A Spectral Algorithm for Decoding Systematic BCH Codes

SERGEI VALENTINOVICH FEDORENKO[1]

[1] HSE University, HSE Campus in St. Petersburg, Department of Informatics, 3 Kantemirovskaya ulitsa, 194100, St. Petersburg, Russian Federation (e-mail: sfedorenko@hse.ru)

ABSTRACT A novel method of spectral decoding for systematic BCH codes has been proposed. This method has a simple description and a small computational complexity.

INDEX TERMS BCH codes, decoding, discrete Fourier transforms, error correction codes, fast Fourier transforms, Galois fields, Reed–Solomon codes.

I. INTRODUCTION

Bose–Chaudhuri–Hocquenghem (BCH) codes are used to correct errors in communication systems (especially in concatenated codes) and digital storage (including flash memory), as well as for many other purposes. Applying systematic encoders can often be of interest. Spectral methods in coding theory have been introduced and popularized by Blahut (see, for example, [3]). A spectral decoding algorithm for Reed–Solomon codes was invented by Shiozaki [31] and Gao [18] independently. A decoding method based on the Euclidean algorithm was proposed by Sugiyama *et al.* [32], [25, 12.8]. Currently only one spectral decoding algorithm for systematic Reed–Solomon codes by Mateer [27] is known. Thus, constructing the spectral decoding algorithm for systematic algebraic codes is a very actual problem.

In this letter we consider only one class of algebraic codes, the BCH codes. The description of the novel algorithms is simpler than, for example, the classical Peterson–Gorenstein–Zierler (PGZ) decoding algorithm. The computational complexity of the novel algorithms for some parameters is smaller than the computational complexity of the best decoding algorithms. A spectral decoding algorithm for BCH codes is useful, for example, for decoding the first component codes of the generalized error locating (GEL) code [9].

The novelty of the method proposed in this letter for spectral decoding systematic BCH codes consists of the following points:

1) the first spectral method for decoding systematic BCH codes;
2) applying the dimension of the Reed–Solomon code k_{RS} for decoding the BCH code with the same designed error-correction capability;
3) calculating the discrete Fourier *sub*transform.

The remainder of this letter is organized as follows. In Section II, we present basic notations and definitions. In Section III, the derivation of the main decoding algorithm is introduced. In Section IV, we propose the novel decoding algorithms. In Section V, we prove the correctness of the decoding algorithms. In Section VI, we calculate the computational complexity of the decoding algorithms. In Section VII, decoding examples are given. In Conclusion, several results are summarized.

II. BASIC NOTIONS AND DEFINITIONS

Every vector in the letter is associated with a polynomial.

Definition 1 ([5, Section 2.5]): An encoder where the data symbols are explicitly visible in the codeword is called a systematic encoder. The corresponding code is called a systematic code.

Definition 2 ([29]): The discrete Fourier transform (DFT) of blocklength n of a vector $f = (f_i)$, $i = 0, 1, \ldots, n-1$, $n \mid (p^m - 1)$, in the finite field $GF(p^m)$ is the vector $F = (F_j)$,

$$F_j = \sum_{i=0}^{n-1} f_i \alpha^{ij}, \quad j = 0, 1, \ldots, n-1,$$

where α is an element of order n in $GF(p^m)$.

Let us denote the DFT calculation by $F = \mathrm{DFT}(f)$ and the inverse DFT (IDFT) calculation by $f = \mathrm{IDFT}(F) = \mathrm{DFT}^{-1}(F)$.

A. SOME PROPERTIES OF THE DISCRETE FOURIER TRANSFORM

Every vector $f = (f_i)$, $i = 0, 1, \ldots, n-1$, is associated with a polynomial $f(x) = \sum_{i=0}^{n-1} f_i x^i$, and we have $F_j = f(\alpha^j)$.

FIGURE 1. The first scheme of the spectral decoding algorithm

step	time-domain $GF(2^m)$		transform-domain $GF(2)$
1	$T(x)$	$\xleftarrow{IDFT}$	$R(x)$
2	solve the key equation output: $M(x)$		
3	$M(x)$	$\xrightarrow{DFT}$	$C(x)$

FIGURE 2. The second scheme of the spectral decoding algorithm

step	time-domain $GF(2^m)$		transform-domain $GF(2)$
1	$T(x)$	$\xleftarrow{IDFT}$	$R(x)$
2	solve the key equation output: $W(x)$		
3	$W(x)$	$\xrightarrow{DFT}$	roots of $W(x)$; $E(x)$

Similarly, every vector $F = (F_j),\ \ j = 0, 1, \ldots, n-1$, is associated with a polynomial $F(x) = \sum_{j=0}^{n-1} F_j x^j$,

The polynomial $F(x)$ has a zero at an element α^i if and only if the $(-i)$th time component f_{-i} equals zero, where all indices are interpreted modulo n [4, Theorem 6.1.5]

$$F(\alpha^i) = 0 \quad \Longleftrightarrow \quad f_{-i} = 0. \tag{1}$$

Further, we consider only the finite fields of characteristic 2, and the computation field is the finite field $GF(2^m)$.

Theorem 1 ([3, Theorem 8.2.1], [4, Theorem 6.3.1]): Let f be a vector of blocklength n of elements $f_i \in GF(2^m)$ where n is a divisor of $2^m - 1$. The codeword components $F = (F_j),\ \ j = 0, 1, \ldots, n-1$, belong to the binary field $GF(2)$ if and only if the conjugacy constraints are satisfied:

$$f_i{}^2 = f_{2i}, \quad i = 0, 1, \ldots, n-1.$$

We assume that the blocklength of the DFT over $GF(2^m)$ is $n = 2^m - 1$. Let α be a primitive element of the finite field $GF(2^m)$.

B. THE SPECTRAL DECODING ALGORITHM

The idea of spectral decoding was first introduced in the original paper by Reed and Solomon in 1960 [30].

Let a codeword, a data polynomial, an error vector, and a received vector belong to the transform-domain. Any vector after the inverse discrete Fourier transformation belongs to the time-domain. In our algorithm a message polynomial, an error locator polynomial, and an interpolating polynomial belong to the time-domain.

The spectral decoding algorithm can be written as

1) The inverse transformation of the received vector into the time-domain.
2) The calculations in the time-domain.
3) The discrete Fourier transformation of the result of the previous step into the transform-domain (optional, if necessary).

We can see the schemes of the spectral decoding algorithms in Figures 1 and 2. The definitions of the received vector $R(x)$, the interpolating polynomial $T(x)$, the message polynomial $M(x)$, the codeword $C(x)$, the error locator

polynomial $W(x)$, and the error vector $E(x)$ are given in Section III.

Spectral decoding is a different way of looking at decoding algebraic codes and for some classes of codes and their parameters it may have the least computational complexity.

These definitions can be found in popular books on coding theory [3], [4], [5], [25].

III. THE DERIVATION OF THE MAIN DECODING ALGORITHM

Let $\mathcal{G}$ be the binary (n, k) BCH code, where n is the blocklength of the code, k is the dimension of the code, the spectrum of this code lies in the extension field $GF(2^m)$, $n \mid (2^m - 1)$, $g(x)$ is a generator polynomial for the code with $2t$ roots $\{\alpha^1, \alpha^2, \cdots, \alpha^{2t}\}$, t is the designed error-correction capability for the BCH code. The BCH code with this generator polynomial $g(x)$ is called a narrow-sense BCH code. The designed Hamming distance d of this BCH code $\mathcal{G}$ is $d = 2t + 1$.

For a binary symmetric channel, with error probability p [25, Fig. 1.1], a codeword error rate $P_{codeword} \approx 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i}$ [25, (26)]. A bit error rate P_{bit} satisfies the inequalities $\frac{1}{k} P_{codeword} \leq P_{bit} \leq P_{codeword}$ [25, Problem 25].

The remainder polynomial $\text{Rem}_{g(x)}[a(x)]$ is the calculation result of a long division of polynomial $a(x)$ by polynomial $g(x)$.

The codeword of the systematic code is

$$C(x) = x^{n-k} D(x) - \text{Rem}_{g(x)}[x^{n-k} D(x)], \tag{2}$$

where $D(x)$ is the data polynomial, $\deg D(x) < k$.

The received vector is represented as a polynomial

$$R(x) = \sum_{i=0}^{n-1} r_i x^i = C(x) + E(x) = \sum_{i=0}^{n-1} c_i x^i + \sum_{i=0}^{n-1} e_i x^i,$$

where $C(x)$ is the codeword, $E(x)$ is the error vector.

The ith error in the error vector $E(x)$ has a locator $Z_i \in \{\alpha^0, \alpha^1, \alpha^2, \cdots, \alpha^{n-1}\}$. The error locator polynomial is

$$W(x) = \prod_{i=1}^{\tau} (x - Z_i),$$

where τ is the actual number of errors, $\tau \leq t$. By definition $W(x) = 1$ if there are no errors.

Let $M(x) = \text{IDFT}(C(x)) = \sum_{i=0}^{n-1} m_i x^i$ be a message polynomial of the BCH code for the codeword $C(x)$, where $m_i \in GF(2^m)$, $i = 0, 1, \ldots, n-1$, and the conjugacy constraints are satisfied according to Theorem 1.

Let us define a dimension of the (n, k_{RS}) Reed–Solomon code $k_{RS} = n - 2t$, where n is the blocklength of the BCH code and t is the designed error-correction capability for the BCH code. Note that we use the parameter k_{RS} for decoding BCH codes. Since (1), we have $\deg M(x) < k_{RS}$ and $M(x) = \sum_{i=0}^{k_{RS}-1} m_i x^i$.

The derivation of the decoding algorithm is based on the patent [35] and the papers [21, Appendix], [11], [13]. An important novelty is in using the parameter k_{RS} instead the code dimension k.

Since $C(x) = \text{DFT}(M(x))$, we see that $c_i = M(\alpha^i)$, $i = 0, 1, \ldots, n-1$. From

$$\begin{cases} r_i = c_i = M(\alpha^i), & \text{if } r_i = c_i, \\ W(\alpha^i) = 0, & \text{if } r_i \neq c_i, \end{cases}$$

it follows that

$$W(\alpha^i) r_i = W(\alpha^i) M(\alpha^i), \quad i = 0, 1, \ldots, n-1.$$

Let $P(x) = W(x) M(x)$. Then

$$W(\alpha^i) r_i = P(\alpha^i), \quad i = 0, 1, \ldots, n-1.$$

Let us construct an interpolating polynomial $T(x) = \text{IDFT}(R(x))$ such that

$$T(\alpha^i) = r_i, \quad i = 0, 1, \ldots, n-1,$$

where $\deg T(x) < n$. Further,

$$W(\alpha^i) T(\alpha^i) = P(\alpha^i), \quad i = 0, 1, \ldots, n-1,$$
$$W(x)T(x) - P(x) = (x - \alpha^i) q_i(x), \quad i = 0, 1, \ldots, n-1,$$
$$W(x)T(x) - P(x) = (x^n - 1) q(x),$$

for some quotient polynomials $q_i(x)$ and $q(x)$.

Then the key equation is

$$\begin{cases} W(x)T(x) \equiv P(x) \mod x^n - 1 \\ \deg W(x) \leq t \\ \text{maximize } \deg W(x). \end{cases}$$

Considering that $\deg P(x) = \deg M(x) + \deg W(x) \leq (k_{RS} - 1) + t < n - t$, we have

$$\begin{cases} W(x)T(x) \equiv P(x) \mod x^n - 1 \\ \deg P(x) < n - t \\ \text{maximize } \deg P(x). \end{cases} \tag{3}$$

We solve the key equation by applying the extended Euclidean algorithm for polynomials (ExEAp) to $x^n - 1$ and

$T(x)$, and we obtain polynomials $P(x)$ and $W(x)$. The message polynomial is $M(x) = \frac{P(x)}{W(x)}$, the codeword is $C(x) = \text{DFT}(M(x))$, and the data polynomial is

$$D(x) = c_{n-k} + c_{n-k+1}x + \ldots + c_{n-1}x^{k-1}$$
$$\longleftrightarrow \quad D = (c_{n-k}, c_{n-k+1}, \ldots, c_{n-1}). \tag{4}$$

IV. THE DECODING ALGORITHMS

Input: The received vector $R(x) = \sum_{i=0}^{n-1} r_i x^i$
           $\longleftrightarrow \quad R = (r_0, r_1, \ldots, r_{n-1})$

Output: The data polynomial $D(x) = \sum_{i=0}^{k-1} d_i x^i$
           $\longleftrightarrow \quad D = (d_0, d_1, \ldots, d_{k-1})$

Algorithm 1: The main decoding algorithm

1) $T = \text{IDFT}(R) \quad \longleftrightarrow \quad T(x)$
2) Solve the congruence
$$\begin{cases} W(x)T(x) \equiv P(x) \mod x^n - 1 \\ \deg P(x) < n - t \\ \text{maximize } \deg P(x) \end{cases}$$
3) $M(x) = \frac{P(x)}{W(x)} \quad \longleftrightarrow \quad M$
4) $C = \text{DFT}(M)$
5) $D = (c_{n-k}, c_{n-k+1}, \ldots, c_{n-1})$

Algorithm 2: The alternative decoding algorithm

1) $T = \text{IDFT}(R) \quad \longleftrightarrow \quad T(x)$
2) Solve the congruence
$$\begin{cases} W(x)T(x) \equiv P(x) \mod x^n - 1 \\ \deg P(x) < n - t \\ \text{maximize } \deg P(x) \end{cases}$$
3) Find the roots of $W(x)$.
The error vector is $E = (e_0, e_1, \ldots, e_{n-1})$,
where $e_i = \begin{cases} 0, & \text{if } W(\alpha^i) \neq 0, \\ 1, & \text{if } W(\alpha^i) = 0, \end{cases}$ $i = 0, 1, \ldots, n-1$
4) $D = (r_{n-k} - e_{n-k}, r_{n-k+1} - e_{n-k+1}, \ldots, r_{n-1} - e_{n-1})$

We can see the differences of the proposed decoding algorithms and the algorithm [18], [11] in Table 1.

V. THE CORRECTNESS OF THE ALGORITHMS

There are one-to-one correspondences between the data polynomial $D(x)$ and the codeword $C(x)$ (see formulae (2) and (4)); and between the codeword $C(x)$ and the message polynomial $M(x)$ ($M(x) = \text{IDFT}(C(x))$ and $C(x) = \text{DFT}(M(x))$).

For decoding up to the designed error-correcting capability t for the BCH code the existence of a solution is obvious. The following theorem implies the correctness of the algorithms.

Theorem 2: For decoding up to the designed error-correcting capability t for the BCH code the decoding algorithm produces a unique data polynomial $D(x)$.

TABLE 1. The differences of the proposed decoding algorithms and the algorithm [18], [11]

property	algorithm [18], [11]	novel method
class of codes	non-systematic Reed–Solomon codes	systematic BCH codes
message polynomial $M(x)$	all nonzero coefficients are independent	all coefficients are under the conjugacy constraints (Theorem 1)
degree of message polynomial	$\deg M(x) < k$	$\deg M(x) < k_{RS} = n - 2t$
additional step	absent	$C(x) = \mathrm{DFT}(M(x))$ or roots of $W(x)$
stopping rule for solving the congruence	$\deg P(x) < \frac{n+k}{2}$	$\deg P(x) < n - t$
possibility of application the binary DFT calculation using the trace function	absent	available

Proof: Let us prove the unique solution for the congruence (3). The congruence (3) is satisfied by the true solution, which leads to the message polynomial $M(x)$, the codeword $C(x)$, and the data polynomial $D(x)$.

We solve the key equation (3) by applying the ExEAp to $x^n - 1$ and $T(x)$, the process stops when a remainder polynomial degree $\deg \widetilde{P}(x) < n - t$, and we obtain the pair of polynomials $\widetilde{P}(x)$ and $\widetilde{W}(x)$. If $\widetilde{P}(x)$ is divisible by $\widetilde{W}(x)$ then the polynomial $\widetilde{M}(x) = \dfrac{\widetilde{P}(x)}{\widetilde{W}(x)}$ becomes the message polynomial for BCH code and leads to the codeword $\widetilde{C}(x)$ and the data polynomial $\widetilde{D}(x)$.

Note that the ExEAp must be carried out from zero (preliminary) step, not from the first step, as usual. It is necessary to finish the ExEAp satisfying the constraint on the remainder polynomial degree $\deg \widetilde{P}(x) < n - t$, and sometimes at the additional (with zero remainder) step.

First, consider two singular cases.

Case 1. If the received vector is $R(x) = 0$ then $T(x) = 0$, and a formal notation of zero (preliminary) step for the ExEAp has the form

$$(x^n - 1)0 + T(x)1 = 0$$

and $\widetilde{W}(x) = 1$, $\widetilde{P}(x) = \widetilde{M}(x) = \widetilde{C}(x) = \widetilde{D}(x) = 0$.

Case 2. If no errors occur and the received vector is $R(x) \neq 0$, then zero (preliminary) step for the ExEAp has the form

$$(x^n - 1)0 + T(x)1 = T(x),$$

where $T(x) = M(x)$, $\deg T(x) = \deg M(x) < k_{RS} < n - t$, $\widetilde{W}(x) = 1$, $T(x) = \widetilde{P}(x) = \widetilde{M}(x) = M(x)$, $C(x) = \widetilde{C}(x)$, and $D(x) = \widetilde{D}(x)$.

Next, consider the main case when there are errors.

Case 3. $E(x) \neq 0$.

The ExEAp is finished when it satisfies the constraint on the remainder polynomial degree $\deg \widetilde{P}(x) < n - t$. From the property for the Bézout polynomials [25, 12.8] it follows that at this step $\deg \widetilde{W}(x) \leq n - (n - t) = t$. After the calculations

$$P(x)\widetilde{W}(x) \equiv \left(W(x)T(x)\right)\widetilde{W}(x) = W(x)\left(T(x)\widetilde{W}(x)\right)$$

$$\equiv W(x)\widetilde{P}(x) \mod x^n - 1,$$

we get the congruence

$$P(x)\widetilde{W}(x) \equiv W(x)\widetilde{P}(x) \mod x^n - 1.$$

The degree of each side of this congruence is less than n and we obtain the equation for polynomials

$$P(x)\widetilde{W}(x) = W(x)\widetilde{P}(x).$$

After two divisions we have $\widetilde{P}(x) = \dfrac{P(x)}{W(x)}\widetilde{W}(x) = M(x)\widetilde{W}(x)$, $M(x) = \dfrac{\widetilde{P}(x)}{\widetilde{W}(x)} = \widetilde{M}(x)$, $C(x) = \widetilde{C}(x)$, and $D(x) = \widetilde{D}(x)$.

Both solutions of the congruence (3) coincide, as well as the codewords and data polynomials. This completes the proof of the theorem. ∎
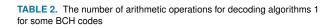
VI. COMPUTATIONAL COMPLEXITY OF THE DECODING ALGORITHMS

A. NUMBER OF OPERATIONS FOR THE DECODING ALGORITHMS

Let us write the upper bounds on the number of arithmetic operations in the computation field $GF(2^m)$. First, consider direct methods for computing each step of decoding algorithm 1.

Algorithm 1: The main decoding algorithm
1. The product of a vector by a matrix: $2n^2$.
2. The solution of the congruence consists of t steps. At each step, division of polynomials ($4n$ operations) and calculation of one Bézout polynomial ($4t$ operations) are performed: $t(4n + 4t) = 4t(n + t)$.
3. The division of a polynomial $P(x)$ of degree $n - t$ by a polynomial $W(x)$ of degree t is performed in $n - 2t$ steps, each step is performed in $2t$ operations: $2t(n - 2t)$.
4. The product of a vector by a matrix: $2n^2$.
5. Extracting a subvector D from a vector C does not require arithmetic operations: 0.

TABLE 2. The number of arithmetic operations for decoding algorithms 1 for some BCH codes

code parameters	t	PGZ algorithm	algorithm 1
$(63, 36)$	5	2995	17766
$(63, 30)$	6	4249	18144
$(63, 24)$	7	5922	18522
$(63, 18)$	10	14615	19656
$(63, 16)$	11	19184	20034
$(63, 10)$	13	31759	20790
$(63, \ 7)$	15	50110	21546

TABLE 3. The number of binary operations for fast decoding algorithm 2 for some BCH codes

code parameters	t	PGZ/BM algorithm	algorithm 2
$(63, 36)$	5	39785	23723
$(63, 30)$	6	50478	28872
$(63, 24)$	7	62083	34957
$(63, 18)$	10	102370	58828
$(63, 16)$	11	117623	68657
$(63, 10)$	13	150865	91123
$(63, \ 7)$	15	187755	117333

The main decoding algorithm requires about $4n^2 + 6tn$ arithmetic operations in the computation field $GF(2^m)$. For comparison, the complexity of the direct implementation of the classical PGZ decoding algorithm [3, Fig. 7.1] is about $6tn + \frac{1}{2}t^4 + \frac{16}{3}t^3 + 5t^2 + \frac{1}{6}t$ arithmetic operations.

Table 2 presents the complexity in terms of the number of arithmetic operations in the computation field $GF(2^m)$ of novel decoding algorithm 1 and the classical Peterson–Gorenstein–Zierler (PGZ) decoding algorithm [3, Fig. 7.1]. The BCH codes have parameters (blocklength, dimension) and t is the designed error-correction capability for the BCH code.

Algorithm 2: The alternative decoding algorithm
To solve the key equation (3), one can use not only the extended Euclidean algorithm for polynomials (ExEAp), but also the Berlekamp–Massey (BM) algorithm. Using the polynomial $T(x)$, which contains redundant coefficients, the error locator polynomial $W(x)$ is calculated under the constraint $\deg W(x) \le t$.

The BM and ExEAp (especially its fast implementation) algorithms are equivalent and have almost the same complexity [10], [22], [26]. Moreover, the description of the ExEAp is much simpler than the description of the BM algorithm.

Further, consider alternative decoding algorithm 2.

1. The IDFT calculations via binary IDFT [17].
2. The fast calculation of an error locator polynomial $W(x)$ of degree t via fast ExEAp [18, Algorithm 1a, Step 2], [28].
3. The DFT calculations via cyclotomic DFT [33] and improved Goertzel–Blahut algorithm [15].
4. Subtracting a subvector from a subvector.

The classical PGZ decoding algorithm with the BM algorithm for solving the key equation [3, Fig. 7.5] has complexity about $(2tn+t)m+(6t^2 \cdot 2m^2+4t^2m)+tn(2m^2+m)+t$ binary operations. Algorithm 2 with fast ExEAp calculating for $GF(2^6)$ has complexity about $(50m+552)+(6t^2(2m^2+m))+(88 \cdot 2m^2+805m)+t$ binary operations.

Table 3 presents the complexity in terms of the number of binary operations of novel decoding algorithm 2 and the classical Peterson–Gorenstein–Zierler decoding algorithm with the Berlekamp–Massey (PGZ/BM) algorithm.

B. APPLICATIONS OF FAST ALGORITHMS FOR COMPUTING EACH STEP OF THE DECODING ALGORITHMS

Next, consider fast methods for computing each step of the decoding algorithms.

Algorithm 1: The main decoding algorithm

1: The DFT (or IDFT) calculations over finite fields with the best asymptotic complexity were published in the papers [19], [23], [24]. To minimize the number of multiplications, there is the cyclotomic DFT algorithm [33], [12]. There are several improvements of this algorithm to reduce the number of multiplications [1], [14], [15], and the number of additions [1], [2], [7], [8], [34], [36], [37].

2: The fast ExEAp is introduced in [28].

3: The fast division algorithm with remainder is reported in [20, Algorithm 9.5].

4: The calculation of the discrete Fourier subtransform can be performed using the trace function [17]. Example is considered in Subsection VII.C.

5: It is trivial.

Algorithm 2: The alternative decoding algorithm

3: The new review of the best methods for finding roots of polynomials over finite field $GF(2^m)$ is published in [16].

4: It is trivial.

C. ASYMPTOTIC COMPLEXITY OF THE DECODING ALGORITHMS

Finally, consider the asymptotic complexity for computing each step of the decoding algorithms.

Algorithm 1: The main decoding algorithm

1: $O(n \log n)$
2: $O(n(\log n)^2)$
3: $O(n \log n \log \log n)$
4: $O(n \log n)$
5: $O(n)$

Algorithm 2: The alternative decoding algorithm
1:     $O(n \log n)$
2:     $O(n(\log n)^2)$
3:     $O(n \log n)$
4:     $O(n)$

The asymptotic complexity of the decoding algorithms is about $O(n(\log n)^2)$. Note that the asymptotic complexity of algebraic codes decoding lies between $O(n \log n)$ and $O(n(\log n)^2)$ [25, Notes on Chapter 12].

VII. EXAMPLES

Let $\mathcal{G}$ be the binary $(7, 4)$ BCH code, where $n = 7$ is the blocklength of the code, $k = 4$ is the dimension of the code, the spectrum of this code lies in the extension field $GF(2^3)$, α is a root of the primitive polynomial $x^3 + x + 1$. Let the designed error-correction capability for the BCH code be $t = 1$. Then the elements $\{\alpha^1, \alpha^2\}$ are the roots of the generator polynomial, and the generator polynomial for the code $\mathcal{G}$ is $g(x) = x^3 + x + 1$. The dimension of the Reed–Solomon code is $k_{RS} = n - 2t = 5$. The position locations are $(\alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6)$.

Let the received vector be $R = (0001001)$.

A. THE MAIN DECODING ALGORITHM

1) $T = \text{IDFT}(R) = (0001001)$

$$\times \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 \\ \alpha^0 & \alpha^5 & \alpha^3 & \alpha^1 & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^0 & \alpha^4 & \alpha^1 & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \\ \alpha^0 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha^1 & \alpha^4 \\ \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \end{pmatrix}$$

$= (0\alpha^2\alpha^4\alpha^2\alpha^1\alpha^1\alpha^4)$
$\longleftrightarrow \alpha^2 x + \alpha^4 x^2 + \alpha^2 x^3 + \alpha^1 x^4 + \alpha^1 x^5 + \alpha^4 x^6 = T(x)$.

2) Solve the congruence

$$W(x)T(x) \equiv P(x) \mod x^7 - 1.$$

Step ExEAp 0: $(x^7 - 1)0 + T(x)1 = T(x)$.
Step ExEAp 1: $x^7 - 1 = T(x)(\alpha^0 + \alpha^3 x)$
$+ (\alpha^0 + \alpha^2 x + \alpha^0 x^2 + \alpha^6 x^3 + \alpha^6 x^4 + \alpha^2 x^5)$; we obtain
$P(x) = \alpha^0 + \alpha^2 x + \alpha^0 x^2 + \alpha^6 x^3 + \alpha^6 x^4 + \alpha^2 x^5$,
$W(x) = \alpha^0 + \alpha^3 x$, the condition $\deg P(x) = 5 < n - t = 6$ is the stopping rule, and $(x^7 - 1)1 + T(x)W(x) = P(x)$.

3) $M(x) = \frac{P(x)}{W(x)}$

$$= \frac{\alpha^0 + \alpha^2 x + \alpha^0 x^2 + \alpha^6 x^3 + \alpha^6 x^4 + \alpha^2 x^5}{\alpha^0 + \alpha^3 x}$$

$= \alpha^0 + \alpha^5 x + \alpha^3 x^2 + \alpha^6 x^4$
$\longleftrightarrow (\alpha^0 \alpha^5 \alpha^3 0 \alpha^6 00) = M$.

4) $C = \text{DFT}(M) = (\alpha^0 \alpha^5 \alpha^3 0 \alpha^6 00)$

$$\times \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ \alpha^0 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha^1 & \alpha^4 \\ \alpha^0 & \alpha^4 & \alpha^1 & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \\ \alpha^0 & \alpha^5 & \alpha^3 & \alpha^1 & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^0 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 \end{pmatrix} \quad (5)$$

$= (0001101)$.
5) $D = (1101)$.

B. THE ALTERNATIVE DECODING ALGORITHM

1) $T = \text{IDFT}(R)$
$\longleftrightarrow \alpha^2 x + \alpha^4 x^2 + \alpha^2 x^3 + \alpha^1 x^4 + \alpha^1 x^5 + \alpha^4 x^6 = T(x)$.
2) $W(x) = \alpha^0 + \alpha^3 x \longleftrightarrow (\alpha^0 \alpha^3 00000) = W$.
3) Find the roots of $W(x)$.
$\text{DFT}(W) = (\alpha^0 \alpha^3 00000)$

$$\times \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^0 & \alpha^2 & \alpha^4 & \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ \alpha^0 & \alpha^3 & \alpha^6 & \alpha^2 & \alpha^5 & \alpha^1 & \alpha^4 \\ \alpha^0 & \alpha^4 & \alpha^1 & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \\ \alpha^0 & \alpha^5 & \alpha^3 & \alpha^1 & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^0 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 \end{pmatrix}$$

$= (\alpha^1 \alpha^5 \alpha^4 \alpha^2 0 \alpha^3 \alpha^6)$.
The error vector is $E = (0000100)$.
4) $C = R - E = (0001101)$. $D = (1101)$.

C. THE DISCRETE FOURIER SUBTRANSFORM

Consider implementation of the union of steps 4 and 5 for the main decoding algorithm. The DFT is called the discrete Fourier subtransform if zero rows and columns of the transform matrix are deleted. We delete zero rows and columns of the transform matrix (5), and get the formula for the discrete Fourier subtransform calculation

$$D = (\alpha^0 \alpha^5 \alpha^3 \alpha^6) \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \alpha^0 \\ \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ \alpha^6 & \alpha^1 & \alpha^3 & \alpha^5 \\ \alpha^5 & \alpha^2 & \alpha^6 & \alpha^3 \end{pmatrix}$$

$= \big(1 + \text{trace}(\alpha^1), 1 + \text{trace}(\alpha^2), 1 + \text{trace}(\alpha^3),$
$1 + \text{trace}(\alpha^4) \big) = (1101)$, where the binary trace of an element $\beta \in GF(2^3)$ is $\text{trace}(\beta) = \beta + \beta^2 + \beta^4$.

Depending on the implementation, the multiplication operations may not be required at all. The details of the calculation of the DFT using the trace function will be published in a separate paper [17].

VIII. CONCLUSION

A novel method of spectral decoding for systematic BCH codes has been proposed. The computational complexity of

the novel method's direct implementation for some parameters is smaller than the computational complexity of the direct implementation of the classical decoding algorithms. The fast implementation of the novel method requires fewer operations than the fast implementation of the classical decoding algorithms. The new method is recommended especially for large values of the designed error-correction capability.

REFERENCES

[1] S. Bellini, M. Ferrari, and A. Tomasoni, "On the structure of cyclotomic Fourier transforms and their applications to Reed–Solomon codes," *IEEE Transactions on Communications*, vol. 59, no. 8, pp. 2110–2118, 2011.

[2] S. Bellini, M. Ferrari, and A. Tomasoni, "On the reduction of additive complexity of cyclotomic FFTs," *IEEE Transactions on Communications*, vol. 60, no. 6, pp. 1465–1468, 2012.

[3] R. E. Blahut, *Theory and Practice of Error Control Codes*. Reading, MA: Addison-Wesley, 1983.

[4] R. E. Blahut, *Algebraic Codes for Data Transmission*. Cambridge, U.K.: Cambridge University Press, 2003.

[5] R. E. Blahut, *Algebraic Codes on Lines, Planes, and Curves: An Engineering Approach*. Cambridge, U.K.: Cambridge University Press, 2008.

[6] R. E. Blahut, *Fast Algorithms for Signal Processing*. Cambridge, U.K.: Cambridge University Press, 2010.

[7] N. Chen and Z. Yan, "Cyclotomic FFTs with reduced additive complexities based on a novel common subexpression elimination algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1010–1020, 2009.

[8] N. Chen and Z. Yan, "Reduced-complexity Reed-Solomon decoders based on cyclotomic FFTs," *IEEE Signal Processing Letters*, vol. 16, no. 4, pp. 279–282, 2009.

[9] A. L. Chmora, S. V. Fedorenko, and V. V. Zyablov, "Method and device for encoding/decoding data by using m-th order GEL codes," Patent. WO 2017078562 A1 dated 11 May 2017.

[10] J. Dornstetter, "On the equivalence between Berlekamp's and Euclid's algorithms," *IEEE Transactions on Information Theory*, vol. 33, no. 3, pp. 428–431, May 1987.

[11] S. V. Fedorenko, "A simple algorithm for decoding Reed–Solomon codes and its relation to the Welch–Berlekamp algorithm," *IEEE Transactions on Information Theory*, vol. IT–51, no. 3, pp. 1196–1198, Mar. 2005.

[12] S. V. Fedorenko, "A method for computation of the discrete Fourier transform over a finite field," *Problems of Information Transmission*, vol. 42, no. 2, pp. 139–151, 2006. Translation of *Problemy Peredachi Informatsii*.

[13] S. V. Fedorenko, "A simple algorithm for decoding algebraic codes," *Information and Control Systems*, no. 3, pp. 23–27, 2008, (in Russian). [Online]. Available: http://www.i-us.ru/index.php/ius/article/view/14757

[14] S. V. Fedorenko, "The discrete Fourier transform over a finite field with reduced multiplicative complexity," Proceedings of the IEEE International Symposium on Information Theory at Saint-Petersburg, Russia, pp. 1200–1204, July–August 2011.

[15] S. V. Fedorenko, "Improving the Goertzel–Blahut algorithm," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 824–827, 2016.

[16] S. V. Fedorenko, "Efficient algorithm for finding roots of error-locator polynomials," *IEEE Access*, vol. 9, pp. 38673–38686, 2021.

[17] S. V. Fedorenko, "The discrete Fourier transform over the binary finite field," *IEEE Signal Processing Letters*, vol. 29, 2022, submitted for publication.

[18] S. Gao, "A new algorithm for decoding Reed-Solomon codes," in *Communications, Information and Network Security*, V. Bhargava, H. V. Poor, V. Tarokh, and S.Yoon, Eds. Norwell, MA: Kluwer, 2003, vol. 712, pp. 55–68.

[19] S. Gao and T. Mateer, "Additive fast Fourier transforms over finite fields," *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6265–6272, 2010.

[20] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*. Third edition. Cambridge, U.K.: Cambridge University Press, 2013.

[21] P. Gemmell and M. Sudan, "Highly resilient correctors for polynomials," *Information Processing Letters*, vol. 43, no. 4, pp. 169–174, 1992.

[22] A. E. Heydtmann and J. M. Jensen, "On the equivalence of the Berlekamp–Massey and the Euclidean algorithms for decoding," *IEEE Transactions on Information Theory*, vol. 46, no. 7, pp. 2614–2624, Nov. 2000.

[23] S.-J. Lin, T. Y. Al-Naffouri, and Y. S. Han, "FFT algorithm for binary extension finite fields and its application to Reed–Solomon codes," *IEEE Transactions on Information Theory*, vol. 62, no. 10, pp. 5343–5358, 2016.

[24] S.-J. Lin, T. Y. Al-Naffouri, Y. S. Han, and W.-H. Chung, "Novel polynomial basis with fast Fourier transform and its application to Reed–Solomon erasure codes," *IEEE Transactions on Information Theory*, vol. 62, no. 11, pp. 6284–6299, 2016.

[25] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam–New York–Oxford: North-Holland Publishing Company, 1977.

[26] T. D. Mateer, "On the equivalence of the Berlekamp–Massey and the Euclidean algorithms for algebraic decoding," 12th Canadian Workshop on Information Theory, pp. 139–142, 2011.

[27] T. D. Mateer, "Simple algorithms for decoding systematic Reed–Solomon codes," *Designs, Codes and Cryptography*, vol. 69, pp. 107–121, 2013.

[28] R. T. Moenck, "Fast computation of GCDs," in *Proceedings of the fifth annual ACM symposium on Theory of computing*, Austin, TX, 1973, pp. 142–151.

[29] J. M. Pollard. "The fast Fourier transform in a finite field," *Mathematics of Computation*, vol. 25, no. 114, pp. 365–374, 1971.

[30] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun. 1960.

[31] A. Shiozaki, "Decoding of redundant residue polynomial codes using Euclid's algorithm," *IEEE Transactions on Information Theory*, vol. IT–34, no. 5, pp. 1351–1354, Sep. 1988.

[32] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A method for solving key equation for decoding Goppa codes," *Information and Control*, vol. 27, pp. 87–99, 1975.

[33] P. V. Trifonov and S. V. Fedorenko, "A method for fast computation of the Fourier transform over a finite field," *Problems of Information Transmission*, vol. 39, no. 3, pp. 231–238, 2003. Translation of *Problemy Peredachi Informatsii*.

[34] P. Trifonov, "On the additive complexity of the cyclotomic FFT algorithm," Proceedings of the IEEE Information Theory Workshop at Lausanne, Switzerland, pp. 537–541, September 2012.

[35] L. Welch and E. R. Berlekamp, "Error correction for algebraic block codes," U.S. Patent 4 633 470, Sep. 27, 1983.

[36] X. Wu, M. Wagh, N. Chen, Y. Wang, and Z. Yan, "Composite cyclotomic Fourier transforms with reduced complexities," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2136–2145, 2011.

[37] X. Wu, Y. Wang, and Z. Yan, "On algorithms and complexities of cyclotomic fast Fourier transforms over arbitrary finite fields," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1149–1158, 2012.

SERGEI VALENTINOVICH FEDORENKO was born in St. Petersburg, U.S.S.R., in 1967. He received his Ph.D. degree in computer science and his Doctor of Technical Science degree from the St. Petersburg State University of Airspace Instrumentation, St. Petersburg, Russia, in 1994 and 2009, respectively.

Currently, he is a Leading Research Fellow/ Professor in the National Research University Higher School of Economics, HSE Campus in St. Petersburg, Department of Informatics, Russian Federation. Prof. Fedorenko is the Alexander von Humboldt Foundation Research Fellow at the Technische Universität Darmstadt, Darmstadt, Germany, since 1999.

His research interests include error-correcting codes, decoding algorithms, discrete Fourier transform over finite fields, and fast algorithms.

Prof. Fedorenko was the Organizer (2008), the Vice-Chair (2008–2009), and the Chair (2010–2012) of the IEEE Russia, Russia (Siberia), and Russia (Northwest) Joint Sections Information Theory Society Chapter.

• • •