $n$, $\mathcal{F}$ (i.e. the code being considered), channel properties and phase $i$. This approach enables one to compare paths $u_0^i$ with different lengths, and prevent the decoder from switching frequently between different paths. For any given channel probabilities $P_j$ can be pre-computed using density evolution.

Computation of probability $R_2(u_0^i, y_0^{n-1})$ for code of length $n$ reduces to computation of two probabilities for codes of length $n/2$, i.e.

$$R_2(u_0^{2i}, y_0^{n-1}) = \max_{u_{2i+1} \in \mathbb{F}_2} R_2\left(u_{0,e}^{2i+1} \oplus u_{0,o}^{2i+1}, y_0^{n/2-1}\right) \cdot R_2\left(u_{0,o}^{2i+1}, y_{n/2}^{n-1}\right), \tag{10}$$

$$R_2(u_0^{2i+1}, y_0^{n-1}) = R_2\left(u_{0,e}^{2i+1} \oplus u_{0,o}^{2i+1}, y_0^{n/2-1}\right) \cdot R_2\left(u_{0,o}^{2i+1}, y_{n/2}^{n-1}\right). \tag{11}$$

The initial value for these recursive expressions is given by $R_2(b, y_j) = P(b|y_j)$, $b \in \mathbb{F}_2$.

## III. Proposed approach

### A. Decoding algorithm

We propose to generalize decoding algorithm [5] to the case of polar codes with dynamic frozen symbols over $\mathbb{F}_{2^m}$, including extended primitive $(n = 2^m, k, n-k+1)$ Reed-Solomon codes. Transition from binary polar codes to polar codes over $\mathbb{F}_{2^m}$ does not affect the idea of the decoding algorithm. For $i \in \mathcal{F}$ the path $u_0^{i-1}$ is extended by symbol with value calculated according to (1), otherwise it is cloned to obtain $2^m$ paths $(u_0, \ldots, u_{i-1}, u_i), u_i \in \mathbb{F}_{2^m}$. So, scores corresponding to these $2^m$ extended paths should be computed. Let path scores $R_{2^m}(u_0^i, y_0^{n-1})$ be defined in the same way as in the binary case, except that maximization in (8) and (10) is performed over $\mathbb{F}_{2^m}$. Moreover, efficient techniques for computing $\hat{\Omega}(i)$ should be provided.

The complexity of the proposed method is at most $O(Ln \log n)$ unit calculations, where each unit is given by (10)

or (11). Observe that $R_{2^m}(u_0^i, y_0^{n-1})$ needs to be computed for $2^m$ distinct values of $u_i$. Since maximization in (10) also needs to be performed over $\mathbb{F}_{2^m}$, one obtains that the complexity of unit calculation is $O(n^2)$.

Calculations can be simplified by taking logarithms on both sides of (7)–(11), so that only addition and comparison operations are used. Furthermore, one can store only a few largest probabilities $R_{2^m}(u_0^i, y_0^{n-1})$ for intermediate symbols, as suggested for the case of LDPC codes over $\mathbb{F}_{2^m}$ (see [8] and references therein).

### B. Computing the heuristic function

*1) Gaussian approximation:* Let us consider the case of transmission of zero codewords over a memoryless output-symmetric channel[2]. To obtain $\hat{\Omega}(t)$ one needs to be able to compute $P_\gamma$. For the case of binary codes and AWGN channel both efficient implementation of density evolution [9] and Gaussian approximation method [10] are available.

For codes over $\mathbb{F}_{2^m}$ we propose to employ Gaussian approximation techniques developed in [11] in the context of LDPC codes. Let $l_r^{(n,\gamma)} = \log \frac{P(u_0^{\gamma-1}, 0 | y_0^{n-1})}{P(u_0^{\gamma-1}, r | y_0^{n-1})}$, be the log-likelihood ratios, which can be considered as jointly Gaussian random variables with $(2^m-1)\times(2^m-1)$ covariance matrix

$$\Sigma_{i,j} = \rho_i^{(n,\gamma)} + \rho_j^{(n,\gamma)} - \rho_{i\oplus j}^{(n,\gamma)}, \quad i,j \in \mathbb{F}_{2^m} \setminus \{0\}. \quad (12)$$

It was shown in [11] that their mean values $\rho_r^{(n,\gamma)}$ are given by

$$\rho^{(n,2\gamma)} = \Phi_{2^m-1}^{-1}(\Phi_{2^m-1}(\rho^{(n/2,\gamma)}) \odot \Phi_{2^m-1}(\rho^{(n/2,\gamma)})) \quad (13)$$

$$\rho^{(n,2\gamma+1)} = 2\rho^{(n/2,\gamma)}, \quad (14)$$

where $\Phi_{2^m-1}(\rho^{(n/2,\gamma)})$ is the vector of expected values of Hadamard transform components of the probability vector corresponding to $l_r^{(n/2,\gamma)}$ values, i.e. $(\Phi_{2^m-1}(\rho^{(n/2,\gamma)}))_r = E[f_r]$, where

$$f_r = \frac{\sum_{i\in\mathbb{F}_{2^m}} (-1)^{r\boxdot i} \exp(-l_i^{(n,\gamma)})}{\sum_{i\in\mathbb{F}_{2^m}} \exp(-l_i^{(n,\gamma)})}, r \in \mathbb{F}_{2^m} \setminus \{0\}.$$

One can compute $E[f_r] = E[\tanh(z_r/2)]$, where

$$z_r = \log \frac{\sum_{i\in\Psi^{(r)}} \exp(-l_i^{(n,\gamma)})}{\sum_{i\in\Psi^{(r)}} \exp(-l_{i\oplus h^{(r)}}^{(n,\gamma)})},$$

where $\Psi^{(r)}$ is a linear subspace of $\mathbb{F}_{2^m}$, and $h^{(r)} : r \boxdot h^{(r)} = 1$. Recursive expression for computing mean value of $z_r$ is provided in [11].

Eventually, one obtains

$$P_\gamma = 1 - \prod_{r=1}^{2^m-1} (1 - P_{\gamma,r}),$$

where $P_{\gamma,r} = Q\left(\sqrt{\widetilde{\rho}_r^{(n,\gamma)}/2}\right)$, and $\widetilde{\rho}^{(n,\gamma)} = \Sigma^{-1/2}\rho^{(n,\gamma)}$, $r \in \mathbb{F}_{2^m}$. Values $P_\gamma$ are employed in (9) to obtain $\hat{\Omega}(t)$.

[2]For $2^m$-QAM modulation averaging over signal constellation can be used to mimic the case of output-symmetric channel.

*2) A simplified method:* Evaluation of (13) even for $m = 4$ is computationally expensive and prone to numerical errors. Therefore, we propose a simpler alternative for it. We employ min-sum approximation $l_r^{(n,2\gamma)} \approx \lambda_r^{(n,2\gamma)} - \lambda_0^{(n,2\gamma)}$, where

$$\lambda_r^{(n,2\gamma)} = \min_{i\in\mathbb{F}_{2^m}} (l_i^{(n/2,\gamma)} + l_{i\oplus r}^{(n/2,\gamma)}), r \in \mathbb{F}_{2^m},$$

so that $\rho_r^{(n,2\gamma)} \approx \xi_r^{(n,2\gamma)} - \xi_0^{(n,2\gamma)}$, where $\xi_r^{(n,2\gamma)} = E[\lambda_r^{(n,2\gamma)}]$.

Consider some fixed non-zero $r$, and define random variable $\tilde{X}_i = -(l_i^{(n/2,\gamma)} + l_{i\oplus r}^{(n/2,\gamma)}), i \in \mathbb{F}_{2^m}$, so that $\mu_i = E[\tilde{X}_i] = -(\rho_i^{(n/2,\gamma)} + \rho_{i\oplus r}^{(n/2,\gamma)})$. Observe that these random variables are dependent and differently distributed (DDD). We still assume that $l_i^{(n,\gamma)}$ are Gaussian random variables, so $\tilde{X}_i$ are Gaussian too. A simple method for estimating $\xi_r^{(n,2\gamma)} = -E[\max_i \tilde{X}_i]$ in such case was suggested in [12]. The idea is to specify a new set of normal variables, whose expected maximum is easier to compute, and such that the new expected maximum is a bound on the original DDD variables' expected maximum. The method is based on the theorem from [13], which is stated in [12] as follows. Consider zero-mean DDD Gaussian random variables $\overline{W}_i$, $\overline{X}_i$ and $\overline{Y}_i$, $i \in \mathbb{F}_{2^m}$, where $\overline{X}_i = \tilde{X}_i - \mu_i$, such that, for all $i$, $j$

$$D[\overline{W}_i - \overline{W}_j] \le D[\overline{X}_i - \overline{X}_j] \le D[\overline{Y}_i - \overline{Y}_j]. \quad (15)$$

Here $D[A]$ is variance of random variable $A$. Then

$$E[\max_i \tilde{W}_i] \le E[\max_i \tilde{X}_i] \le E[\max_i \tilde{Y}_i], \quad (16)$$

where $\tilde{W}_i = \overline{W}_i + \mu_i$, and $\tilde{Y}_i = \overline{Y}_i + \mu_i$ for any $\mu_i$.

We cannot compute $E[\max_i \tilde{X}_i]$, however we can obtain its estimate via its lower bound $E[\max_i \tilde{W}_i]$ and its upper bound $E[\max_i \tilde{Y}_i]$. In [12] two approaches are considered: in the first one it is assumed that $\tilde{W}_i$ and $\tilde{Y}_i$ are perfectly DDD variables, and in the second case they are independent differently distributed variables (IDD). We employ the latter method, since it provides more accurate bounds. Lower bound $E[\max_i \tilde{W}_i]$ is given by

$$E[\max_i \tilde{W}_i] = \int_0^\infty 1 - \prod_{i\in\mathbb{F}_{2^m}} P\left\{\tilde{W}_i \le w\right\} - \prod_{i\in\mathbb{F}_{2^m}} P\left\{\tilde{W}_i < -w\right\} dw. \quad (17)$$

Since $\tilde{W}_i$ is Gaussian, one obtains $P\left\{\tilde{W}_i \le w\right\} = 1 - \frac{1}{2}\operatorname{erfc}\left(\frac{w-\mu_i}{\sqrt{2\sigma_{W_i}^2}}\right)$, where $\sigma_{W_i}^2$ is variance of $\tilde{W}_i$. The same expression can be used to compute $E[\max_i \tilde{Y}_i]$, except that $\sigma_{W_i}^2$ is replaced with $\sigma_{Y_i}^2$. One needs to identify variances $\sigma_{W_i}^2$ and $\sigma_{Y_i}^2$, so that bounds (16), which are computed via (17), are sufficiently tight.

Let us further define zero-mean random variables $\overline{L}_i = l_i^{(n/2,\gamma)} - \rho_i^{(n/2,\gamma)}, i \in \mathbb{F}_{2^m}$. One obtains from (12) that its variance is given by $\sigma_{L_i}^2 = 2\rho_i^{(n/2,\gamma)}$ and covariance

$\sigma_{L_{i,j}} = \rho_i^{(n/2,\gamma)} + \rho_j^{(n/2,\gamma)} - \rho_{i\oplus j}^{(n/2,\gamma)}$. We can compute

$$\begin{aligned} b_{i,j} =& D[\overline{X}_i - \overline{X}_j] = D[\overline{L}_i + \overline{L}_{i\oplus r} - \overline{L}_j - \overline{L}_{j\oplus r}] = \\ & 2(\sigma_{L_{i,i\oplus r}} + \sigma_{L_{j,j\oplus r}} - \sigma_{L_{i,j}} - \sigma_{L_{i\oplus r,j\oplus r}}) + \\ & \sigma^2_{L_i} + \sigma^2_{L_{i\oplus r}} + \sigma^2_{L_j} + \sigma^2_{L_{j\oplus r}} = 2\rho_i^{(n/2,\gamma)} + 2\rho_j^{(n/2,\gamma)} - \\ & 4\rho_r^{(n/2,\gamma)} + 4\rho_{i\oplus j}^{(n/2,\gamma)} + 2\rho_{i\oplus r}^{(n/2,\gamma)} + 2\rho_{j\oplus r}^{(n/2,\gamma)}. \end{aligned}$$

Condition (15) can be expressed via $b_{i,j}$ as $\sigma^2_{W_i} + \sigma^2_{W_j} \le b_{i,j}$ and $\sigma^2_{Y_i} + \sigma^2_{Y_j} \ge b_{i,j}$, since in the case of IDD variables $\tilde{W}_i$ and $\tilde{Y}_i$ their covariances are equal to zero.

The optimal lower bound on $E[\max_i \tilde{X}_i]$ for DDD variables $\tilde{X}_i$ via IDD variables $\tilde{W}_i$ is given by

$$B^{(Low)} = \max_{\sigma^2_{W_i}+\sigma^2_{W_j} \le b_{i,j}} E[\max_i \tilde{W}_i].$$

Similar upper bound on $E[\max_i \tilde{X}_i]$ is given by

$$B^{(Upper)} = \min_{\sigma^2_{Y_i}+\sigma^2_{Y_j} \ge b_{i,j}} E[\min_i \tilde{Y}_i].$$

The complexity of finding optimal values of $\sigma_{W_i}$ and $\sigma_{Y_i}$ is rather high. Therefore, we propose to construct a feasible solution given by to $\sigma^2_{W_i} = \frac{1}{2}\min_j b_{ij}$, $\sigma^2_{Y_i} = \frac{1}{2}\max_j b_{ij}$.

Such suboptimal solution for $\sigma_{W_i}$ provides rather good accuracy for the lower bound. However, the value of an upper bound appears to be too high in the case of substantially different $b_{i,j}$. It is easy to see that values $b_{i,j}$ depend on the means $\mu_i$ and $\mu_j$, more precisely $b_{i,j} = 2(\rho_i^{(n/2,\gamma)} + \rho_{i\oplus r}^{(n/2,\gamma)}) + 2(\rho_j^{(n/2,\gamma)} + \rho_{j\oplus r}^{(n/2,\gamma)}) - 4\rho_r^{(n/2,\gamma)} + 4\rho_{i\oplus j}^{(n/2,\gamma)} = -2\mu_i - 2\mu_j - 4\rho_r^{(n/2,\gamma)} + 4\rho_{i\oplus j}^{(n/2,\gamma)}$. Recall that $\rho_i^{(n/2,\gamma)} \ge 0$, $\mu_i \le 0$ and $b_{i,j} \ge 0$. Thus, by excluding variables $\tilde{X}_i$ with lowest means $\mu_i$ from consideration one can reduce difference between maximum and minimum among $b_{i,j}$. On the other hand, in the case of $\tilde{X}_i$ with substantially different mean values $\mu_i$, variables $\tilde{X}_i$ with lowest mean $\mu_i$ have negligible impact on the value of $E[\max_i \tilde{X}_i]$. Therefore, we propose to choose an appropriate coefficient $\alpha$ and perform maximization in (16) over subset $i \in \Gamma = \{j | \mu_j \ge \alpha\mu_{max}\}$, where $\mu_{max} = \max_i \mu_i$. That is, we employ approximation

$$E[\max_{i\in\mathbb{F}_{2^m}} \tilde{X}_i] \approx E[\max_{i\in\Gamma} \tilde{X}_i] \le E[\max_{i\in\Gamma} \tilde{Y}_i].$$

For example, in the simulations we used $\alpha = 3$.

There are different way to combine lower and upper bound for $\xi_r^{(n,2\gamma)} = -E[\max_i \tilde{X}_i]$ to obtain its estimate. The simplest solution, i.e. average value, appears to provide rather accurate results. That is, we propose to replace (13) with $\rho_r^{(n,2\gamma)} \approx \xi_r^{(n,\gamma)} - \xi_0^{(n,\gamma)}$, where

$$\xi_r^{(n,2\gamma)} = -\frac{1}{2}(E[\max_{i\in\mathbb{F}_{2^m}} \tilde{W}_i] + E[\max_{i\in\Gamma} \tilde{Y}_i])$$

## IV. Decoding of the binary image of the code

In the case of transmission of a binary image of RS code over a memoryless channel it can be seen that

$$P(u_0^i | y_0^{n-1}) = \prod_{j=0}^{m-1} P\left(u_0^i[j] | y_0^{n-1}[j]\right), \qquad (18)$$

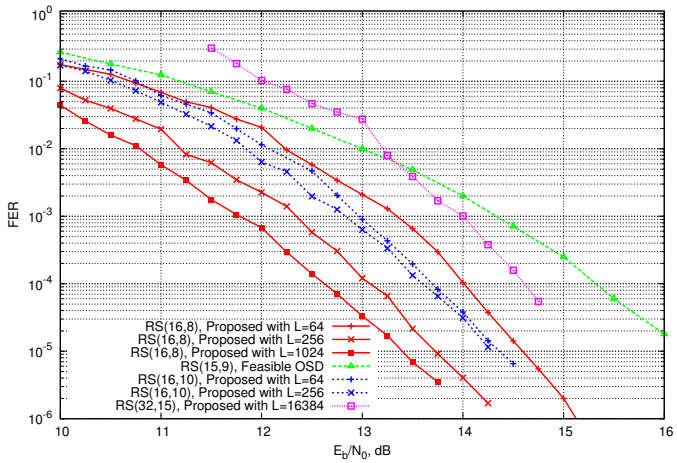


Fig. 1. Performance of Reed-Solomon codes over $\mathbb{F}_{2^m}$ in AWGN channel with $2^m$-QAM

and

$$R_{2^m}(u_0^i, y_0^{n-1}) = \prod_{j=0}^{m-1} R_2\left(u_0^i[j], y_0^{n-1}[j]\right), \qquad (19)$$

where $u_0^i[j] = (u_0[j], \dots, u_i[j])$, $u_s = \sum_{j=0}^{m-1} u_s[j]a_j$, $u_s[j] \in \mathbb{F}_2$, $(a_0, \dots, a_{m-1})$ is some basis of $\mathbb{F}_{2^m}$, and $y_0^{n-1}[j]$ is the subvector of $y_0^{n-1}$ corresponding to the $j$-th bits of transmitted symbols. Hence, decoding of the code over $\mathbb{F}_{2^m}$ can be implemented using $m$ instances of the decoder for binary codes, and (19) needs to be used only on the leftmost layer of the polarizing transformation, which corresponds to the information symbols. Evaluation of dynamic frozen symbols still needs to be performed jointly using arithmetics of $\mathbb{F}_{2^m}$. This results in significant complexity reduction.

Expression (18) implies that symbol error probability can be computed as $P_i = 1 - (1 - p_i)^m$, where $p_i$ is bit error probability calculated for the $i$-th bit subchannel of binary polar code. Hence, $\hat{\Omega}(i)$ can be computed using the techniques developed for binary polar codes [9], [10].

## V. Numeric results

Figure 1 presents simulation results illustrating the performance of the proposed decoding method for the case of Reed-Solomon codes over $\mathbb{F}_{2^m}$ and $2^m$-QAM modulation. At most $L$ paths $u_0^{i-1}$ for any length $0 \le i < 2^m$ within code tree were considered by the decoder. This corresponds to SC list decoding [3] with list size $L$. Observe that increasing $m$ requires exponential increase of $L$ in order to obtain reasonable performance. It can be seen that the proposed approach provides 2 dB gain compared to the feasible order statistic decoding method introduced in [14].

Figure 2 presents the comparison of error probabilities for the subchannels of $32 \times 32$ polarizing transformation estimated via the method proposed in Section III-B2, and those obtained by simulations. It can be seen that they agree quite well, except for a few initial channels.

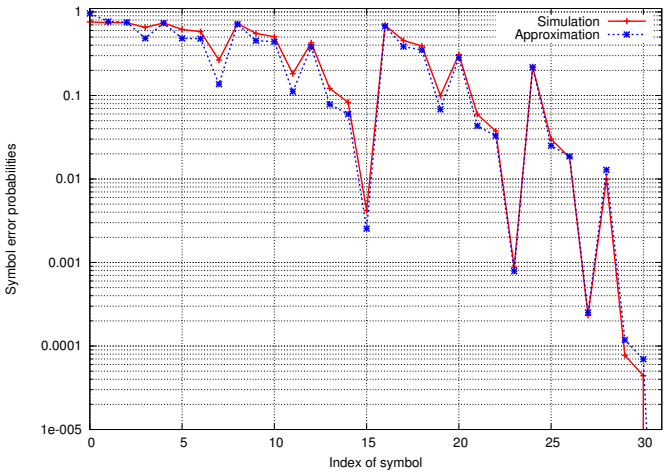


Fig. 2. Symbol error probabilities for AWGN channel with $2^5$-QAM, $N_0 = 0.08$

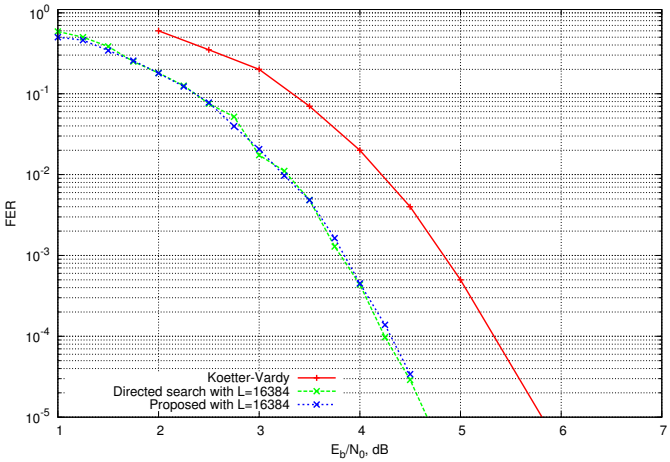


Fig. 3. Performance of $(31, 15)$ Reed-Solomon code in the case binary image transmission over AWGN channel

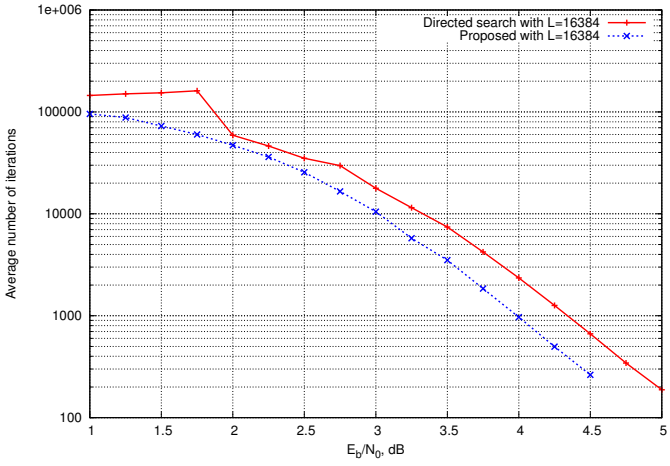


Fig. 4. Average number of iterations for $(31, 15)$ Reed-Solomon code in the case binary image transmission over AWGN channel

Figure 3 illustrates performance of the proposed decoding method, as well as Koetter-Vardy algebraic soft-decision decoding algorithm [15], and the directed search successive cancellation stack decoding method suggested in [6], for the case of transmission of a binary image of $(31, 15, 17)$ RS code over the AWGN channel. It can be seen that the proposed method provides exactly the same performance as the directed search method, and both of them outperform Koetter-Vardy algorithm by 1 dB. Figure 4 shows the average number of iterations performed by the proposed algorithm and directed search one. It can be seen that at $E_b/N_0 = 4$ dB the proposed method requires in average 2.4 times less iterations. It was found that the average decoding time for the case of proposed method is also 2.4 times less than the one for directed search.

## VI. Conclusions

In this paper a novel method for decoding Reed-Solomon codes over $\mathbb{F}_{2^m}$ was introduced, which is based on their representation as Arikan polar codes with dynamic frozen symbols, and application of the sequential decoding algorithm presented in [5]. This method was shown to provide better performance compared to Koetter-Vardy and feasible order statistic methods. Furthermore, the complexity of the proposed method is less than that of directed search successive cancellation decoding algorithm given in [6].

## 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, July 2009.

[2] P. Trifonov and V. Miloslavskaya, "Polar codes with dynamic frozen symbols and their decoding by directed search," in *Proceedings of IEEE International Workshop on Information Theory*, September 2013.

[3] I. Tal and A. Vardy, "List decoding of polar codes," in *Proceedings of IEEE International Symposium on Information Theory*, 2011.

[4] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, October 2012.

[5] V. Miloslavskaya and P. Trifonov, "Sequential decoding of polar codes," *IEEE Communications Letters*, 2014, submitted for publication.

[6] P. Trifonov, "Successive cancellation decoding of Reed-Solomon codes," *Problems of information transmission*, 2014, submitted.

[7] D. Declercq and M. P. Fossorier, "Decoding algorithms for nonbinary LDPC codes over $GF(q)$," *IEEE Transactions On Communications*, vol. 55, no. 4, April 2007.

[8] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity, low-memory EMS algorithm for non-binary LDPC codes," in *Proceedings of IEEE International Conference on Communications*, June 2007, pp. 671 – 676.

[9] I. Tal and A. Vardy, "How to construct polar codes," *IEEE Transactions On Information Theory*, vol. 59, no. 10, October 2013.

[10] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE Transactions on Communications*, vol. 60, no. 11, November 2012.

[11] G. Li, I. Fair, and W. A. Krzymien, "Density evolution for nonbinary LDPC codes under gaussian approximation," *IEEE Transactions On Information Theory*, vol. 55, no. 3, March 2009.

[12] A. Ross, "Computing bounds on the expected maximum of correlated normal variables," *Methodology and Computing in Applied Probability*, vol. 12, no. 1, pp. 111–138, 2010.

[13] R. Vitale, "Some comparisons for gaussian processes," *Proceedings of the American Mathematical Society*, vol.128, no.10, p.3043-3046, 2000.

[14] Y.-W. Ching and T.-H. Hu, "A feasible ordered statistic decoding for Reed-Solomon codes with QAM signaling," *Journal Of Chung Cheng Institute Of Technology*, vol. 42, no. 2, 2013.

[15] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, November 2003.