ISSN 0032-9460, Problems of Information Transmission, 2014, Vol. 50, No. 3, pp. 303–312. © Pleiades Publishing, Inc., 2014.
Original Russian Text © P.V. Trifonov, 2014, published in Problemy Peredachi Informatsii, 2014, Vol. 50, No. 3, pp. 3–14.

CODING THEORY

Successive Cancellation Decoding of Reed–Solomon Codes[1]

P. V. Trifonov

St. Petersburg State Polytechnical University
petert@dcn.icc.spbstu.ru

Received December 30, 2013; in final form, August 12, 2014

Abstract—A novel soft-decision decoding algorithm for Reed-Solomon codes over $GF(2^m)$ is proposed, which is based on their representation as polar codes with dynamic frozen symbols and the successive cancellation method. Further performance improvement is obtained by exploiting multiple permutations of codewords, which are taken from the automorphism group of Reed-Muller codes. It is also shown that the proposed algorithm can be simplified in the case of decoding of a binary image of the Reed-Solomon code.

DOI: 10.1134/S0032946014040012

1. INTRODUCTION

Despite of several years of research, the problem of soft-decision decoding of Reed-Solomon codes has still no satisfactory solution. The decoding error probability of the existing methods [1–4] significantly exceeds that of maximum likelihood decoding. Reducing it requires one to increase considerably the decoding complexity.

Polar codes introduced in [5] asymptotically achieve the capacity of a wide class of communication channels, but at moderate length provide inferior performance compared to other known classes of codes. The successive cancellation algorithm, which was suggested for decoding of polar codes, fails to provide maximum likelihood performance. However, its list extension introduced in [6], as well as the analogues of sequential decoding presented in [7,8], enable one to perform near maximum likelihood decoding with complexity $O(Ln\log n)$, where n is code length, and L is the maximum number of branches at each level of the code tree considered by the decoder.

In this paper application of the successive cancellation method and its analogues to the problem of decoding of Reed-Solomon codes over $GF(2^m)$. The proposed approach is based on their representation as polar codes with dynamic frozen symbols [8]. Furthermore, multiple decoding attempts are performed for various permutations of the received sequence, which are drawn from the automorphism group of Reed-Muller codes.

The paper is organized as follows. § 2 provides a survey of polar codes and their decoding techniques. A new decoding method for Reed-Solomon codes is presented in § 3. Numeric results illustrating the performance of the proposed method are provided in § 4.

2. POLAR CODES AND SUCCESSIVE CANCELLATION DECODING

A $(n=2^m,k)$ polar code over $GF(2)$ is generated by k rows of matrix $A=B_mF^{\otimes m}$, where $F=\begin{pmatrix}1 & 0\\ 1 & 1\end{pmatrix}$, $\otimes m$ denotes m-times Kronecker product of the matrix with itself, B_m is the bit-reversal permutation matrix. It is possible to show that the linear transformation given by matrix

[1] Supported in part by the Russian Foundation for Basic Research, project no. 12-01-00365.

A together with 2^m instances of a binary input memoryless output-symmetric channel gives rise to 2^m symbol subchannels, and the capacities of these subchannels in the binary case converge with m to 0 or 1. The classical construction of polar codes assumes that the rows of A to be included into the generator matrix are those corresponding to subchannels with capacity close to 1 [5]. Hence, the encoding operation for polar codes is given by $c_0^{n-1} = u_0^{n-1}A$, where vector u_0^{n-1} has 0 in positions given by set $\mathcal{F}$ of subchannels with capacities close to 0. Here a_i^j denotes $(a_i, a_{i+1}, \dots, a_j)$. In what follows, A will be referred to as the polarizing transformation matrix, and u_0^{n-1} will be denotes as an information vector of the polarizing transformation.

It will be convenient here to consider polar codes as subcodes of Reed-Muller codes. One can see that the codewords of polar codes can be represented as vectors of values of Zhegalkin polynomials at various points of $GF(2)^m$, where the coefficients at monomials $x_0^{t_0}x_1^{t_1}\dots x_{m-1}^{t_{m-1}}$: $\sum\limits_{i=0}^{m-1} 2^i(1-t_i) \in \mathcal{F}$, $t_i \in \{0,1\}$, are set to zero.

Let y_0^{n-1} be the result of transmission of codeword $c_0^{n-1} = u_0^{n-1}A$ over a memoryless symmetric channel. The classical decoding method for polar codes is the successive cancellation algorithm. It consists in making successive decisions according to

$$\widehat{u}_i = \begin{cases} 0, & i \in \mathcal{F}, \\ \arg\max\limits_{a\in\{0,1\}} P_{U_0^i|Y_0^{n-1}}\{\widehat{u}_0^{i-1}, a \,|\, y_0^{n-1}\}, & i \notin \mathcal{F}, \end{cases} \tag{1}$$

where U_i, Y_i are random variables corresponding to the i-th elements of vectors u and y, respectively. Probabilities $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\} = W_n^{(i)}(u_0^i \,|\, y_0^{n-1})$ can be recursively computed as

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

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

where $W_1^{(0)}(u_0 \,|\, y_0) = P_{U_0|Y_0}\{u_0 \,|\, y_0\}$ is the probability of transmission of symbol $U_0 = u_0$ over the channel, provided that the output of the channel is $Y_0 = y_0$, while $u_{0,e}^j$ and $u_{0,o}^j$ denote subvectors of u_0^j consisting of elements with even and odd indices, respectively. If intermediate values in these expressions are re-used, the complexity of the successive cancellation decoding algorithm is given by $O(n \log n)$.

One can assume that symbols U_i are transmitted over memoryless symmetric channels with binary input, which are given by distributions $P_{Y_0^{n-1}, U_0^{i-1}|U_i}\{y_0^{n-1}, u_0^{i-1} \,|\, u_i\}$. It was shown in [5] that the Bhattacharyya parameters $Z_{n,i}$ of these channels satisfy

$$Z_{n,2i} \le 2Z_{n/2,i} - Z_{n/2,i}^2, \tag{4}$$

$$Z_{n,2i+1} = Z_{n/2,i}^2, \tag{5}$$

where $Z_{1,0}$ is the Bhattacharyya parameter of the underlying communication channel.

The main drawback of the successive cancellation decoder is that it cannot recover erroneous decisions $\widehat{u}_i$ which may occur at early stages of the algorithm. Hence, this method fails to provide maximum likelihood decoding. Therefore, its generalizations are widely used, such as the list successive cancellation decoder [6], as well as analogues of sequential decoding [7,8]. These methods assume that one simultaneously considers a number of paths within code tree, which are given by various values u_0^{i-1}, so that the number of paths of length i for any $i \in \{0, \dots, n-1\}$ is upper-bounded by some integer L; paths with sufficiently small values of $P_{U_0^i|Y_0^{n-1}}\{\widehat{u}_0^i \,|\, y_0^{n-1}\}$ are eliminated. Application of these techniques enables one to perform near maximum likelihood decoding (for sufficiently large L) of polar codes with complexity $O(Ln \log n)$.

3. DECODING OF REED-SOLOMON CODES

3.1. Dynamic frozen symbols

It can be seen that matrix A is invertible, and any vector of length 2^m can be obtained as $c_0^{n-1} = u_0^{n-1}A$. Let us consider the constraints which need to be imposed on vector u so that the corresponding vector c belongs to some $(n = 2^m, k, d)$ code over $GF(q)$ with check matrix H. Obviously, these constraints are given by [8]

$$uAH^T = 0. \tag{6}$$

By applying elementary linear operations given by invertible matrix Q to rows of matrix HA^T, one obtains matrix $V = QHA^T$, such that at most one row ends[2] in each column. Let j_i be the index of the column where row i ends. Without loss of generality, let us assume that $V_{i,j_i} = -1$. Then $uV^T = 0$ and

$$u_{j_i} = \sum_{s<j_i} V_{i,s}u_s, \quad 0 \le i < n-k. \tag{7}$$

These constraints can be considered as a generalization of the concept of symbol freezing, which is used in the construction of polar codes. Symbols u_{j_i}, such that the right-hand side of (7) is not identically zero, will be referred to as dynamic frozen ones. The remaining symbols u_{j_i} will be called statically frozen. Matrix V will be referred to as constraint matrix.

Observe that encoding vector $x \in GF(q)^k$ with the considered code can be performed as $c = xWB_mF^{\otimes m}$, where matrix W is a solution of equation $WV^T = 0$.

Obviously, this approach is applicable to all linear block codes of length 2^m, and enables one to employ the successive cancellation method for their decoding. It must be, however, recognized that for arbitrary linear code the set of frozen symbols does not need to coincide with the set of low-capacity subchannels. Therefore, the error probability under successive cancellation decoding may be much higher compared to the maximum likelihood method. Furthermore, applying this approach to equivalent codes may result in different sets of frozen symbols, which results in different error probabilities under the successive cancellation algorithm and its analogues.

Let us consider application of this approach to extended primitive Reed-Solomon codes over $GF(2^m)$. Any codeword of $(n = 2^m, k, 2^m - k + 1)$ Reed-Solomon code can be represented as $c = (f(x_0), f(x_1), \ldots, f(x_{n-1}))$, where $f(x) = \sum\limits_{i=0}^{k-1} f_i x^i$, $f_i \in GF(2^m)$. On the other hand, for any extended Reed-Solomon code one can find a check matrix, and use it to represent the code as a polar code with dynamic frozen symbols.

Let $x = \sum\limits_{i=0}^{m-1} x_i a_i$, $x_i \in GF(2)$ be an expansion of $x \in GF(2^m)$ in some basis $a_0, \ldots, a_{m-1}$ of $GF(2^m)$, and $j = \sum\limits_{s=0}^{m-1} j_s 2^s$ be a binary expansion of integer j. Then $x^j = \Big(\sum\limits_{i=0}^{m-1} x_i a_i\Big)^{\sum\limits_{s=0}^{m-1} j_s 2^s} = \prod\limits_{s=0}^{m-1}\Big(\sum\limits_{i=0}^{m-1} x_i a_i^{2^s}\Big)^{j_s}$. Expanding this expression, one obtains a polynomial in variables $x_0, \ldots, x_{m-1}$ of degree $\mathrm{wt}(j) = \sum\limits_{s=0}^{m-1} j_s$, which can be represented as $x^j = \sum\limits_{t\in Q_j} w_{jt} \prod\limits_{i=0}^{m-1} x_i^{t_i}$, where Q_j is the set of multi-degrees $t = (t_0, \ldots, t_{m-1})$ of non-zero terms, and $w_{jt} \in GF(2^m)$ are some coefficients. Therefore,

$$f(x) = \sum_{j=0}^{k-1} f_j \sum_{t\in Q_j} w_{jt} \prod_{i=0}^{m-1} x_i^{t_i}. \tag{8}$$

[2] Row i ends in column j iff $V_{i,j} \ne 0$ and $V_{i,j'} = 0$ for all $j' > j$.

This expression is a generalized Zhegalkin polynomial $f\colon GF(2)^m \to GF(2^m)$. Hence, any codeword of the Reed-Solomon code can be represented as

$$c = fWB_mF^{\otimes m}, \tag{9}$$

where $W = (w_{jt})$ and $WB_mF^{\otimes m}$ is a generator matrix of the Reed-Solomon code. Observe that row $t = \sum\limits_{i=0}^{m-1} t_{m-1-i}2^i$ of matrix $B_mF^{\otimes m}$ is a sequence of values of mononomial $\prod\limits_{i=0}^{m-1} x_i^{1-t_i}$ in various points $(x_0, \ldots, x_{m-1})$. The corresponding matrix V satisfies $WV^T = 0$.

Combining the terms of $f(x)$ with indices j corresponding to the same cyclotomic coset $C_s = \{s2^i \mid 0 \le i < m_s,\ s2^{m_s} \equiv s \bmod 2^m - 1\}$, one obtains

$$f(x) = \sum_{s\in S}\sum_{i=0}^{m_s-1} f_{s2^i \bmod (2^m-1)}x^{s2^i} = \sum_{s\in S} L_s(x^s) = \sum_{s\in S}\sum_{t\in Q_s} L_s(w_{st}) \prod_{i=0}^{m-1} x_i^{t_i}, \tag{10}$$

where $L_s(x) = \sum\limits_{i=0}^{m_s-1} f_{s2^i \bmod (2^m-1)}x^{2^i}$, and S is the set of cyclotomic coset representatives.

Theorem. *$(2^m, k, 2^m-k+1)$ extended Reed-Solomon code over $GF(2^m)$ can be represented as a polar code with the set of dynamic frozen symbols $\mathcal{F} = \{0, \ldots, 2^m-1\} \setminus \{2^m-1-r_m(i) \mid 0 \le i < k\}$, where $r_m(i) = \sum\limits_{j=0}^{m-1} i_j 2^{m-j-1}$ is an integer obtained by reversing the bits of integer $i = \sum\limits_{j=0}^{m-1} i_j 2^j$, $i_j \in \{0, 1\}$.*

Proof. Let us show the statement by induction over the number of check symbols $r = 2^m - k$. For $r = 1$ the statement is obvious. Let us assume that it holds for the code with $r-1$ check symbols. From (10) one can see that reducing code dimension by 1, i.e. introducing an additional check symbol, causes the rank of W to drop by 1. The columns of weight $m - \mathrm{wt}(k) = \mathrm{wt}(2^m-1-k)$, $k = 2^m - r$ become linearly dependent. The last such column, which is not included into the set of frozen symbols for a code of higher dimension, has number $2^m - 1 - r_m(k)$. It corresponds to a symbol subchannel which becomes frozen while going to a code with r check symbols. $\triangle$

It can be seen from the above proof that all symbol subchannels of the polarizing transformation with numbers i: $\mathrm{wt}(i) < m - r$, where $r = \max\limits_{0\le j<k} \mathrm{wt}(j)$, are statically frozen. Similar approach was used in [9, 10] to show that extended BCH codes are subcodes of Reed-Muller codes.

Example 1. Consider $(8, 4, 4)$ Reed-Solomon code over $GF(2^3)$. Its check matrix is given by

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & \alpha & 1+\alpha & \alpha^2 & 1+\alpha^2 & \alpha+\alpha^2 & 1+\alpha+\alpha^2 \\ 0 & 1 & \alpha^2 & 1+\alpha^2 & \alpha+\alpha^2 & 1+\alpha+\alpha^2 & \alpha & 1+\alpha \\ 0 & 1 & 1+\alpha & \alpha^2 & 1+\alpha^2 & \alpha+\alpha^2 & 1+\alpha+\alpha^2 & \alpha \end{pmatrix},$$

where α is a primitive root of $x^3 + x + 1$. By applying elementary row operations to matrix $H(B_3F^{\otimes 3})^T$, one obtains

$$V = \begin{pmatrix} 0 & \alpha & 0 & 1+\alpha & 0 & \alpha+\alpha^2 & 1 & 0 \\ 0 & 1+\alpha & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \alpha+\alpha^2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Hence, u_0 is a statically frozen symbol, while u_2, u_4, u_6 are dynamic frozen symbols.

3.2. Successive cancellation decoding

Computing probabilities of paths within the code tree. The construction of polar codes and the successive cancellation decoding algorithm can be extended to the case of arbitary fields. Application of the successive cancellation decoding method to the case of codes over $GF(2^\mu)$, $\mu > 1$, requires one to be able to efficiently compute $W_n^{(i)}(u_0^i \,|\, y_0^{n-1}) = P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}$, where $u_i \in GF(2^\mu)$, $u_i \in GF(2^\mu)$, are information symbols of the polarizing transformation given by matrix $A \in GF(2)^{n\times n}$. Here (2) needs to be changed to

$$W_n^{(2i)}(u_0^{2i} \,|\, y_0^{n-1}) = \sum_{u_{2i+1}\in GF(2^\mu)} W_{\frac{n}{2}}^{(i)}(u_{0,e}^{2i+1} + u_{0,o}^{2i+1} \,|\, y_0^{\frac{n}{2}-1}) W_{\frac{n}{2}}^{(i)}(u_{0,o}^{2i+1} \,|\, y_{\frac{n}{2}}^{n-1}). \tag{11}$$

Straightforward evaluation of this expression requires $2^{2\mu}$ operations. Observe that (11) represents a multi-dimensional cyclic convolution. Therefore, one can construct more efficient implementation of the procedure for computing $W_n^{(2i)}(u_0^{2i} \,|\, y_0^{n-1})$ with complexity $O(\mu 2^\mu)$ by employing a fast Hadamard transform. Furthermore, one can employ complexity reduction techniques developed in the area of LDPC codes [11].

In the case of transmission of a binary image of the code over a binary input memoryless output symmetric channel one obtains that

$$P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\} = \prod_{j=0}^{\mu-1} P_{U_{0,j}^i|Y_0^{n-1}}\{u_{0,j}^i \,|\, y_0^{n-1}\}, \tag{12}$$

where $u_{0,j}^i = (u_{0,j}, \ldots, u_{i,j})$, $u_s = \sum\limits_{j=0}^{\mu-1} u_{s,j}a_j$, $u_{s,j} \in GF(2)$, and $(a_0, \ldots, a_{\mu-1})$ is a basis of $GF(2^\mu)$. Probabilities $P_{U_{0,j}^i|Y_0^{n-1}}\{u_{0,j}^i \,|\, y_0^{n-1}\}$ can be computed using expressions (2), (3), and transformed into $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}$ $u_i \in GF(2^\mu)$ with complexity $O(2^\mu)$ operations.

Using the representation of a linear code of length $n = 2^m$ over $GF(2^\mu)$ as a polar code with dynamic frozen symbols, one can perform its successive cancellation decoding according to the rule

$$\widehat{u}_i = \begin{cases} \sum\limits_{s<j_l} V_{l,s}\widehat{u}_s, & i \in \mathcal{F}, \\ \arg\max\limits_{a\in GF(2^\mu)} P_{Y_0^{n-1},U_0^{i-1}|U_i}\{y_0^{n-1}, \widehat{u}_0^{i-1} \,|\, a\}, & i \notin \mathcal{F}, \end{cases} \tag{13}$$

where $l : j_l = i$, and $\mathcal{F} = \{j_0, \ldots, j_{n-k-1}\}$ is the set of dynamic frozen symbols.

The decoding error probability under this approach can be computed as

$$P = 1 - \prod_{j\notin\mathcal{F}} (1 - P_j),$$

where P_j is the error probability in the j-th symbol subchannel of the polarizing transformation provided that values $u_0, \ldots, u_{j-1}$ are known to the decoder. In general, for $\mu > 1$ the problem of computing values P_j has not yet been solved. However, for the special case of transmission of a binary image of the code over a binary input memoryless output-symmetric channel one can obtain P_j from the probabilities p_j computed for the case of $\mu = 1$. Indeed, using the representation $u_i = \sum\limits_{j=0}^{\mu-1} u_{i,j}a_j$, where $a_0, \ldots, a_{\mu-1}$ is some basis of $GF(2^\mu)$, one obtains that encoding, transmission and decoding of blocks $(u_{0,j}, \ldots, u_{n-1,j})$, $0 \le j < \mu$, is performed independently. Hence,

$$1 - P_i = (1 - p_i)^\mu.$$

Values p_i can be obtained using density evolution [12] or Gaussian approximation [13, 14].

Furthermore, it was shown in [15] that the capacities of subchannels $P_{Y_0^{n-1},U_0^{i-1}|U_i}(y_0^{n-1}, u_0^{i-1} \,|\, u_i)$ of the polarizing transformation $B_m F^{\otimes m}$ with input alphabet $GF(2^\mu)$ converge with m to values $0, 1, \ldots, \mu$. For subchannels with capacities close to $i \in \{1, \ldots, \mu - 1\}$, error probability P_j is close to $1 - 2^{i-\mu}$. Hence, in some cases many values P_j, $j \notin \mathcal{F}$, may be quite high, so that the probability of error under the successive cancellation decoding may be unacceptably high. However, for the case of practically important case of transmission of a binary image of the code over a memoryless symmetric channel $p_i \le Z_{n,i}$, where $Z_{n,i}$ are given by (4). It can be seen that $Z_{n,i} = O(Z_{1,0}^{2^{\mathrm{wt}(i)}})$, where $\mathrm{wt}(i)$ is the number of non-zero bits in the binary representation of integer i, and $Z_{1,0}$ is the Bhattacharyya parameter of the communication channel being used. Hence, the successive cancellation decoding error probability of a linear code over $GF(2^\mu)$ can be estimated as $P \le O(n\mu Z_{1,0}^{2^{w_0}})$, where $w_0 = \max\limits_{i \notin \mathcal{F}} \mathrm{wt}(i)$.

The successive cancellation decoding algorithm can be also used for decoding of codes of length other than a power of 2. To do this, one should consider the code as the one obtained by puncturing code of some suitable code of length 2^m, and assume during decoding that the missing codeword symbols are erased. This must be taken into account while computing error probabilities P_i in symbol subchannels.

The above described approach can be used for decoding of any linear code. However, in general set $\mathcal{F}$ appears to be such that P_j, $j \notin \mathcal{F}$, are quite high (i.e. w_0 is close to zero), rendering the described approach almost useless. In the case of Reed-Solomon codes the above theorem implies that $w_0 = \max\limits_{i<k}(m - \mathrm{wt}(i))$, i.e. the decoding error probability decreases sufficiently fast while improving the quality of the original communication channel (at least in the case of transmission of a binary image of the code over a memoryless channel).

Sequential decoding Calculations show that in the case of Reed-Solomon codes the error probability under successive cancellation decoding appears to be higher compared to other known approaches. In order to improve the performance one may employ the list successive cancellation algorithm introduced in [6]. However, in the case of Reed-Solomon codes huge list size is needed in order to obtain performance comparable with other decoding algorithms. It was shown in [7] that in the case of polar codes the average decoding complexity can be substantially reduced if one sequentially constructs paths within code tree corresponding to different vectors u_0^i, so that at each step one selects for extension the path with the highest value of $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}$, which is referred to as path metric. Different paths are stored in a stack (priority queue), which provides efficient implementation of new element insertion and extraction of the element with the highest metric.

It was shown in [8] that further complexity reduction can be achieved if one selects at each step for expansion path u_0^i with the highest value of an estimate of

$$P_{U_0^{n-1}|Y_0^{n-1}}\{u_0^{n-1} \,|\, y_0^{n-1}\} = P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\} \prod_{j=i+1}^{n-1} P_{U_j|U_0^{j-1},Y_0^{n-1}}\{u_j \,|\, u_0^{j-1}, y_0^{n-1}\}.$$

Since values u_j^{n-1} are unknown, the second multiple in this expression can be estimated by averaging over all Y_0^{n-1}. Assuming that u_0^{n-1} is the most probable path in code tree, one obtains that $P_{U_j|U_0^{j-1},Y_0^{n-1}}\{u_j \,|\, u_0^{j-1}, y_0^{n-1}\}$ is the probability of transmission of the most probable value of symbol U_j, i.e. the expected value of the second term is

$$\psi(i) = \prod_{j=i+1}^{n-1} (1 - P_j).$$

The values of $\psi(i)$ can be pre-computed.

In the case of $(n = 2^m, k)$ Reed-Solomon code over $GF(2^m)$ the proposed decoding algorithm involves the following steps:

1. Put into the stack a zero-length path and set $q_i = 0$, $0 \le i < n$;
2. Extract from the stack path u_0^{i-1} with the highest metric. If $i = n$, return the corresponding codeword $c = u_0^{n-1}A$;
3. Compute via (11) and (3) (or (12) in the case of transmission of a binary image of the code) $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}$, $u_i \in GF(2^m)$;
4. If $i \in \mathcal{F}$, compute $u_i = \sum\limits_{s<j_l} V_{l,s}u_s, j_l = i$, and push into the stack path u_0^i with metric $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}\psi(i)$;
5. If $i \notin \mathcal{F}$, push into the stack 2^m paths u_0^i, $u_i \in GF(2^m)$, with metrics $P_{U_0^i|Y_0^{n-1}}\{u_0^i \,|\, y_0^{n-1}\}\psi(i)$;
6. Let $q_i = q_i + 1$. If $q_i \ge L$, remove from the stack all paths with length less than i;
7. Go to step 2.

Step 6 ensures that at most L paths passing through each layer of the code tree are considered. Since computing (11) requires $O(m2^m)$ operations, one obtains that the worst case complexity of the algorithm is $O(Ln^2 \log^2 n)$. In the case of transmission of a binary image of the code computing $P_{U_{0,j}^i|Y_0^{n-1}}\{u_{0,j}^i \,|\, y_0^{n-1}\}$, $0 \le j < m$, for each path of the tree requires at most $O(mn \log n)$ operations, and transforming these values according to (12) requires at most $O(n^2)$ operations. Hence, the worst case complexity of the proposed algorithm is given by $O(L(n \log^2 n + n^2))$. The same complexity estimates are valid for the case of list decoding algorithm presented in [6].

Observe that the proposed approach can be used for decoding of Reed-Solomon codes of length $n < 2^m$, including cyclic Reed-Solomon codes. One can consider such codes as ones of length 2^m with $2^m - n$ punctured symbols. For decoding, one can consider these symbols as punctured.

3.3. Permutation decoding

It can be seen from the proof of the theorem that the set of frozen symbols, and therefore the successive cancellation decoding error probability, does not depend on the basis $a_0, \ldots, a_{m-1}$ of finite field $GF(2^m)$ being used. Different bases induce different permutations of the vector being decoded. But for a given noisy sequence the successfullness of its decoding does depend on the permutation being used. One can use this in order to reduce the decoding error probability.

The proposed approach consists in application of the above described sequential decoding algorithm to permuted instances of the received sequence, which are given by different bases $\mathcal{A}_i = (a_{i,0}, \ldots, a_{i,m-1})^T$, and selection of the most probable among the obtained codewords. More specifically, let us associate each element of the received sequence with some element $\beta \in GF(2^m)$, so that the corresponding codeword symbol is given by $f(\beta)$, $\deg f(x) < k$. Let us denote this element of the received sequence as y_β. The proposed decoding algorithm includes the following steps:

1. $i \leftarrow 0$, $\lambda \leftarrow 0$, $\widehat{\mathbf{c}} \leftarrow (0, \ldots, 0)$;
2. Construct vector z_0^{n-1}: $z_{\sum_{s=0}^{m-1} a_{i,s}j_s} = y_j$, where $j = \sum\limits_{s=0}^{m-1} j_s 2^s$, $j_s \in \{0, 1\}$, $0 \le j < n$;
3. Perform decoding of vector z_0^{n-1} using the algorithm presented in Section 3.2. Let c_0^{n-1} be the result of decoding and $\pi_i = \prod\limits_{j=0}^{n-1} P_{U_0|Y_0}(c_j \,|\, z_j)$;
4. If $\pi_i > \lambda$, let $\lambda \leftarrow \pi_i$, $\widehat{\mathbf{c}}_j \leftarrow c_{\sum_{s=0}^{m-1} a_{i,s}j_s}$, $0 \le j < n$;
5. $i \leftarrow i + 1$. If $i < N - 1$, go to step 2;

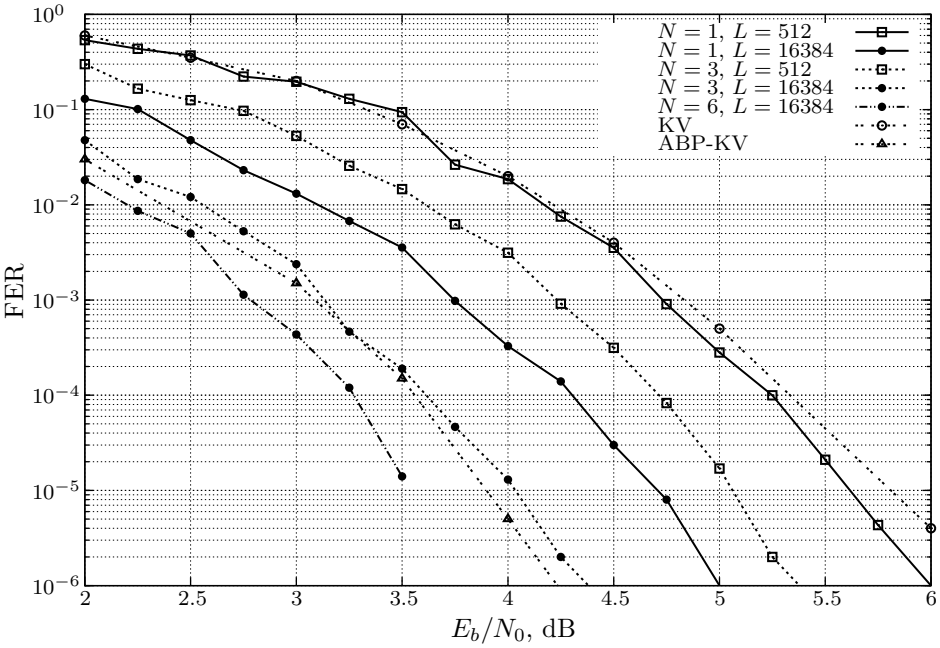

Fig. 1. $(31, 15, 16)$

6. Return codeword $\widehat{\mathbf{c}}$.

Hence, the complexity of this method can be estimated as $O(LNn^2 \log^2 n)$, or $O(LN(n \log^2 n + n^2))$ in the case of transmission of a binary image of the code, where L is the maximal list size and N is the number of bases being used.

Average decoding complexity can be significantly reduced if one stops the algorithm after obtaining the first codeword with probability $\lambda \geq \delta$, where δ is some pre-defined threshold.

4. NUMERIC RESULTS

Figure 1 presents simulation results for the case of transmission of a binary image of $(31, 15, 16)$ code over $GF(2^5)$ over the additive white Gaussian noise channel. The results are provided for the case of the proposed method with different values of list size L and number of bases N, as well as for the case of Koetter-Vardy method (KV) [1] and adaptive belief propagation algorithm combined with algebraic soft decision decoding (ABP-KV) [2]. In the latter case up to $N_2 = 155$ permutations of the received sequence were considered, and for each permutation up to $N_1 = 30$ iterations were performed, where each permutation consists in transforming of the check matrix of the binary image of the code to the canonical form on some $m(n-k)$ positions, $It_H = 3$ iterations of the belief propagation algorithm and execution of the Koetter-Vardy algorithm with refined symbol reliability values, as described in [2].

The results for the proposed algorithm were obtained for the case of bases $\mathcal{A}_i = A_i(\alpha^0, \alpha^1, \ldots, \alpha^4)^T$, where α is a primitive root of polynomial $x^5 + x^2 + 1$, which are given by first N matrices A_i in

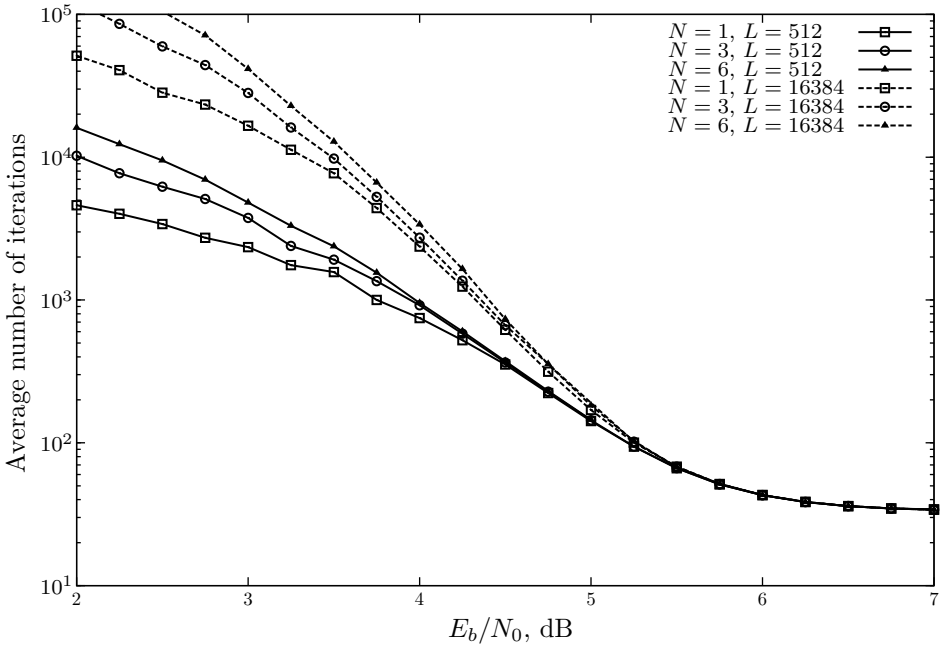

Fig. 2. $(31, 15, 16)$

the following list [3]:

$$\begin{pmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{pmatrix}, \begin{pmatrix} 0&0&0&0&1\\ 0&0&0&1&0\\ 0&0&1&0&0\\ 0&1&0&0&0\\ 1&0&0&0&0 \end{pmatrix}, \begin{pmatrix} 0&1&0&1&0\\ 1&0&0&0&0\\ 1&0&0&0&1\\ 1&0&1&0&0\\ 1&1&1&0&1 \end{pmatrix}, \begin{pmatrix} 1&1&1&0&1\\ 1&0&1&0&0\\ 1&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&1&0 \end{pmatrix}, \begin{pmatrix} 1&1&0&1&1\\ 1&0&0&0&1\\ 0&0&0&0&1\\ 1&0&0&1&0\\ 0&1&1&1&1 \end{pmatrix}, \begin{pmatrix} 0&1&1&1&1\\ 1&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&0&1\\ 1&1&0&1&1 \end{pmatrix}.$$

The presented results show that for sufficiently large list size the proposed method can provide better performance compared to the adaptive belief propagation algorithm combined with algebraic soft decision decoding [2], as well as Koetter-Vardy algorithm [1].

Figure 2 illustrates the dependency of the average number of iterations performed by the algorithm presented in Section 3.2 on the signal-to-noise ration for different values of N and L. It can be seen that increasing the signal-to-noise ratio causes the average complexity of the algorithm quickly to decrease rapidly, and converge eventually to the complexity of the classical successive cancellation algorithm.

5. CONCLUSIONS

In this section a novel decoding method for Reed-Solomon codes was proposed. The method is based on their representation as polar codes with dynamic frozen symbols. It was shown that this method can be slightly simplified in the case of decoding of a binary image of the code.

It was also shown that applying an analogue of the sequential decoding method introduced originally for the case of binary polar codes enables one to significantly reduce average complexity of the algorithm presented in the paper. Furthermore, the performance of the proposed algorithm can be improved by employing permutation decoding techniques. Simulation results show that the

[3] This set of bases was obtained empirically.

proposed algorithm provides better performance compared to the adaptive successive cancellation algorithm combined with algebraic soft-decision decoding, as well as Koetter-Vardy algorithm. A significant drawback of the proposed method is the need for very large list size (see parameter L in the results presented in § 4). Hence, a very important problem from the practical point of view is reduction of the amount of memory needed by the algorithm, as well as further reduction of the decoding complexity.

The proposed method can be, in principle, used for decoding of any linear block code. It is still an open question, however, to identify a class of codes (besides Reed-Solomon codes) and communication channels, such that the decoding error probability of the proposed method is sufficiently small.

The author thanks the reviewer for many helpful suggestions.

REFERENCES

1. Koetter, R. and Vardy, A., Algebraic Soft-Decision Decoding of Reed–Solomon Codes, *IEEE Trans. Inform. Theory*, 2003, vol. 49, no. 11, pp. 2809–2825.
2. El-Khamy, M. and McEliece, R., Iterative Algebraic Soft-Decision List Decoding of Reed–Solomon Codes, *IEEE J. Select. Areas Commun.*, 2006, vol. 24, no. 3, pp. 481–490.
3. Vardy, A. and Beery, Y., Bit-Level Soft-Decision Decoding of Reed–Solomon Codes, *IEEE Trans. Commun.*, 1991, vol. 39, no. 3, pp. 440–444.
4. Bellorado, J., Kavčić, A., Marrow, M., and Ping, L., Low-Complexity Soft-Decoding Algorithms for Reed–Solomon Codes—Part II: Soft-Input Soft-Output Iterative Decoding, *IEEE Trans. Inform. Theory*, 2010, vol. 56, no. 3, pp. 960–967.
5. Arikan, E., Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels, *IEEE Trans. Inform. Theory*, 2009, vol. 55, no. 7, pp. 3051–3073.
6. Tal, I. and Vardy, A., List Decoding of Polar Codes, in *Proc. 2011 IEEE Int. Sympos. on Information Theory (ISIT'2011), St. Petersburg, Russia, July 31–August 5, 2011*, pp. 1–5.
7. Chen, K., Niu, K., and Lin, J., Improved Successive Cancellation Decoding of Polar Codes, *IEEE Trans. Commun.*, 2013, vol. 61, no. 8, pp. 3100–3107.
8. Trifonov, P. and Miloslavskaya, V., Polar Codes with Dynamic Frozen Symbols and Their Decoding by Directed Search, in *Proc. 2013 IEEE Information Theory Workshop (ITW'2013), Sevilla, Spain, Sept. 9–13, 2013*, pp. 1–5.
9. Kolesnik, V.D. and Mironchikov, E.T., Cyclic Reed–Muller Codes and Their Decoding, *Probl. Peredachi Inf.*, 1968, vol. 4, no. 4, pp. 20–25 [*Probl. Inf. Trans.* (Engl. Transl.), 1968, vol. 4, no. 4, pp. 15–19].
10. Kasami, T., Lin, S., and Peterson, W., New Generalizations of the Reed–Muller Codes. I: Primitive Codes, *IEEE Trans. Inform. Theory*, 1968, vol. 14, no. 2, pp. 189–199.
11. Declercq, D. and Fossorier, M.P., Decoding Algorithms for Nonbinary LDPC Codes over $GF(q)$, *IEEE Trans. Commun.*, 2007, vol. 55, no. 4, pp. 633–643.
12. Tal, I. and Vardy, A., How to Construct Polar Codes, *IEEE Trans. Inform. Theory*, 2013, vol. 59, no. 10, pp. 6562–6582.
13. Chung, S.-Y., Richardson, T.J., and Urbanke, R.L., Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation, *IEEE Trans. Inform. Theory*, 2001, vol. 47, no. 2, pp. 657–670.
14. Trifonov, P., Efficient Design and Decoding of Polar Codes, *IEEE Trans. Commun.*, 2012, vol. 60, no. 11, pp. 3221–3227.
15. Abbe, E. and Telatar, E., Polar Codes for the m-User Multiple Access Channel, *IEEE Trans. Inform. Theory*, 2012, vol. 58, no. 8, pp. 5437–5448.