Caruso, F., Orsini, E., Sala, M., & Tinnirello, C. (2017). On the Shape of the General Error Locator Polynomial for Cyclic Codes. IEEE Transactions on Information Theory, 63(6), 3641-3557. DOI: 10.1109/TIT.2017.2692213

On the shape of the general error locator polynomial for cyclic codes

Fabrizio Caruso, Emmanuela Orsini, Massimiliano Sala, Claudia Tinnirello

***Abstract*—General error locator polynomials were introduced in 2005 as an alternative decoding for cyclic codes. We present now a conjecture on their sparsity which would imply polynomial-time decoding for all cyclic codes. A general result on the explicit form of the general error locator polynomial for all cyclic codes is given, along with several results for specific code families, providing evidence to our conjecture. From these, a theoretical justification of the sparsity of general error locator polynomials is obtained for all binary cyclic codes with $t \leq 2$ and $n < 105$, as well as for $t = 3$ and $n < 63$, except for some cases where the conjectured sparsity is proved by a computer check. Moreover, we summarize all related results, previously published, and we show how they provide further evidence to our conjecture. Finally, we discuss the link between our conjecture and the complexity of bounded-distance decoding of cyclic codes.**

***Index Terms*—Cyclic codes, bounded-distance decoding, general error locator polynomial, symmetric functions, computational algebra, finite fields, Groebner basis.**

I. Introduction

This paper focuses primarily on some issues concerning the efficiency of bounded-distance decoding for cyclic codes. Cyclic codes form a large class of widely used error correcting codes. They include important codes such as the Bose-Chaudhuri-Hocquenghem (BCH) codes, quadratic residue (QR) codes and Golay codes. In the last fifty years many efficient bounded-distance decoders have been developed for special classes, e.g. the Berlekamp-Massey (BM) algorithm ([1]) designed for the BCH codes. Although BCH codes can be decoded efficiently, it is known that their decoding performance degrades as the length increases ([2]). Cyclic codes are not known to suffer from the same distance limitation, but no efficient bounded-distance decoding algorithm is known for them (up to their actual distance).

On the other hand, the BM algorithm can also be applied to some cyclic codes, if there are enough consecutive known syndromes; namely, $2t$ consecutive syndromes are needed to correct a corrupted word with at most t errors. Unfortunately, for an arbitrary cyclic code the number of consecutive known syndromes is less than $2t$. When few unknown syndromes are needed to get $2t$ consecutive syndromes, it is sometimes possible to determine expressions of unknown syndromes in terms of known syndromes. In [3] Feng and Tzeng proposed a matrix method which is based on the existence of a syndrome matrix with a particular structure. This method depends on the weight of the error pattern, so it leads to a step-by-step decoding algorithm, and hence the error locator polynomial may not be determined in one step. In [4] He et al. developed a modified version of the Feng-Tzeng method, and used it to determine the needed unknown syndrome and to decode the binary QR code of length 47. In [5], [6], [7] Chang et al. presented algebraic decoders for other binary QR codes combining the Feng-He matrix method and the BM algorithm. Another method used to yield representations of unknown syndromes in terms of known syndromes is the Lagrange interpolation formula (LIF)[8]. This method has two limitations: it can be applied only to codes generated by irreducible polynomials and its computational time grows substantially as the number of errors increases. The first problem was overcome by Chang et al. in [9]. Here the authors introduced a multivariate interpolation formula (MVIF) over finite fields and used it to get an unknown syndrome representation method similar to that in [8]. They also apply the MVIF to obtain the coefficients of the general error locator polynomial of the $[15, 11, 5]$ Reed-Solomon (RS) code. Later, trying to overcome the second problem, Lee et al.[10] presented an algorithm which combines the syndrome matrix search and the modified Chinese remainder theorem (CRT). Compared to the Lagrange interpolation method, this substantially reduces the computational time for binary cyclic codes generated by irreducible polynomials.

Besides the unknown syndrome representation method, other approaches have been proposed to decode cyclic codes. In 1987 Elia [11] proposed a seminal efficient algebraic decoding algorithm for the Golay code of length 23. Orsini and Sala [12] introduced

Fabrizio Caruso is with ASTEK, France (email: caruso.fabrizio@gmail.com)

Emmanuela Orsini is with the Department of Computer Science, University of Bristol, UK (email: emmanuela.orsini@bristol.ac.uk).

Massimiliano Sala and Claudia Tinnirello are with the Department of Mathematics, University of Trento, Italy.

Emmanuela Orsini was supported in part by EPSRC via grant EP/N021940/1.

Corresponding authors: Massimiliano Sala (email: maxsalacodes@gmail.com) and Claudia Tinnirello (email: claudia.tinnirello@gmail.com).

the general error locator polynomials and presented an algebraic decoder which permits to determine the correctable error patterns of a cyclic code in one step. They constructively showed that a general error locator polynomial exists for any cyclic code (this locator polynomial is shown to exist in a Gröbner basis of the syndrome ideal), and they gave some theoretical results on the structure of such polynomials in [13], without the need to actually compute a Gröbner basis. In particular, for all binary cyclic codes with length less than 63 and correction capability less or equal than 2, they provided a sparse implicit representation, and showed that most of these codes may be grouped in a few classes, each allowing a theoretical interpretation for an explicit sparse representation. In any case, direct computer computations show that the general error locator polynomial for all these codes is actually sparse.

The efficiency of the decoding based on general error locator polynomials depends on their sparsity. There is no (known) theoretical proof that any cyclic code admits a sparse general locator, but there is some experimental evidence in the binary case. The proof of its sparsity in the general case may be of interest in complexity theory, since it would imply that the complexity of the bounded-distance decoding problem for cyclic codes (allowing unbounded preprocessing) is polynomial-time in the code length. Yet, no published article contains a formal definition for this sought-after sparsity, therefore the link with complexity theory is unclear.

In [13] the authors also provide a structure theorem for the general error locator polynomials of a class of binary cyclic codes. A generalization of this result is given in [8]. The low computational complexity of the general error locator polynomial for the two error-correctable cyclic codes has motivated the studies for variations on this polynomial [14], [15], [16]. We note that Gröbner bases could also be used for online decoding. In [17] Augot et al. proposed an online Gröbner basis decoding algorithm which consists of computing for each received word a Gröbner basis of the syndrome ideal with the Newton identities, in order to express the coefficients of the error locator polynomial in terms of the syndromes of the received word.

Our results

In what follows, we list the main original contributions of this paper.

- We introduce the notion of *functional density* for a general locator, which allows to formalize the notion of *sparse general locator*. Thanks to this formalization, we can present a rigorous conjecture on the locator sparsity and its first consequence on complexity theory.
- We give a general result on the structure of the general error locator polynomial for *all* cyclic codes, which generalizes Theorem 1 of [8].
- We provide some results on the general error locator polynomial for several families of binary cyclic codes with $t \leq 3$, adding theoretical evidence to the sparsity of the general error locator polynomial for infinite classes of codes.
- As a first direct consequence to $t = 2$, we theoretically justify the sparsity of the general error locator polynomial for all the five remaining cases which were not classified in [13].
- As a second direct consequence to $t = 3$, we classify the cyclic codes with $n < 63$ and $t = 3$ according to the shape of their general error locator polynomial, justifying theoretically the results for all cases except three. For the remaining three cases, the general error locator polynomial can be computed explicitly.
- Finally, we provide some more results on the complexity of bounded-distance decoding of some classes of cyclic codes. Some results are conditioned to our conjecture and others hold unconditionally.

Paper organization

The remainder of the paper is organized as follows. In Section II we review some definitions concerning cyclic codes: we recall Cooper's philosophy, the notion of general error locator polynomial for cyclic codes and how this polynomial can be use to decode. In Section III we state our conjecture on the sparsity of locators and we identify a first link with the complexity of decoding cyclic codes. In Section IV we show our main result, Theorem 19 which provides a general structure of the error locator polynomial for all cyclic codes. In Section V we present an infinite class of binary cyclic codes along with an explicit formula to represent a sparse general error locator polynomial for any code in this family. This single family covers 4195 codes out of the 4810 codes with $t = 2$ and length < 105. For the remaining 615 codes, explicit sparse general locators have been computed. As a comparison, observe that [13] dealt with $t = 2$ and $n < 63$, that is, with a total of only 952 codes. In Section VI we provide a general error locator polynomials for all binary cyclic codes with $t = 3$ and $n < 63$. A sparse explicit representation is theoretically justified for all cases, except three. We also give new results on the structure of the general error locator polynomial for some *infinite* classes of binary cyclic codes with $t = 3$. In Section VII we analyze more deeply the links between our conjecture, some related results by other authors and complexity theory. In Section VIII, we draw some conclusions.

II. Preliminaries

In this section we review standard notation. The reader is referred to [18], [19] and [20] for general references on coding theory.
Throughout the paper we adopt the following conventions. n denotes an odd number $n \geq 3$. For any two integers a and b, (a, b) denotes their greatest common divisor (written as a non-negative integer). Vectors are denoted by bold lower-case letters.

A. Some algebraic background and notation

Let $q = p^s$, where $p \geq 2$ is any prime and $s \geq 1$ is any positive integer. In this paper, $\mathbb{F}_q$ denotes the finite field with q elements.

Sometimes we will deal with rational expressions of the kind $\frac{f}{g}$, with $f, g \in \mathbb{F}_q[x_1, \ldots, x_\ell]$ for some $\ell \geq 1$. When we evaluate this expression at any point $P \in (\mathbb{F}_q)^\ell$ it is possible that $g(P) = 0$. However, our rational expressions are evaluated only in points such that if $g(P) = 0$ then also $f(P) = 0$, and when this happens we always use the convention that $\frac{f(P)}{g(P)} = 0$.

B. Cyclic codes

A linear code C is a cyclic code if it is invariant under any cyclic shift of the coordinates. Cyclic codes have been extensively studied in coding theory for their useful algebraic properties. We only consider $[n, k, d]_q$ cyclic codes with $(n, q) = 1$, that is n and q are coprime. Let $R = \mathbb{F}_q[\eta]/(\eta^n - 1)$, each vector $\mathbf{c} \in (\mathbb{F}_q)^n$ is associated to a polynomial $c_0 + c_1\eta + \cdots + c_{n-1}\eta^{n-1} \in R$, and it is easy to prove that cyclic codes of length n over $\mathbb{F}_q$ are ideals in R. Let $\mathbb{F}_{q^m}$ be the splitting field of $\eta^n - 1$ over $\mathbb{F}_q$, and let α be a primitive n-th root of unity over $\mathbb{F}_q$, then it holds $\eta^n - 1 = \prod_{i=0}^{n-1}(\eta - \alpha^i)$. For the rest of the paper, we assume that, given q and n, the primitive root α is fixed. Let $g(\eta) \in \mathbb{F}_q[\eta]$ be the generator polynomial of an $[n, k, d]_q$-cyclic code C, i.e. the monic polynomial of degree $n-k$ such that $\langle g(\eta) \rangle = C$. It is well-known that $g(\eta)$ divides $\eta^n - 1$ and the set $\tilde{S}_C = \{i_1, \ldots, i_{n-k} \mid g(\alpha^{i_j}) = 0,\ j = 1, \ldots, n-k\}$ is called the *complete defining set* of C. Also, the roots of unity $\{\alpha^i \mid i \in \tilde{S}_C\}$ are called the *zeros* of the cyclic code C. Notice that the complete defining set permits to specify a cyclic code. By this fact, we can write a parity-check matrix for C as an $(n-k) \times n$ matrix $H = \{h_{j\ell}\}_{j,\ell}$ over $\mathbb{F}_{q^m}$ such that $h_{j\ell} = \alpha^{\ell i_j}$, where $i_j \in \tilde{S}_C$ and $\ell = 0, \ldots, n-1$. This H is called the *standard parity-check matrix*. As the complete defining set is partitioned into cyclotomic classes, any subset of $\tilde{S}_C$ containing at least one element per cyclotomic class is sufficient to specify the code. We call such a set a *defining set* of C. We will use S_C to denote a defining set which is not necessarily a complete defining set.

C. Cooper's philosophy

In this section we describe the so-called *Cooper's philosophy* approach to decode cyclic codes up to their true error correction capability [21]. The high-level idea here is to reduce the decoding problem to that of solving a polynomial system of equations where the unknowns are the error locations and the error values.

Given an $[n, k, d]_q$ code C, we recall that the *error correction capability* of C is $t = \lfloor (d-1)/2 \rfloor$, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x. Let $\mathbf{c}, \mathbf{r}, \mathbf{e} \in (\mathbb{F}_q)^n$ be, respectively, the transmitted codeword, the received vector and the error vector, then $\mathbf{r} = \mathbf{c} + \mathbf{e}$. If we apply the standard parity-check matrix H to $\mathbf{r}$, we get $H\mathbf{r} = H(\mathbf{c} + \mathbf{e}) = H\mathbf{e} = \mathbf{s} \in (\mathbb{F}_{q^m})^{n-k}$. The vector $\mathbf{s}$ is called *syndrome vector* and its components $s_1, \ldots, s_{n-k}$ are called the *syndromes*. Recall that a *correctable syndrome vector* is a syndrome vector corresponding to an error vector $\mathbf{e}$ with Hamming weight $\mu \leq t$. If there is an error vector $\mathbf{e}$ of weight $\mu \leq t$, then we can write it as

$$\mathbf{e} = (\underbrace{0, \ldots, 0}_{l_1 - 1}, \underset{\overset{\uparrow}{l_1}}{e_{l_1}}, 0, \ldots, 0, \underset{\overset{\uparrow}{l_k}}{e_{l_k}}, 0, \ldots, 0, \underset{\overset{\uparrow}{l_\mu}}{e_{l_\mu}}, \underbrace{0, \ldots, 0}_{n-1-l_\mu}) .$$

We say that the set $L = \{l_1, \ldots, l_\mu\} \subset \{0, \ldots, n-1\}$ is the set of the *error positions*, the set $\{\alpha^l \mid l \in L\}$ is the set of the *error locations*, and $\{e_{l_1}, \ldots, e_{l_\mu}\}$ is the set of the *error values*. With this notation, the relation $H\mathbf{e} = \mathbf{s}$ becomes the well-known equations

$$s_j = \sum_{h=1}^{\mu} e_{l_h} (\alpha^{l_h})^{i_j} = \sum_{l \in L} e_l (\alpha^l)^{i_j}, \quad 1 \leq j \leq n-k . \tag{1}$$

The *classical error locator polynomial* associated to the error $\mathbf{e}$ is the polynomial $\sigma_e(z) = \prod_{l \in L}(1 - z\alpha^l)$, i.e. the polynomial having as zeros the inverses of the error locations; whereas the *plain error locator polynomial* is the polynomial $\mathbf{L}_e(z) = \prod_{l \in L}(z - \alpha^l)$, i.e. the polynomial having as zeros the error locations. Obviously, the knowledge of $\sigma_e(z)$ is equivalent to the knowledge of $\mathbf{L}_e(z)$, since one polynomial is the reciprocal of the other. It is well-known that finding $\mathbf{L}_e(z)$ or $\sigma(z)$ is the **hard** part of the decoding. Indeed, once $\mathbf{L}_e(z)$ is found, the decoding proceeds by applying the Chien search [22] to find the error locations, from which the error positions are immediately established, and concludes by determining the error values via solving an easy linear system.

Traditional decoding methods, such as those based on the Berlekamp-Massey algorithm and its numerous variations, start from the received vectors, compute the syndromes and then iteratively calculate a univariate polynomial, whose degree grows until it reaches μ (assuming $\mu \leq t$), and they have a termination condition that ensures that the last obtained polynomial is indeed $\sigma(z)$.

The so-called Cooper's philosophy takes a completely different approach since it uses multivariate polynomials, as we elaborate below. Associating variables $Z = (z_1, \dots, z_t)$ to the error locations, $X = (x_1, \dots, x_{n-k})$ to the syndromes $\{s_i\}_{1 \le i \le n-k}$, and $Y = (y_1, \dots, y_t)$ to the error values, we would like to write a system of polynomial equations, useful for decoding. The starting point are the equations (1), which can be rewritten in terms of variables Z, X and Y as

$$x_j \ = \ \sum_{h=1}^{\mu} y_h (z_h)^{i_j}, \quad 1 \le j \le n-k\,.$$

A first problem here is that obviously we do not know μ when we start correcting. To solve this problem, it is convenient to assume that the last $t - \mu$ Z variables take the value 0. This allows us to write equations

$$x_j \ = \ \sum_{h=1}^{t} y_h (z_h)^{i_j}, \quad 1 \le j \le n-k\,. \qquad (2)$$

which have at least the following common solution, which is of interest for us:

$$\begin{aligned} & x_j = s_j, && 1 \le j \le n-k \\ & y_h = e_{l_h},\ z_h = \alpha^{l_h}, && 1 \le h \le \mu, \\ & y_h = 1,\ z_h = 0, && \mu + 1 \le h \le t\,. \end{aligned} \qquad (3)$$

Remark 1: We note that, at least in the case of affine-variety codes (constructed by evaluating multivariate polynomials at the rational ponits of a variety, often a curve), when we have a variable that should correspond to a location but that it is allowed to take also a different value which cannot be a valid location, such as the z_j in our case, then this value is called a *ghost error location*.

If we want to use (2) to decode, once a vector is received we would compute the syndromes and substitue them in equations (2), which become a system of $n-k$ equations in the indeterminates Y and Z. Assuming we can solve it, we would need to identify our interesting solution (3). However, it is easy to see that this system has an infinite number of solutions and so this naive approach would not work. Instead, we aim at adding equations such that, at the same time, they are satisfied by (3) and they discard other uninteresting solutions. To do that, we observe that the syndromes lie in $\mathbb{F}_{q^m}$, that the valid error locations are powers of α, and thus are n-th roots of unity, and that the error values lie in $\mathbb{F}_q$ but are non-zeros, so that we can safely consider the following equations

$$x_j^{q^m} - x_j, \quad z_h^{n+1} - z_h, \quad y_h^{q-1} - 1, \qquad (4)$$

for $1 \le h \le t$, $1 \le j \le n-k$. Indeed, $\eta^{q^m} - \eta = 0$ is just the field equation for $\mathbb{F}_{q^m}$, $\eta^{n+1} - \eta = \eta(\eta^n - 1) = 0$ is equivalent to $\eta = 0$ or $\eta^n = 1$, and $\eta^{q-1} - 1 = \frac{\eta^q - \eta}{\eta}$ is the field equation for non-zero elements of $\mathbb{F}_q$.

There are other equations that we can safely add, but their justification is more involved and can be found in [12]). These equations are

$$z_h \cdot z_{h'} \cdot p_{h,h'}, \qquad (5)$$

where

$$p_{h,h'} = (z_h^n - z_{h'}^n)/(z_h - z_{h'}), \quad 1 \le h < h' \le t,$$

and they guarantee that the locations (if not zero) are all distinct. We can finally consider the system obtained by putting together (2), (4), (5). This system can be used to unambigously decode, by evaluating the X variables at the syndromes and obtaining our solution (3), plus its permutations (the system is obviously invariant by any permutation of the Z, provided we apply the same permutation to the Y), or similar solutions: any solution will be sufficient to decode.

However, this decoding relies on solving a system *every time* a vector is received and so its complexity is difficult to estimate, although experimentally it is very high. The approach presented in [12] is more radical. Orsini and Sala consider the ideal generated by the (2), (4), (5) (without evaluating any syndrome) and use advanced commutative algebra to prove the existence of a special polynomial, called *general error locator polynomial*, for every cyclic code (with $(q, n) = 1$).

In more details, a general error locator polynomial $\mathcal{L}$ for an $[n, k, d]_q$ cyclic code C is a polynomial in $\mathbb{F}_q[X, z]$, with $X = (x_1, \dots, x_{n-k})$ such that

- $\mathcal{L}(X, z) \ = \ z^t + a_{t-1}(X) z^{t-1} + \cdots + a_0(X)$, with $a_j \in \mathbb{F}_q[X]$, $0 \le j \le t-1$;
- given a correctable syndrome $\mathbf{s} = (s_1, \dots, s_{n-k})$, if we evaluate the X variables at $\mathbf{s}$, then the t roots of $\mathcal{L}(\mathbf{s}, z)$ are the μ error locations plus zero counted with multiplicity $t - \mu$.

Remark 2: The above second property is equivalent to $\mathcal{L}(\mathbf{s}, z) = z^{t-\mu} \mathbf{L_e}(z)$, where $\mathbf{e}$ is the error associated to syndrome $\mathbf{s}$.

Note that the general error locator polynomial $\mathcal{L}$ does not depend on the errors actually occurred, but it is computed in a preprocessing fashion once and for all and depends only on the code itself. As a consequence, the decoding algorithm proposed in [12], which needs $\mathcal{L}$, performs the following steps:

- Compute the syndrome vector $\mathbf{s}$ corresponding to the received vector $\mathbf{r}$;
- Evaluate $\mathcal{L}$ at the syndromes $\mathbf{s}$;
- Apply the Chien search on $\mathcal{L}(\mathbf{s}, z)$ to compute the error locations $\{\alpha^l \mid l \in L\}$;
- Deduce the error positions L from the error locations.
- Compute the error values $\{e_l \mid l \in L\}$.

This approach is efficient as long as the evaluation of $\mathcal{L}$ is efficient (see Section VII).

D. General error locator polynomials for some binary cyclic codes

Here we recall some techniques used in [13] to efficiently compute a general error locator polynomial for binary cyclic codes without using Gröbner bases. In this section we only deal with binary cyclic codes and we will often shorten "binary cyclic (linear) code" to "code" when it is clear from the context.

If we want to compute the general error locators for a range of codes (such that for example $t = 2$ and $n < 63$ as in [13]), our first problem is to reduce the cases we must consider. The following theorem shows that there are two facts in our help. The former is that if we can decode a code then we can decode any of its equivalent codes. The latter is that if code contains a subcode with the same correction capability, then we can use the general locator of the larger code to correct also the smaller code.

Theorem 3 ([13]): Let C,C' and C'' be three codes with the same length and the same correction capability. Let $\mathcal{L}_C$,$\mathcal{L}_{C'}$ and $\mathcal{L}_{C''}$ denote their respective general error locator polynomials.

If C is a subcode of C', then we can assume $\mathcal{L}_C = \mathcal{L}_{C'}$.

If C is equivalent to C'' via the coordinate permutation function $\phi : (\mathbb{F}_2)^n \longrightarrow (\mathbb{F}_2)^n$, then we can decode C using $\mathcal{L}_{C''}$ (via ϕ).

So the first thing to do is to group all codes under consideration in sets such that the computation of one locator per set would allow the decoding of all codes. Then we need to identify in each set a specific code for which the computation of the general locator is realively easier. For example, in [13] we presented a classification result where all binary cyclic codes up with $t = 2$ and $7 \leq n < 63$, which are 952 in total, would fall in five classes, plus their equivalent codes and subcodes (with the same t). Each of the first four classes contains an infinite number of codes (considering all possible lengths), while the fifth is just a list of five given codes. For each code of the fifth class, a computer computation provided a general locator. For three of the other classes, an explicit representation of the general locator was given, while for the remaining class an implicit representation was given. The implicit representation allows in practice an efficient evaluation of the general locator, but it is theoretically unpleasant, because an explicit sparse representation would be preferred and makes formal complexity estimations easier.

Given a class, the method followed to identify a better code started from two observations. The first observation is that if $1 \in S_C$, then its corresponding syndrome x_1 satisfies the relation $x_1 = z_1 + z_2$. The second observation requires more explanation, provided below. Let C be a code with error capability $t = 2$, $\mathbf{s}$ a correctable syndrome, and $\hat{z_1}$ and $\hat{z_2}$ the (possibly ghost) error locations corresponding to the syndrome $\mathbf{s}$. Then, by definition we know that

$$\mathcal{L}(X, z) = z^2 + az + b = (z - \hat{z_1})(z - \hat{z_2}),$$

where $a, b \in \mathbb{F}_2[X]$, and $b(\mathbf{s}) = \hat{z_1}\hat{z_2}$, $a(\mathbf{s}) = \hat{z_1} + \hat{z_2}$. Moreover, there are exactly two errors if and only if $b(\mathbf{s}) \neq 0$, and there is exactly one error if and only if $b(\mathbf{s}) = 0$ and $a(\mathbf{s}) \neq 0$ (in this case the error location is $a(\mathbf{s})$). Therefore, if $1 \in S_C$, we can write $a = x_1$ and so only b needs to be found, that is,

$$\mathcal{L}(X, z) = z^2 + x_1 z + b, \qquad \text{if } 1 \in S_C$$

Which means that if we can find a class representative with $1 \in S_C$, it is preferrable to use it to derive the general locator. Fortunately, in [13] it was proved that given a binary cyclic codes C_1 with $t = 2$ and $7 \leq n < 105$, then C_1 is equivalent to a code C_2 with $1 \in S_{C_2}$. Therefore, since when we search for general locators we proceed modulo code equivalence, we could always assume that $1 \in S_C$.

Definition 4: We denote by $\mathcal{V}^\mu$ the set of syndrome vectors corresponding to μ errors, with $0 \leq \mu \leq t$. The set of *correctable syndromes* $\mathcal{V}$ is given by the (disjoint) union of sets $\mathcal{V}^\mu$ for $\mu = 0, \ldots, t$ (corresponding to 0, ..., t errors, respectively), i.e. $\mathcal{V} = \mathcal{V}^0 \sqcup \mathcal{V}^1 \sqcup \cdots \sqcup \mathcal{V}^t$.

Remark 5: When the defining set is not complete, the general error locator polynomial will not contain $n - k$ X variables, but a smaller number. If the defining set is given as small as possible, then there is only one syndrome per cyclotomic class and we call such syndromes **primary syndromes**. Although we can keep only the primary syndromes to build a locator, sometimes it is convenient to keep also other syndromes in order to arrive at a general formula for a code class with infinite members. Given a specific code, if desired, we can trivially convert our polynomial into a polynomial with only primary syndromes as variables.

From now on, we reserve the letter r to denote the number of syndromes we are actually working on and so r will be at least the number of primary syndromes and at most $n - k$. In particular, we will assume $\mathcal{V} \subset (\mathbb{F}_{q^m})^r$ and $X = (x_1, \ldots, x_r)$.
1

III. A CONJECTURE AND ITS FIRST LINK TO THE COMPLEXITY OF DECODING CYCLIC CODES

In this section we present a conjecture on the sparsity of general error locator polynomials along with an estimate of the complexity of the decoding approach presented in Section II for any cyclic code. In order to do that, first we need to provide a rigorous notion of *sparsity* that is appropriate for these polynomials.

Definition 6: Let $\mathbb{K}$ be any field and let f be any (possibly multivariate) polynomial with coefficients in $\mathbb{K}$, that is, $f \in \mathbb{K}[\mathsf{a}_1, \dots, \mathsf{a}_N]$ for a variable set $A = \{\mathsf{a}_1, \dots, \mathsf{a}_N\}$. We will denote by $|f|$ the number of terms (monomials) of f.

Definition 7: Let $A = \{\mathsf{a}_1, \dots, \mathsf{a}_N\}$ and $B = \{\mathsf{b}_1, \dots, \mathsf{b}_M\}$ be two variable sets. Let $\mathbb{K}$ be a field and let $\mathcal{F}$ be a rational function in $\mathbb{K}(A)$. Let $F \in \mathbb{K}[B]$, $f_1, \dots, f_M \in \mathbb{K}[A]$ and $g_1, \dots, g_M \in \mathbb{K}[A]$. We say that the triple $(F, \{f_1 \dots, f_M\}, \{g_1 \dots, g_M\})$ is a **rational representation** of $\mathcal{F}$ if

$$\mathcal{F} = F(f_1/g_1, \dots, f_M/g_M)\,.$$

We say that the number

$$|F| + \sum_{i=1}^{M}(|f_i| - 1) + \sum_{j=1, g_j \notin \mathbb{K}}^{M} |g_j|$$

is the **rational density** of the rational representation $(F, \{f_1 \dots, f_M\}, \{g_1 \dots, g_M\})$.

Given a rational function $\mathcal{F}$, the concept behind the rational density of a rational representation is to measure the complexity of *evaluating it* when we think to have a clever way to write it. For example, if we have $\mathcal{F} \in \mathbb{F}_2[x, y]$,

$$\mathcal{F} = \frac{x^7 + x^6y + x^5y^2 + x^4y^3 + x^3y^4 + x^2y^5 + xy^6 + y^7}{x^5 + x^4y^2 + xy^8 + y^{10}}\,,$$

one might observe that $\mathcal{F} = \frac{(x+y)^7}{(x+y^2)^5}$, so it is convenient first to evaluate $f_1 = x + y$ and $g_1 = (x + y^2)$, and then to compute $F\left(\frac{f_1}{g_1}\right) = \frac{f_1{}^7}{g_1{}^5}$. In other words, f_1 and g_1 represents intermediate evaluations that are convenient to perform as first steps, while $F = \frac{b_1{}^7}{b_2{}^5}$ is the last evaluation, that uses the intermediate evaluations obtained previously to compute the (global) evaluation of $\mathcal{F}$.

However, there could always be a more clever way to write a function, in order to minimize the monomials we must compute. To provide an unambiguous value, albeit very difficult to determine precisely, we propose the following defintion.

Definition 8: Let $A = \{\mathsf{a}_1, \dots, \mathsf{a}_N\}$ and let $\mathcal{F}$ be a rational function in $\mathbb{K}(A)$. Then, we define the **functional density** of $\mathcal{F}$, $||\mathcal{F}||$, as the minimum among the rational densities of all rational representations of $\mathcal{F}$.

With the notation of Definition 6 and 8, we have the following result, that shows their interlink and how natural Definition 8 is.

Theorem 9: Let $A = \{\mathsf{a}_1, \dots, \mathsf{a}_N\}$. If $\mathcal{F}$ is a polynomial, i.e. $\mathcal{F} \in \mathbb{K}[A]$, then

$$||\mathcal{F}|| \le |\mathcal{F}|\,.$$

Moreover, if $\mathcal{F} = \mathsf{a}_1 + \mathsf{a}_2$ then $||\mathcal{F}|| = |\mathcal{F}| = 2$.

Proof: Let $\mathcal{F} \in \mathbb{K}[A]$ and let $\rho = |\mathcal{F}|$. Then $\mathcal{F} = \sum_{i=1}^{\rho} h_i$, where any h_i is a monomial for $1 \le i \le \rho$. Let us consider the following rational representation for $\mathcal{F}$

$$B = \{\mathsf{b}_1, \dots, \mathsf{b}_\rho\}, \quad F = \sum_{i=1}^{\rho} \mathsf{b}_i\,, \quad f_i = h_i\,, \quad g_i = 1,,$$

where $1 \le i \le \rho$, then the rational density of

$$(F, \{f_1, \dots, f_\rho\}, \{g_1, \dots, g_\rho\})$$

is

$$|F| + \sum_{i=1}^{\rho}(|f_i| - 1) + \sum_{j=1, g_j \notin \mathbb{K}}^{\rho} |g_j| \quad = \rho + 0 + 0 = \rho\,,$$

which implies $||\mathcal{F}|| \le \rho$, as claimed. To prove the case $\mathcal{F} = \mathsf{a}_1 + \mathsf{a}_2$, we argument by contradiction assuming $||\mathcal{F}|| = 1$. Let us consider any rational representation of $\mathcal{F}$ providing

$$||\mathcal{F}|| \quad = \quad |F| + \sum_{i=1}^{M}(|f_i| - 1) + \sum_{j=1, g_j \notin \mathbb{K}}^{M} |g_j| \quad = \; 1\,.$$

Since $|F| \ge 1$, we must have $|F| = 1$, $\sum_{i=1}^{M}(|f_i| - 1) = 0$ and $\sum_{j=1, g_j \notin \mathbb{K}}^{M} |g_j| = 0$.

Therefore, $M = 1$, $|f_1| = 1$ and $g_1 = \nu \in \mathbb{K}$. From $M = 1$ and $|F| = 1$ we have $F = \lambda \mathsf{b}_1^{\mu}$ for $\lambda \in \mathbb{K} \setminus \{0\}$ and $\mu \ge 1$, and so $\mathcal{F} = F(f_1/g_1) = \lambda\left(\frac{f_1}{\nu}\right)^{\mu}$. Recalling that $\mathcal{F} = \mathsf{a}_1 + \mathsf{a}_2$, we finally have a contradiction

$$|f_1| = 1 \implies \left|\lambda\left(\frac{f_1}{\nu}\right)^{\mu}\right| = 1\,, \quad \text{but} \quad |\mathcal{F}| = |\mathsf{a}_1 + \mathsf{a}_2| = 2\,.$$

■

For example, let us consider a possible locator $\mathbb{F}_2[z, x_1, x_2, x_3, x_4, x_5]$ for a code with $t = 2$, with b of the form:

$$b = \left(\frac{x_2x_4 + x_5}{x_3}\right)^{l^+}, \quad \mathcal{L} = z^2 + x_1z + b,$$

where $1 \le l^+ \le n$. Its functional density satisfies $||\mathcal{L}|| \le 6$ thanks to the following rational representation

$$\mathcal{L} = F(f_1/g_1, f_2/g_2, f_3/g_3),$$

where $F \in \mathbb{F}_2[\mathsf{b}_1, \mathsf{b}_2, \mathsf{b}_3]$, $f_1, f_2, f_3, g_1, g_2, g_3 \in \mathbb{F}_2[z, x_1, x_2, x_3, x_4, x_5]$ and

$$F = b_1^2 + b_1b_2 + b_3^{l^+}, \quad f_1 = z, g_1 = 1,$$
$$f_2 = x_1, g_2 = 1, \quad f_3 = x_2x_4 + x_5, g_3 = x_3\,.$$

Remark 10: The apparently-trivial result $||\mathsf{a}_1 + \mathsf{a}_2|| = |\mathsf{a}_1 + \mathsf{a}_2|$ in Theorem 9 is essential to show that our notion of functional density is meaningful. Indeed, it is straightforward to provide easier alternative definitions enjoying $||F|| \le |F|$ for any F but which will give $||\mathsf{a}_1 + \mathsf{a}_2|| = 1$.

Now that we have provided a rigorous notion, we can finally write Sala's conjecture, which was presented orally at the conference MEGA2005 in 2005 together with the first experiments in the computation of general locators.

Conjecture 11 (Sala, MEGA2005):
Let $p \geq 2$ be a prime, $m \geq 1$ a positive integer and let $q = p^m$. There is an integer $\epsilon = \epsilon(q)$ such that, for any cyclic code C over the field $\mathbb{F}_q$ with $n \geq q^4 - 1$, $\gcd(n, q) = 1$, $3 \leq d \leq n - 1$,
C admits a general error locator polynomial $\mathcal{L}_c$ whose functional density is bounded by

$$||\mathcal{L}_c|| \leq n^{\epsilon} .$$

Moreover, for binary codes we have $\epsilon = 3$, that is, $\epsilon(2) = 3$.

Remark 12: There are some indications that $\epsilon(q)$ may grow with q and so we do not believe in just one ϵ for all finite fields. For example, we will see that properties of the univariate interpolation in Theorem 37 suggest denser locators for larger fields.
Thanks to this conjecture, we can now formally define a *sparse locator*, as follows.
Let C be a cyclic code over $\mathbb{F}_q$ of length n. Let d be its distance, t its correction capability and $S_C = \{i_1, \ldots, i_r\}$ a defining set of C. Let $\mathcal{L}_C$ be a general error locator polynomial of C.

Definition 13: If $\mathcal{L}_C \in \mathbb{F}_2[x_1, \ldots, x_r]$, then we say that $\mathcal{L}_C$ is **sparse** if $||\mathcal{L}_C|| \leq n^3$.
If Conjecture 11 holds and $\mathcal{L}_C \in \mathbb{F}_q[x_1, \ldots, x_r]$, then we say that $\mathcal{L}_C$ is **sparse** if $||\mathcal{L}_C|| \leq n^{\epsilon}$.
The decoding procedure developed by Orsini and Sala in [13] consists of five steps:

1) Computation of the r syndromes $s_1, \ldots, s_r$ corresponding to the received vector.
2) Evaluation of $\mathcal{L}_C(x_1, \ldots, x_r, z)$ at $\mathbf{s} = (s_1, \ldots, s_r)$.
3) Computation of the roots of $\mathcal{L}_C(\mathbf{s}, z)$, which are the valid locations and the ghost locations. The number of valid locations immediately gives μ.
4) Computations of the error positions from the (valid) locations.
5) Computation of the error values $e_{l_1}, \ldots, e_{l_\mu}$.

By analyzing the above decoding algorithm, we observe that the main computational cost is the evaluation of the polynomial $\mathcal{L}_C(x_1, \ldots, x_r, z)$ at $\mathbf{s}$, which reduces to the evaluation of its z-coefficients. Indeed, the computation of the r syndromes $s_1, \ldots, s_r$ and of the roots of $\mathcal{L}_C(\mathbf{s}, z)$ cost, respectively, $O(t\sqrt{n})$ and $\max(O(t\sqrt{n}), O(t\log(\log(t))\log(n)))$ ([23]), while the computation of the error values using Forney's algorithm costs $O(t^2)$ ([24]). Therefore, we can bound the total cost of steps 1, 3 and 4 with $O(n^2)$.
The following theorem is then clear and should be compared with the results in [25], which suggest that for linear codes an extension of Conjecture 11 is very unlikely to hold.

Theorem 14: Let us consider all cyclic codes over the same field $\mathbb{F}_q$ with $\gcd(n, q) = 1$ and $d \geq 3$. If Conjecture 11 holds, they can be decoded in polynomial time in n, once a preprocessing has produced sparse general error locator polynomials.

Proof: The only special situations not tackled by Conjecture 11 are the finite cases when $n < q^4 - 1$, which of course do not influence the asymptotic complexity, and the degenerate case when $d = n$, which can be decoded in polynomial time without using the general error locator algorithm. ■

Remark 15: In assessing the plausibility of Conjecture 11 it is important to keep in mind that Sala claimed the existence of *at least one sparse locator per code*. It is possible that for a specific code only one such locator exists and that its computation is extremely difficult. This is the reason why Theorem 14 allows for unbounded preprocessing time. However, readers familiar with results in [25] will know that for linear codes the decoding problem remains difficult *even* allowing for unbounded preprocessing time and so they will appreciate the potential impact of Theorem 14.

Although all reported experiments (especially in the binary case) confirm Conjecture 11, we are far from having a formal proof of it. In the next sections we will give theoretical and experimental results that provide some evidence to Conjecture 11. In Section VII we will discuss complexity issues more in depth and we will make comparison with previous results by other authors.

IV. A GENERAL DESCRIPTION FOR THE LOCATOR POLYNOMIAL

In this section we give a new general result on the structure of the error locator polynomial for *all* cyclic codes over $\mathbb{F}_q$.

Let $R_n = \{\alpha^i \mid i = 0, \ldots, n-1\}$. Let us denote with $T_{n,t}$ the following set (compare with [8, p. 131] and Def. 13 of [13])

$$\begin{aligned} T_{n,t} = \; & \{(\alpha^{l_1}, \ldots, \alpha^{l_\mu}, 0, \ldots, 0) \mid 0 \leq l_1 < \cdots < l_\mu < n, \\ & \qquad 0 \leq \mu \leq t\} \subset (R_n \cup \{0\})^t . \end{aligned}$$

Let C be a cyclic code over $\mathbb{F}_q$, with length n and correction capability t, defined by $S_C = \{i_1, \ldots, i_r\}$ and let x_j be the syndrome corresponding to i_j for $j \in \{1, \ldots r\}$. The following theorem (Theorem 19) generalizes Theorem 1 of [8], which dealt with the case where the code could be defined by only one syndrome. Here we provide a description of the shape of the coefficients of a general error locator polynomial for cyclic codes over $\mathbb{F}_q$. We recall that these coefficients are polynomials in the syndrome variables X. When they are evaluated at a correctable syndrome, corresponding to an error of weight $\mu \leq t$, they can be expressed as the elementary symmetric functions on the roots of the general error locator polynomials, which are exactly the error locations $z_1, \dots, z_\mu$ and zero (with multiplicity $t - \mu$). By definition of elementary symmetric functions, they can then be expressed as elementary symmetric polynomials in μ variables on the $z_1, \dots, z_\mu$. We will need the existence of a polynomial representation for arbitrary functions from $(\mathbb{F}_q)^n$ to $\mathbb{F}_q$. This is not unique and can be obtained in several ways, including multivariate interpolation ([9]). We report a standard formulation in the following lemma.

Lemma 16 ([26, p. 26]): Let $f : (\mathbb{F}_q)^n \to \mathbb{F}_q$. Then f can be represented by a polynomial in $\mathbb{F}_q[x_1, \dots, x_n]$, i.e. there is a polynomial $P \in \mathbb{F}_q[x_1, \dots, x_n]$ such that $P(b_1, \dots, b_n) = f(b_1, \dots, b_n)$ for all $(b_1, \dots, b_n) \in (\mathbb{F}_q)^n$. In particular, the polynomial

$$\sum_{\mathbf{a} \in \mathbb{F}_q^n} f(a_1, \dots, a_n)[1 - (x_1 - a_1)^{q-1}] \cdots [1 - (x_n - a_n)^{q-1}],$$

where $\mathbf{a} = (a_1, \dots, a_n)$, represents f.

The next two lemmas clarify some links between syndromes and error locations which will be essential in our proof of Theorem 19.

Lemma 17: Let $\sigma \in \mathbb{F}_q[y_1, \dots, y_t]$ be a symmetric function. Then there exists $a \in \mathbb{F}_q[X]$ such that for $(\bar{x}_1, \dots, \bar{x}_r) \in \mathcal{V}^\mu$

$$a(\bar{x}_1, \dots, \bar{x}_r) = \sigma(z_1, \dots, z_\mu, 0 \dots, 0)$$

with $z_1, \dots z_\mu$ the error locations corresponding to $\bar{x}_1 \dots, \bar{x}_r$.

Proof: We claim that the statement is obvious for elementary symmetric functions, as follows. Let $\sigma_1 \dots \sigma_t$ be the elementary symmetric functions in $\mathbb{F}_q[y_1, \dots, y_t]$. The existence of a general error locator polynomial for any cyclic code guarantees that, for any $1 \le i \le t$, for any σ_i there is $a_i \in \mathbb{F}_q[x_1, \dots, x_r]$ such that for $(\bar{x}_1, \dots, \bar{x}_r) \in \mathcal{V}^\mu$

$$a_i(\bar{x}_1, \dots, \bar{x}_r) = \sigma_i(z_1, \dots, z_\mu, 0 \dots, 0)$$

with $z_1, \dots z_\mu$ the error locations corresponding to $\bar{x}_1 \dots, \bar{x}_r$.

For the more general case of any symmetric function $\sigma \in \mathbb{F}_q[y_1, \dots, y_t]$, we need the fundamental theorem on symmetric functions, which shows the existence of a polynomial $H \in \mathbb{F}_q[y_1 \dots, y_t]$ such that $\sigma(y_1, \dots, y_t) = H(\sigma_1(y_1, \dots, y_t), \dots, \sigma_t(y_1 \dots, y_t))$. We can define $a = H(a_1, \dots, a_t) \in \mathbb{F}_q[X]$. So for $(\bar{x}_1, \dots, \bar{x}_r) \in \mathcal{V}^\mu$ and the corresponding locations $z_1, \dots z_\mu$, we have

$$\begin{aligned}
&\sigma(z_1, \dots, z_\mu, 0, \dots, 0) = \\
&= H(\sigma_1(z_1, \dots, z_\mu, 0, \dots, 0), \dots, \sigma_t(z_1, \dots, z_\mu, 0, \dots, 0)) \\
&= H(a_1(\bar{x}_1, \dots, \bar{x}_r), \dots, a_t(\bar{x}_1 \dots, \bar{x}_r)) = \\
&= a(\bar{x}_1, \dots, \bar{x}_r) .
\end{aligned}$$

■

Lemma 18: Let $h \in \mathbb{F}_q[X]$ with $\deg_{x_i} h < q$ for all $i = 1, \dots, r$, and $h(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_r) = 0$ for all $(\bar{x}_1, \dots, \bar{x}_r) \in (\mathbb{F}_q)^r$ with $\bar{x}_1 \neq 0$. Let $l \in \mathbb{F}_q[X]$ and $g(x_2 \dots, x_r) \in \mathbb{F}_q[x_2, \dots, x_r]$ such that

$$h(X) = x_1 l(x_1, x_2, \dots, x_r) + g(x_2, \dots, x_r) .$$

Then

$$h = 0 \quad \text{or} \quad h = (1 - x_1^{q-1}) \cdot g(x_2, \dots, x_r).$$

Proof: Clearly, for any h the two polynomials l and g are uniquely determined.

If $h(X) \in \mathbb{F}_q[x_2 \dots, x_r]$, trivially, we have that $h(X) = g(x_2, \dots, x_r)$. But then $h = 0$ since the value of h does not depend on $\bar{x}_1$ and h must vanish any time that $\bar{x}_1 \neq 0$.

So we can suppose that $h(X) \notin \mathbb{F}_q[x_2 \dots, x_r]$ and we can define a polynomial $\bar{h}(X) = (1 - x_1^{q-1}) \cdot g(x_2, \dots, x_r)$. Note that $\deg_{x_i} \bar{h} < q$ for all $i = 1, \dots, r$. We claim that $h = \bar{h}$. Since the degree w.r.t. each variable x_i of both the polynomials h and $\bar{h}$ is less than q, to prove our claim it is sufficient to show that $h(\hat{\mathbf{X}}) = \bar{h}(\hat{\mathbf{X}})$ for all $\hat{\mathbf{X}} \in (\mathbb{F}_q)^r$. Let us distinguish the cases $\hat{x}_1 = 0$ and $\hat{x}_1 \neq 0$.

If $\hat{x}_1 = 0$ then $h(\hat{X}) = 0 \cdot l(0, \hat{x}_2, \dots, \hat{x}_r) + g(\hat{x}_2, \dots, \hat{x}_r) = g(\hat{x}_2, \dots, \hat{x}_r)$ and $\bar{h}(\hat{X}) = (1 - 0) \cdot g(\hat{x}_2, \dots, \hat{x}_r) = g(\hat{x}_2, \dots, \hat{x}_r)$. So, in this case $h(\hat{X}) = \bar{h}(\hat{X})$.

Otherwise, let $\hat{x}_1 \neq 0$. By hypothesis, $h(\hat{X}) = 0$. On the other hand, $\bar{h}(\hat{X}) = (1 - 1) \cdot g(\hat{x}_2, \dots, \hat{x}_r) = 0$. So, also in this case $h(\hat{X}) = \bar{h}(\hat{X})$. ■

Theorem 19: Let C be a cyclic code over $\mathbb{F}_q$, with length n and correction capability t, defined by $S_C = \{i_1, \dots, i_r\}$ and let x_j be the syndrome corresponding to i_j for $j \in \{1, \dots r\}$. Let $\sigma \in \mathbb{F}_q[y_1, \dots, y_t]$ be a symmetric homogeneous function of total degree δ, with δ a multiple of i_1, and let λ be a divisor of n. Then there exist polynomials $a \in \mathbb{F}_q[X]$, $g \in \mathbb{F}_q[x_2, \dots, x_r]$, some non-negative integers $\delta_2, \dots, \delta_r$ and some univariate polynomials $F_{h_2,\dots,h_r} \in \mathbb{F}_q[y]$ such that for any $0 \le \mu \le t$, for any $(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_r) \in \mathcal{V}^\mu$ and the corresponding error locations $z_1, \dots, z_\mu$, we have

$$a(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_r) = \sigma(z_1, \dots, z_\mu, 0, \dots, 0) \qquad (6)$$

and

$$\begin{aligned}
a(X) = x_1^{\delta/i_1} \cdot & \qquad (7) \\
\cdot \sum_{h_r=0}^{\delta_r} \Biggl(\left(\frac{x_r}{x_1^{i_r}} \right)^{h_r} & \cdots \sum_{h_2=0}^{\delta_2} \left(\left(\frac{x_2}{x_1^{i_2}} \right)^{h_2} F_{h_2,\dots,h_r}(x_1^\lambda) \right) \cdots \Biggr) + \\
+ (1 - x_1^{q^m - 1}) \cdot g(x_2, \dots, x_r), &
\end{aligned}$$

where $\delta_i \le q^m - 1$, $\deg F_{h_2,\dots,h_r} \le (q^m - 1)/\lambda$.

Proof: We observe that (6) is immediate by Lemma 17. To prove (7) we first show the case $\bar{x}_1 \neq 0$ and then the general case.

Case $\bar{x}_1 \neq 0$

Let us consider the following map $A : \{X \in \mathcal{V} \mid x_1 \neq 0\} \longrightarrow \mathbb{F}_{q^m}$ defined by

$$A(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_r) = \frac{\sigma(z_1, \ldots, z_\mu, 0, \ldots, 0)}{\bar{x}_1^{\delta/i_1}}, \qquad (8)$$

where $(z_1, \ldots, z_\mu, 0, \ldots, 0)$ is the element of $T_{n,t}$ associated to the syndrome vector $(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_r)$. We claim that A depends only on $(\bar{x}_1^\lambda, \bar{x}_2/\bar{x}_1^{i_2}, \ldots, \bar{x}_r/\bar{x}_1^{i_r})$. If our claim is true, then we have that

$$A(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_r) = f(\bar{x}_1^\lambda, \bar{x}_2/\bar{x}_1^{i_2}, \ldots, \bar{x}_r/\bar{x}_1^{i_r}) \qquad (9)$$

for a function $f : (\mathbb{F}_{q^m})^r \longrightarrow \mathbb{F}_{q^m}$, and so, by Lemma 16, we can view f as a polynomial in $\mathbb{F}_{q^m}[x_1, \ldots, x_r]$. Since $\mathcal{V} \subset (\mathbb{F}_{q^m})^r$, we can also view A as a (non-unique) polynomial $A(X) \in \mathbb{F}_{q^m}[X]$. On the other hand, (8) and Lemma 17 show that $A(X)x_1^{\delta/i_1} \in \mathbb{F}_{q^m}[X]$ equals a polynomial $a \in \mathbb{F}_q[X]$ and so also $A(X)$ can be chosen in $\mathbb{F}_q[X]$. Therefore, by (9) also f can be chosen in $\mathbb{F}_q[X]$.

Let $\delta_2 = \deg_{x_2}(f), \ldots, \delta_r = \deg_{x_r}(f)$. Then, by collecting the powers of x_r in f, we will have $f = \sum_{h=0}^{\delta_r} x_r^h f_h$, for some f_h's, which are polynomials in $\mathbb{F}_q[x_1, \ldots, x_{r-1}]$. We observe that for any $2 \leq i \leq r-1$ we have that, for any $0 \leq h \leq \delta_r$, $\deg_{x_i}(f_h) \leq \deg_{x_i}(f) = \delta_i$ and there is at least one h such that $\deg_{x_i}(f_{\mathsf{h}}) = \delta_i$. Note that, since we are interested in the values of f only at the points of $\mathbb{F}_{q^m}$, we can assume $\delta_i \leq q^m - 1$ due to the field equation. We can repeat this argument on all f_h's by collecting powers of x_{r-1} and iterate on the other X variables, x_1 excluded, until we obviously obtain the following formal description

$$\begin{aligned} f(x_1, x_2, \ldots, x_r) \quad &= \qquad (10) \\ = \sum_{h_r=0}^{\delta_r} x_r^{h_r} \sum_{h_{r-1}=0}^{\delta_{r-1}} x_{r-1}^{h_{r-1}} \cdots &\sum_{h_2=0}^{\delta_2} x_2^{h_2} F_{h_2,\ldots,h_r}(x_1), \end{aligned}$$

where any $F_{h_2,\ldots,h_r}$ is a univariate polynomial in $\mathbb{F}_q[x_1]$.

From (8), (9) and (10) we directly obtain the restriction of (7) to the case $x_1 \neq 0$, considering that $x_j^{q^m} = x_j$ implies $(1 - x_j^{q^m-1}) = 0$, $\delta_i \leq q^m - 1$ and $\deg F_{h_1,\ldots h_r} \leq (q^m - 1)/\lambda$.

We now prove our claim that gives (9). Let us take $(\tilde{x}_1, \ldots, \tilde{x}_r)$ and $(\bar{x}_1, \ldots, \bar{x}_r)$ such that $\tilde{x}_1^\lambda = \bar{x}_1^\lambda$, and $\tilde{x}_j/\tilde{x}_1^{i_j} = \bar{x}_j/\bar{x}_1^{i_j}$, for $j = 2, \ldots, r$.

The first relation implies

$$\tilde{x}_1 = \beta\bar{x}_1, \qquad (11)$$

for some β such that $\beta^\lambda = 1$.

Substituting $\tilde{x}_1$ for $\beta\bar{x}_1$ in the second relation for $j = 2, \ldots, r$, we obtain

$$\frac{\tilde{x}_j}{\tilde{x}_1^{i_j}} = \frac{\bar{x}_j}{\bar{x}_1^{i_j}} \implies \frac{\tilde{x}_j}{(\beta\bar{x}_1)^{i_j}} = \frac{\bar{x}_j}{\bar{x}_1^{i_j}} \implies \tilde{x}_j = \beta^{i_j}\bar{x}_j. \qquad (12)$$

Suppose that $(\tilde{x}_1, \ldots, \tilde{x}_r) \in \mathcal{V}^\mu$ and $(\bar{x}_1, \ldots, \bar{x}_r) \in \mathcal{V}^{\mu'}$, with $\mu, \mu' \leq t$. From (11), we get $\tilde{y}_1\tilde{z}_1^{i_1} + \cdots + \tilde{y}_\mu\tilde{z}_\mu^{i_1} = \beta(\bar{y}_1\bar{z}_1^{i_1} + \cdots + \bar{y}_{\mu'}\bar{z}_{\mu'}^{i_1}) = \beta\bar{y}_1\bar{z}_1^{i_1} + \cdots + \beta\bar{y}_{\mu'}\bar{z}_{\mu'}^{i_1}$, where $\bar{z}_i$'s and $\bar{y}_i$'s are the locations and the error values, respectively, associated to $(\bar{x}_1, \ldots, \bar{x}_r)$; and similarly for $\tilde{z}_i$'s and $\tilde{y}_i$'s. Also, from (12) we get $\tilde{y}_1\tilde{z}_1^{i_j} + \cdots + \tilde{y}_\mu\tilde{z}_\mu^{i_j} = \beta^{i_j}(\bar{y}_1\bar{z}_1^{i_j} + \cdots + \bar{y}_{\mu'}\bar{z}_{\mu'}^{i_j})$, for $j = 2, \ldots, r$.

Let us now take $\hat{y}_j = \bar{y}_j$ and $\hat{z}_j = \beta\bar{z}_j$, for $j = 1, \ldots, \mu'$. Since the $\hat{z}_j$ are distinct valid error locations (i.e. $\hat{z}_j^n = 1$, for $j = 1, \ldots, \mu'$) we have that their syndromes are

$$\begin{aligned} \hat{x}_j &= \hat{y}_1\hat{z}_1^{i_j} + \cdots + \hat{y}_{\mu'}\hat{z}_{\mu'}^{i_j} = \\ &= \beta^{i_j}\bar{y}_1\bar{z}_1^{i_j} + \cdots + \beta^{i_j}\bar{y}_{\mu'}\bar{z}_{\mu'}^{i_j} = \\ &= \beta^{i_j}(\bar{y}_1\bar{z}_1^{i_j} + \cdots + \bar{y}_{\mu'}\bar{z}_{\mu'}^{i_j}) = \\ &= \tilde{y}_1\tilde{z}_1^{i_j} + \cdots + \tilde{y}_\mu\tilde{z}_\mu^{i_j} = \tilde{x}_j, \quad \text{for j = 1, \ldots, r}. \end{aligned}$$

Hence $(\hat{x}_1, \ldots, \hat{x}_r) = (\tilde{x}_1, \ldots, \tilde{x}_r)$, which implies that their corresponding locations and values must be the same and unique, because $\mu, \mu' \leq t$. Therefore $\mu = \mu'$, $\{\tilde{z}_1, \ldots, \tilde{z}_\mu\} = \{\beta\bar{z}_1, \ldots, \beta\bar{z}_\mu\}$, and $\{\tilde{y}_1, \ldots, \tilde{y}_\mu\} = \{\bar{y}_1, \ldots, \bar{y}_\mu\}$, from which, using the fact that σ is a symmetric homogeneous function of degree δ, we have

$$\begin{aligned} A(\tilde{x}_1, \ldots, \tilde{x}_r) &= \frac{\sigma(\tilde{z}_1, \ldots, \tilde{z}_\mu, 0, \ldots, 0)}{(\tilde{y}_1\tilde{z}_1^{i_1} + \cdots + \tilde{y}_\mu\tilde{z}_\mu^{i_1})^{\delta/i_1}} = \\ &\frac{\sigma(\beta\bar{z}_1, \ldots, \beta\bar{z}_\mu, 0, \ldots, 0)}{(\bar{y}_1(\beta\bar{z}_1)^{i_1} + \cdots + \bar{y}_\mu(\beta\bar{z}_\mu)^{i_1})^{\delta/i_1}} = \\ &\frac{\beta^\delta\sigma(\bar{z}_1, \ldots, \bar{z}_\mu, 0, \ldots, 0)}{\beta^\delta(\bar{y}_1\bar{z}_1^{i_1} + \cdots + \bar{y}_\mu\bar{z}_\mu^{i_1})^{\delta/i_1}} = \\ &\frac{\sigma(\bar{z}_1, \ldots, \bar{z}_\mu, 0, \ldots, 0)}{(\bar{y}_1\bar{z}_1^{i_1} + \cdots + \bar{y}_\mu\bar{z}_\mu^{i_1})^{\delta/i_1}} = A(\bar{x}_1, \ldots, \bar{x}_r). \end{aligned}$$

General case

Let us consider the map A and the polynomial a introduced in the case $\bar{x}_1 \neq 0$. Both enjoy a representation as polynomials in $\mathbb{F}_q[X]$. Since we are only interested in their evaluations at points of $\mathbb{F}_{q^m}$, we can assume that $\deg_{x_i}(A) < q^m$ and $\deg_{x_i}(a) < q^m$.

Now, let us define the polynomial $a_*(X)$ as $a_*(X) = a(X) - x_1^{\delta/i_1}A(X)$. Thanks to what we proved in the case $x_1 \neq 0$, we have that $a_*(X)$ satisfies the hypothesis of Lemma 18. Indeed, by construction, trivially $a_* \in \mathbb{F}_q[X]$, $\deg_{x_i}(a_*) < q$ for any $1 \leq i \leq r$.

Since $h(X) \notin \mathbb{F}_{q^m}[x_2, \ldots, x_r]$, by Lemma 18, we have that $h(X) = (1 - x_1^{q^m}) \cdot g(x_2, \ldots, x_r)$ for some $g(x_2, \ldots, x_r) \in \mathbb{F}_{q^m}$. So $a(X) = x_1^{\delta/i_1}A(X) + (1 - x_1^{q^m}) \cdot g(x_2, \ldots, x_r)$. ■

Corollary 20: Let C be a cyclic code over $\mathbb{F}_q$ as in Theorem 19. Then the coefficients of the general error locator polynomial can be written in the form given by the previous theorem.

Corollary 21: Let C be a code with $t = 2$ defined by $S_C = \{i_1, \dots, i_r\}$, with $i_1 = 1$, and let $\mathcal{L} = z^2 + x_1 z + b$ be a general error locator polynomial for C. If C is a primitive code, i.e. $n = q^m - 1$, then $b = x_1^2 A$ with $A \in \mathbb{F}_q[x_2/x_1^{i_2}, \dots, x_r/x_1^{i_r}]$.

Proof: Since $t = 2$, x_1 is zero if and only if there are no errors. Then, applying the previous theorem to C, we get that $b = x_1^2 A$ with $A \in \mathbb{F}_q[x_1^n, x_2/x_1^{i_2}, \dots, x_r/x_1^{i_r}]$. On the other hand, since C is primitive, x_1^n is zero when x_1 is zero, and it is 1 when x_1 is not zero. So for $\mu \in \{1, 2\}$, $x_1^n = 1$ and $b = x_1^2 \bar{A}$ with $\bar{A} = A|_{x_1^n=1}$. We claim that $b^* = x_1^2 \bar{A}$ is a valid location product also for the case $\mu = 0$, which follows from the fact that $\mu = 0$ if and only if $x_1 = 0$. ■

The previous corollary basically shows that in the case $t = 2$ the term of the form $(1 - x_1^{q^m-1})g$ does not appear in the expression of the locator coefficients.

A. Complexity of the proposed decoding approach

Now that we have Theorem 19 and Corollary 20, we can estimate the cost of evaluating the polynomial $\mathcal{L}_C$ (at the syndrome vector $\mathbf{s}$) in the more general case.

First, we observe that we can always choose $\lambda = n$, neglect the cost of computing the values $\frac{x_h}{x_1^{i_h}}$ and consider polynomials in the new obvious variables.
Second, we recall that in [27] Ballico, Elia and Sala describe a method to evaluate a polynomial in $\mathbb{F}_q[x_1, \dots, x_r]$ of total degree δ with a complexity $O(\delta^{r/2})$.
Finally, we can estimate our δ by observing that, thanks to Corollary 20, we have a bound on the degree of *each* z-coefficient of $\mathcal{L}_C$ in any new variable, so that its total degree is easily shown to be at most

$$\delta \quad \leq \quad \left((q^m - 1)(r - 1) + \frac{q^m - 1}{n}\right),$$

then, using the method in [27], the evaluation of (the z-coefficients of) $\mathcal{L}_C$ at $\mathbf{s}$ costs at most

$$O\left(t\left((q^m - 1)(r - 1) + \frac{q^m - 1}{n}\right)^{r/2}\right). \qquad (13)$$

So, we get that the cost of the decoding approach we are proposing is upper bounded by

$$O\left(n^2 + t\left((q^m - 1)(r - 1) + \frac{q^m - 1}{n}\right)^{r/2}\right). \qquad (14)$$

We conclude this section showing that there are infinite families of codes for which this approach is competitive with more straightforward methods (even for low values of t).

Let us fix the number of syndromes r, and let γ be an integer $\gamma \geq 1$. Let $C^q_{r,\gamma}$ be the set of all codes over $\mathbb{F}_q$ with length n such that the splitting field of $x^n - 1$ over $\mathbb{F}_q$ is $q^m - 1 = O(n^\gamma)$ (and $\gcd(n, q) = 1$).
For codes in $C^q_{r,\gamma}$, the complexity (14) of this decoding is at most

$$r \geq 2, \quad O\left(rtn^{\gamma r/2}\right), \qquad\qquad r = 1, \quad O(n^2 + tn^{\frac{\gamma-1}{2}}). \qquad (15)$$

So, any family $C^q_{r,\gamma}$ provides a class containing infinite codes which can be decoded in polynomial time, with infinite values of distance and length (r and γ are fixed). These classes extend widely the classes which are known to be decodable in polynomial time up to the *actual distance*.

V. On some classes of binary codes with $t = 2$

In this section we treat the case of 2-error correcting codes, vastly expanding and generalizing previous results in [13] (and obtaining simpler descriptions).

In [13] all codes with $t = 2$ and $n < 63$ were analyzed, which is a total of 952 codes. For each code (except for 5 cases), it was shown that either it belongs to one of four given classes, which have a sparse representation for their general error locator polynomials, or it is an equivalent code/subcode that could be decoded with the same locator (see Section II-D). However, one of these four classes enjoyed only an implicit representation for the general error locator polynomial, which would make the evaluation still efficient, but whose explicit representation might be non-sparse. We now present our improvement: we show that all codes with $t = 2$ and $n < 105$ can be grouped in **one** class, enjoying an **explicit** sparse general error locator polynomial, except for some cases, whose (sparse) locator can be determined easily with computer-assisted calculations. The codes spanned by this unified representation are now 4195 out of 4810.

To write our results in a more readable way, we adopt here a different notation for syndromes. Instead of writing x_j for the j-th syndrome, which corresponds to the number i_j in the defining set, we will write X_{i_j}. In other words,

$$X_h \;=\; z_1^h + z_2^h, \qquad \text{for any } 0 \leq h \leq n,$$

where z_1 and z_2 are the two error locations, which are different from zero and distinct only when $\mu = t$.
The main proposition of this section is the following.

Proposition 22: Let C be a code with $\{\lambda, s, s - \lambda, s - \lambda - l, \lambda - l, s - 2l\} \subset S_C$ and let $t = 2$. Then

$$X_{s-\lambda} X_\lambda + X_s \;=\; (z_1 z_2)^l (X_{s-\lambda-l} X_{\lambda-l} + X_{s-2l}) \qquad (16)$$

Proof: Let us start considering the left-hand side of the equality:

$$X_{s-\lambda}X_\lambda + X_s \;=\; (z_1^{s-\lambda} + z_2^{s-\lambda})(z_1^\lambda + z_2^\lambda) + z_1^s + z_2^s \;=$$

$$z_1^{s-\lambda}z_2^\lambda + z_1^\lambda z_2^{s-\lambda} \;=\; (z_1z_2)^l(z_1^{s-\lambda-l}z_2^{\lambda-l} + z_1^{\lambda-l}z_2^{s-\lambda-l})\,.$$

Now we consider the right-hand side:

$$\begin{aligned} & X_{s-\lambda-l}X_{\lambda-l} + X_{s-2l} \\ = \;& (z_1^{s-\lambda-l} + z_2^{s-\lambda-l})(z_1^{\lambda-l} + z_2^{\lambda-l}) + z_1^{s-2l} + z_2^{s-2l} \\ = \;& z_1^{s-\lambda-l}z_2^{\lambda-l} + z_1^{\lambda-l}z_2^{s-\lambda-l}\,, \end{aligned}$$

proving relation (16). ■

From Proposition 22, we can easily derive a sparse general locator for an ample class of codes, as described in the following corollary.

Corollary 23: Let C be a code with $t = 2$, $\{1, \lambda, s, s-\lambda, s-\lambda-l, \lambda-l, s-2l\} \subset S_C$, $(l, n) = 1$ and $(s-2\lambda, n) = 1$. Let l^+ be the inverse of l modulo n. Then the locator polynomial is

$$\mathcal{L} = z^2 + X_1z + \left(\frac{X_{s-\lambda}X_\lambda + X_s}{X_{s-\lambda-l}X_{\lambda-l} + X_{s-2l}}\right)^{l^+}. \qquad (17)$$

Proof: Since $1 \in S_C$, $\mathcal{L}$ can be written as $\mathcal{L} = z^2 + X_1z + b$, where b must satisfy $b(\mathbf{s}) = \hat{z}_1\hat{z}_2$ when $\mu = 2$ and $b(\mathbf{s}) = 0$ when $\mu = 1$.

$\mu = \mathbf{2}$. From (16) we have

$$(z_1z_2)^l \;=\; \frac{X_{s-\lambda}X_\lambda + X_s}{X_{s-\lambda-l}X_{\lambda-l} + X_{s-2l}}$$

and from the fact that $(l, n) = 1$ we immediately have (passing to the evaluations at $mathbfs$)

$$\hat{z}_1\hat{z}_2 \;=\; \left(\frac{X_{s-\lambda}X_\lambda + X_s}{X_{s-\lambda-l}X_{\lambda-l} + X_{s-2l}}\right)^{l^+}$$

However, the numerator or the denominator might be zero (once evaluated at $\mathbf{s}$). Since the eveluation of their ratio is $\hat{z}_1\hat{z}_2$, which is nonzero when $\mu = 2$, then in this case $X_{s-\lambda}X_\lambda + X_s \neq 0$ if and only if $X_{s-\lambda-l}X_{\lambda-l}+X_{s-2l} \neq 0$. Therefore, it is enough to prove $X_{s-\lambda}X_\lambda+X_s \neq 0$ (when evaluated at a syndrome vector corresponding to an error of weight 2). We must then prove that

$$(\hat{z}_1{}^{s-\lambda} + \hat{z}_2{}^{s-\lambda})(\hat{z}_1{}^\lambda + \hat{z}_2{}^\lambda) + (\hat{z}_1{}^s + \hat{z}_2{}^s) \neq 0$$

which is $(\hat{z}_1\hat{z}_2)^\lambda(\hat{z}_1{}^{s-2\lambda} + \hat{z}_2{}^{s-2\lambda}) \neq 0$. Since $\hat{z}_1\hat{z}_2 \neq 0$, we must only show that $\hat{z}_1{}^{s-2\lambda} \neq \hat{z}_2{}^{s-2\lambda}$. Since $\hat{z}_1{}^n = \hat{z}_2{}^n = 1$, if by contradiction we have $\hat{z}_1{}^{s-2\lambda} = \hat{z}_2{}^{s-2\lambda}$, we would also have $\hat{z}_1{}^{(s-2\lambda,n)} = \hat{z}_1{}^{(s-2\lambda,n)}$, i.e. $\hat{z}_1 = \hat{z}_2$, which is impossible ($\mu = 2$).

$\mu = \mathbf{1}$. When there is only one error, the evaluation of $X_{s-\lambda}X_\lambda + X_s$ is zero, as well as the evaluation of $X_{s-\lambda-l}X_{\lambda-l}+X_{s-2l}$, and so by our convention their ratio is zero, that is, $b(\mathbf{s}) = 0$ when $\mu = 1$, as claimed. ■

The code family defined in the previous corollary contains nearly all codes with $t = 2$ and $7 \leq n < 105$, but some codes are outside it (and they are not equivalent codes/subcodes with the same t).

Definition 24: Let C be a code such that:

- C satisfies the hypotheses of Corollary 23,
- or C is equivalent to a code satisfying them,
- or C is a subcode of a code satisfying them.

Then we call C a class-λ code.

There are not so many codes outside, at least for $n < 105$. To be more precise, a computer MAGMA check shows the following

Proposition 25: Let C be a 2-error correcting binary cyclic code with length $7 \leq n < 105$ and n odd. If C is not a class-λ code, then

- either its defining set is one of the following list

n	Defining set
25	$\{1\}$
31	$\{1, 5\}$
39	$\{0, 1\}$
51	$\{0, 1, 5\}$
63	$\{1, 5, 21\}, \{1, 21, 31\}, \{1, 15, 27\}, \{1, 9, 31\},$ $\{0, 1, 5, 13\}, \{0, 1, 7, 13\}, \{1, 7, 27\}$
69	$\{0, 1\}$
85	$\{1, 5\}, \{1, 15\}, \{1, 9\}$
87	$\{0, 1\}$
91	$\{1, 13, 39\}$
93	$\{1, 21\}, \{0, 1, 5\}, \{1, 9\}, \{0, 1, 11\}, \{1, 15\}, \{1, 45\}$

- or C is equivalent to one of the previous list,
- or C is equivalent to a subcode of one of the previous list.

For each code in the list is possible to determine a general locator with functional density at most 15.

We observe that for some of these codes a sparse general locator was already provided in [13] and that we could group all them in a few classes, but we prefer not to do it.

A. *Sparsity*

Thanks to Corollary 23 and Proposition 25, we can derive the following theorem.

Theorem 26: For all codes in Corollary 23 the functional density of their presented locator is **constant**. In particular, these locators are sparse according to Conjecture 11. Moreover, the functional density for the general error locator polynomial of all codes with $t = 2$ and $n < 105$ does not exceed 15.

VI. On some classes of binary codes with $t = 3$

In this section we provide explicit sparse representations for some infinite classes of binary codes with correction capability $t = 3$. We also consider all binary codes with $t = 3$ and $n < 63$, showing that they can be regrouped in few classes and we provide a general error locator polynomial for all these codes. In [28] Chen produced a table of the minimum distances of binary cyclic codes of length at most 65. This table was extended to length at most 99 by Promhouse and Tavares [29].

The following theorem lists binary cyclic codes with $t = 3$ and $n < 63$ up to equivalence and subcodes that we obtain with MAGMA computer algebra system [30].

Theorem 27: Let C be an $[n, k, d]$ code with $d \in \{7, 8\}$ and $15 \leq n < 63$ (n odd). Then there are only three cases.

1) Either C is one of the following:
   $n = 15, S_C = \{1, 3, 5\}, n = 21, S_C = \{1, 3, 5\}, S_C = \{1, 3, 7, 9\}, S_C = \{0, 1, 3, 7\}$;
   $n = 23, S_C = \{1\}, n = 31, S_C = \{1, 3, 5\}, S_C = \{0, 1, 7, 15\}$;
   $n = 35, S_C = \{1, 3, 5\}, S_C = \{1, 5, 7\}, n = 45, S_C = \{1, 3, 5\}, S_C = \{1, 5, 9, 15\}$;
   $n = 49, S_C = \{1, 3\}, n = 51, S_C = \{1, 3, 9\}, n = 55, S_C = \{0, 1\}$;
2) or C is a subcode of one of the codes of case 1);
3) or C is equivalent to one of the codes of the above cases.

Subcodes and equivalences are described in Table VII in the Appendix A. By Theorem 12 [13], we need to find a general error locator polynomial only for the codes in 1). For our purposes, it is convenient to regroup the codes as showed in the following theorem.

Theorem 28: Let C be an $[n, k, d]$ code with $d \in \{7, 8\}$ and $15 \leq n < 63$ (n odd). Then there are six cases

1) either C is a BCH code, i.e. $S_C = \{1, 3, 5\}$,
2) or C admits a defining set containing $\{1, i, i+1, i+2, i+3, i+4\}$ where i and $i+2$ are not zero modulo n,
3) or C admits a defining set containing $\{1, 3, 2^i + 2^j, 2^j - 2^i, 2^j - 2^{i+1}\}$ with $i \geq 0$ and $j \geq i + 2$,
4) or C admits a defining set containing $\{1, 3, 9\}$ and $(n, 3) = 1$,
5) or C is one of the following:
   - $n = 21$, $S_C = \{0, 1, 3, 7\}$;
   - $n = 51$, $S_C = \{1, 3, 9\}$;
   - $n = 55$, $S_C = \{0, 1\}$.
6) or C is a subcode of one of the codes of the above cases,
7) or C is equivalent to one of the codes of the above cases.

Proof: It is enough to inspect Case 1) of Theorem 27 ■

Corollary 29: Let C be a code with length $n < 63$ and distance $d \in \{7, 8\}$. Then C is equivalent to a code D s.t $1 \in S_D$.

Proof: It is an immediate consequence of Theorem 27. ■

Let C be a code with $t = 3$, $\mathbf{s}$ a correctable syndrome and $\bar{z}_1$, $\bar{z}_2$, $\bar{z}_3$ the error locations. Then $\mathcal{L}(X, z) = z^3 + az^2 + bz + c$, where $a, b, c \in \mathbb{F}_2[X]$, and $a(\mathbf{s}) = \bar{z}_1 + \bar{z}_2 + \bar{z}_3$, $b(\mathbf{s}) = \bar{z}_1\bar{z}_2 + \bar{z}_1\bar{z}_3 + \bar{z}_2\bar{z}_3$, $c(\mathbf{s}) = \bar{z}_1\bar{z}_2\bar{z}_3$. Moreover, there are three errors if and only if $c(\mathbf{s}) \neq 0$, there are two errors if and only if $c(\mathbf{s}) = 0$ and $b(\mathbf{s}) \neq 0$, and there is one error if and only if $c(\mathbf{s}) = b(\mathbf{s}) = 0$ and $a(\mathbf{s}) \neq 0$. Note that from the previous corollary any code with $t = 3$ and $n < 63$ is equivalent to a code with 1 in the defining set. This means that for all our codes the general error locator polynomial is of the form

$$\mathcal{L}(X, z) = z^3 + x_1 z^2 + bz + c,$$

where x_1 is the syndrome corresponding to $1 \in S_C$. So we are left with finding the coefficients b and c. Of course, b in the $t = 3$ case should not be confused with b in the case of $t = 2$ case. Also, when $3 \in S_C$, actually we need to find only one of the two coefficients because in this case by Newton's identities [18] we get $c = x_1^3 + x_3 + x_1 b$, which involves only known syndromes, so from one coefficient we can easily obtain the other. In the following, $\Sigma_{l,m}$ will denote all the six terms of the type $z_i^l z_j^m$, $i, j \in \{1, 2, 3\}$, and $\Sigma_{l,m,r}$ denotes all the six terms of the type $z_i^l z_j^m z_k^r$, $i, j, k \in \{1, 2, 3\}$.

Let us consider the codes in 1) of Theorem 28. We have the following well-known result.

Theorem 30: Let C be a BCH code with $t = 3$. Then $\mathcal{L}(X, z) = z^3 + x_1 z^2 + bz + c$ with

$$b = \frac{(x_1^2 x_3 + x_5)}{(x_1^3 + x_3)}, \quad c = \frac{(x_1^3 x_3 + x_1^6 + x_3^2 + x_1 x_5)}{(x_1^3 + x_3)}$$

Proof: It enough to apply Newton's identities. ■

The next theorem provides a general error locator polynomial for codes in 2) of Th. 28.

Theorem 31: Let C be a code with $t = 3$ and S_C containing $\{1, i, i+1, i+2, i+3, i+4\}$ where i and $i+2$ are not zero modulo n. Then $\mathcal{L}(X, z) = z^3 + x_1 z^2 + bz + c$ with

$$b = \frac{x_i U + x_{i+1} V}{W}, \quad c = \frac{x_{i+1} U + x_{i+2} V}{W}$$

where $U = x_{i+4} + x_1 x_{i+3}$, $V = x_{i+3} + x_1 x_{i+2}$ and $W = x_{i+1}^2 + x_i x_{i+2}$.

Proof: Let us suppose that three errors occur, that is, $\mathbf{e}$ has weight three, and let $\mathbf{s}$ be its syndrome vector. It is a simple computation to show, using the Newton's identities

$$\begin{cases} x_{i+4} = x_1 x_{i+3} + b x_{i+2} + c x_{i+1} \\ x_{i+3} = x_1 x_{i+2} + b x_{i+1} + c x_i \end{cases}$$

that $b = \frac{x_i U + x_{i+1} V}{W}$, and $c = \frac{x_{i+1} U + x_{i+2} V}{W}$, where $W = x_{i+1}^2 + x_i x_{i+2} = \Sigma_{i,i+2}$ which cannot be zero because i and $i + 2$ are not zero modulo n. Then, when $\mu = 3$, $\mathcal{L}(\mathbf{s}, z)$ is the error locator polynomial for C.

Let us show that it is actually a general error locator polynomial for C. We have that

$$\begin{aligned} x_{i+1}U + x_{i+2}V = {} & (z_1^{i+1} + z_2^{i+1} + z_3^{i+1}) \cdot \qquad (18)\\ & \cdot \left(z_1^{i+4} + z_2^{i+4} + z_3^{i+4} + (z_1 + z_2 + z_3)(z_1^{i+3} + z_2^{i+3} + +z_3^{i+3})\right)\\ & + (z_1^{i+2} + z_2^{i+2} + z_3^{i+2})\cdot\\ & \cdot \left(z_1^{i+3} + z_2^{i+3} + z_3^{i+3} + (z_1 + z_2 + z_3)(z_1^{i+2} + z_2^{i+2} + z_3^{i+2})\right)\\ & = \Sigma_{1,i+1,i+3}, \end{aligned}$$

and

$$\begin{aligned} x_iU + x_{i+1}V = {} & (z_1^{i} + z_2^{i} + z_3^{i}) \cdot \qquad (19)\\ & \cdot \left(z_1^{i+4} + z_2^{i+4} + z_3^{i+4} + (z_1 + z_2 + z_3)(z_1^{i+3} + z_2^{i+3} + z_3^{i+3})\right)\\ & + (z_1^{i+1} + z_2^{i+1} + z_3^{i+1})\\ & \left(z_1^{i+3} + z_2^{i+3} + z_3^{i+3} + (z_1 + z_2 + z_3)(z_1^{i+2} + z_2^{i+2} + z_3^{i+2})\right)\\ & = \Sigma_{1,i,i+3} + \Sigma_{1,i+1,i+2} + \Sigma_{i+1,i+3}, \end{aligned}$$

Let us suppose that $\mu = 2$. In this case, $W = z_1^i z_2^{i+2} + z_1^{i+2} z_2^i$, which is again different from zero. Furthermore, $x_{i+1}U + x_{i+2}V = \Sigma_{1,i+1,i+3}$ is zero because $\mu = 2$. Finally, $x_iU + x_{i+1}V$ is different from zero because $x_iU + x_{i+1}V = \Sigma_{1,i,i+3} + \Sigma_{1,i+1,i+2} + \Sigma_{i+1,i+3}$ and $\Sigma_{i+1,i+3}$ cannot be zero. When $\mu = 1$, $W = z_1^i z_2^{i+2} + z_1^{i+2} z_2^i = 0$ and $x_iU + x_{i+1}V = \Sigma_{i+1,i+3} = 0$. ■

To obtain a general error locator polynomial for codes in 3) of Theorem 28, we need the following lemma.

Lemma 32: Let $\sigma_k = \sum_{1 \le i_1 < \cdots < i_k \le 3} z_{i_1} \cdots z_{i_k}$ be the kth elementary symmetric polynomial in the variables z_1, z_2, z_3 over $\mathbb{F}_2$, where $k \in \{1, 2, 3\}$, and let $x_h = \sum_{l=1}^{3} z_l^h \in \mathbb{F}_2[z_1, z_2, z_3]$ be the power sum polynomial of degree h, with $h \ge 0$. Then, for $i \ge 0$ and $j \ge i+2$,

$$x_1^{2^i+2^j} + x_{2^i+2^j} = \sigma_2^{2^i} x_{2^j-2^i} + \sigma_3^{2^i} x_{2^j-2^{i+1}}$$

Proof: $x_1^{2^i+2^j} = (z_1 + z_2 + z_3)^{2^i+2^j} = (z_1 + z_2 + z_3)^{2^i}(z_1 + z_2 + z_3)^{2^j} = x_{2^i+2^j} + \Sigma_{2^i,2^j}$. On the other hand, $\sigma_2^{2^i} x_{2^j-2^i} = (z_1z_2 + z_1z_3 + z_2z_3)^{2^i}(z_1^{2^j-2^i} + z_2^{2^j-2^i} + z_3^{2^j-2^i}) = \Sigma_{2^i,2^j} + \Sigma_{2^i,2^i,2^j-2^i}$ and $\sigma_3^{2^i} x_{2^j-2^{i+1}} = (z_1z_2z_3)^{2^i}(z_1^{2^j-2^{i+1}} + z_2^{2^j-2^{i+1}} + z_3^{2^j-2^{i+1}}) = \Sigma_{2^i,2^i,2^j-2^i}$. So $x_1^{2^i+2^j} = x_{2^i+2^j} + \sigma_2^{2^i} x_{2^j-2^i} + \sigma_3^{2^i} x_{2^j-2^{i+1}}$.

■

Theorem 33: Let C be a code with $t = 3$ and S_C containing $\{1, 3, 2^i + 2^j, 2^j - 2^i, 2^j - 2^{i+1}\}$ with $i \ge 0$ and $j \ge i + 2$. Then $\mathcal{L}(X, z) = z^3 + x_1z^2 + bz + c$ with

$$b = \left(\frac{x_{2^j-2^{i+1}}U + V}{W}\right)^{(2^i)^+}, \quad c = \left(\frac{x_{2^j-2^i}U + x_1^{2^i}V}{W}\right)^{(2^i)^+}$$

where $U = (x_1^3 + x_3)^{2^i}$, $V = x_1^{2^i+2^j} + x_{2^i+2^j}$, $W = x_{2^j-2^i} + x_1^{2^i} x_{2^j-2^{i+1}}$ and $(2^i)^+$ is the inverse of 2^i modulo n.

Proof: Since the syndrome x_1 is a known syndrome, that is, $1 \in S_C$, we have that $a = x_1$. From the Newton identity $c = x_1^3 + x_3 + x_1b$ we get that

$$c^{2^i} = x_1^{3\times 2^i} + x_3^{2^i} + x_1^{2^i} b^{2^i} \qquad (20)$$

On the other hand, by the previous lemma, we have that

$$x_1^{2^i+2^j} + x_{2^i+2^j} = b^{2^i} x_{2^j-2^i} + c^{2^i} x_{2^j-2^{i+1}} \qquad (21)$$

Taking into account (20) and (21), a few computations lead to the equalities $b^{2^i} = \left(\frac{x_{2^j-2^{i+1}}U+V}{W}\right)$ and $c^{2^i} = \left(\frac{x_{2^j-2^i}U+x_1^{2^i}V}{W}\right)$. Suppose that $\mu = 3$. Then $W = (z_1^{2^j-2^i} + z_2^{2^j-2^i} + z_3^{2^j-2^i}) + (z_1^{2^i} + z_2^{2^i} + z_3^{2^i})(z_1^{2^j-2^{i+1}} + z_2^{2^j-2^{i+1}} + z_3^{2^j-2^{i+1}}) = \Sigma_{2^i,2^j-2^{i+1}}$. Since j is an integer, it is not possible that $2^i = 2^j - 2^{i+1}$, then W is different from zero. Also $x_{2^j-2^i}U + x_1^{2^i}V = \Sigma_{2^i,2^{i+1},2^j-2^i} + \Sigma_{2^i,2^i,2^j}$ and $x_{2^j-2^{i+1}}U + V = \Sigma_{2^i,2^{i+1},2^j-2^{i+1}} + \Sigma_{2^{i+1},2^j-2^i}$. From the previous computations we get that if $\mu = 2$ then $W \ne 0$, $x_{2^j-2^i}U + x_1^{2^i}V = 0$, and $x_{2^j-2^{i+1}}U + V \ne 0$. The last equality is because $\Sigma_{2^{i+1},2^j-2^i} \ne 0$. Furthermore, if $\mu = 1$, then $x_{2^j-2^{i+1}}U + V = 0$.

■

Finally, let us consider the codes in 4) of Theorem 28. In [11] Elia presents an algebraic decoding for the $(23, 12, 7)$ Golay code providing the error locator polynomials for μ errors, for μ from one to three. In [16] Lee proves that the error locator polynomial $L^{(3)}$ corresponding to three errors is actually a weak error locator polynomial for this code. Notice that $L^{(3)}$ is a weak error locator polynomial for all cyclic codes C with $t = 3$, S_C containing $\{1, 3, 9\}$ and $(n, 3) = 1$. Next theorem proves that one can obtain a general error locator polynomial for these codes by slightly modifying $L^{(3)}$.

Theorem 34: Let C be a code with $t = 3$ and S_C containing $\{1, 3, 9\}$ with $(n, 3) = 1$. Then $\mathcal{L}(X, z) = z^3 + x_1z^2 + bz + c$ with

$$b = (x_1^2 + D^{l^*})h, \quad c = (x_3 + x_1D^{l^*})h,$$

where $D = \left(\frac{x_9+x_1^9}{x_3+x_1^3}\right) + (x_1^3 + x_3)^2$, $h = \frac{(x_1^3+x_3)}{(x_1x_2+x_3)}$, $l = 3$ and l^* is the inverse of l modulo $2^m - 1$ with $\mathbb{F}_{2^m}$ the splitting field of $x^n - 1$ over $\mathbb{F}_2$.

Proof: Since $1 \in S_C$, we have that $a = x_1$. From the following Newton identities

$$\begin{cases} x_9 = x_1x_8 + bx_7 + cx_6\\ x_7 = x_1x_6 + bx_5 + cx_4\\ x_5 = x_1x_4 + bx_3 + cx_2\\ x_3 = x_1x_2 + bx_1 + c \end{cases}$$

using the equalities $x_6 = x_3^2$, and $x_{2^i} = x_1^{2^i}$ for $i \ge 0$, we get

$$\left(\frac{x_9 + x_1^9}{x_3 + x_1^3}\right) + (x_1^3 + x_3)^2 = (b + x_1^2)^3 \qquad (22)$$

So $b = x_1^2 + D^{l^*}$. From $x_3 = x_1 x_2 + b x_1 + c$, we find $c = x_3 + x_1 D^{l^*}$. Let us prove that $\mathcal{L}$ is a general error locator polynomial. By Lemma 1 and Lemma 2 in [16], it is enough to note that when there is one error $h = 0$, while when there are two or three errors $h = 1$. ■

In Tables I, II we list binary cyclic codes, up to equivalence and subcodes, with length less than 121 which are covered by Theorem 31 and Theorem 33 respectively. We observe that in each table we also report BCH codes.

Table III shows a general error locator polynomial for each code in Case 1) of Theorem 27 with $n < 55$. Since the codes in Cases 2) and 3) of Theorem 27 are equivalent or subcodes of the codes in Case 1), so (Theorem 3) their general error locator polynomial is the same or can be easily deduced from one of the general error locator polynomials in the table.
In Table III the codes are grouped according to increasing lengths and are specified with defining sets containing only primary syndromes. For each of these codes, the coefficients b and c of the general error locator polynomial is reported respectively in the second column and in the third column; The value in the fourth column explains which point of Theorem 28 has been used to describe the corresponding code family. In all cases except case 4 and for the codes with length $n = 49$ and $n = 51$, b and c are expressed in terms of primary syndromes: if the defining set in the last column is $S_C = \{i_1, i_2, \ldots, i_j\}$ with $i_1 < i_2 < \cdots < i_j$, then x_k denotes the syndrome corresponding to i_k, for $k = 1, 2, \ldots, j$. When 0 belongs to the defining set, it will be treated as if it were an n, with n the length of the code. For instance, for the code with length $n = 21$ and defining set $\{0, 1, 3, 7\}$ the syndrome corresponding to 0 is x_4.

Codes described by the point 4 of Theorem 28 maintain the notation of Proposition 34, so x_i denotes the syndrome corresponding to i. In the case of the code with $n = 49$, x_1, x_2, x_3 denote the syndromes corresponding to $1, 3, 5$ respectively, while for the codes with length 51, x_1, x_2, x_3, x_4, x_5 denote the syndromes corresponding to $1, 3, 9, 13, 15$ respectively. The coefficient a of the general error locator polynomial is not reported in Table III because any code in Case 1) of Theorem 27 has 1 in its defining set, so in all cases $a = x_1$. A general error locator for the codes with $t = 3$ and $n = 55$ is showed in Table VIII in the Appendix A.

A. Sparsity

An easy ispection of Theorem 31, 33, 34 provides the following theorem.

Theorem 35: For all codes in the cases covered by Theorem 31, 33, 34, the function density of their presented locator is **constant**.
In particular, these locators are sparse according to Conjecture 11.

VII. On the complexity of decoding cyclic codes

In this section we complete the investigation of the link between Conjecture 11 and the decoding problem for cyclic codes.

A. Complexity of the proposed decoding approach: t=2,3

Theorem 26 and Theorem 35 provide (infinite) classes of codes with $t = 2$ and $t = 3$ for which the evaluation of $\mathcal{L}_C$ costs $O(1)$, and so the decoding process costs $O(n^2)$. For $t = 2$ and $t = 3$ exhaustive searching method cost, respectively, $O(n^2)$ and $O(n^3)$. For $t = 2$ we match the best-known complexity and for $t = 3$ our method is better.

B. Comparison with other approaches

In the last years, several methods were proposed for decoding *binary* quadratic residue (QR) codes generated by irreducible polynomials. In [8], Chang and Lee propose three algebraic decoding algorithms based on Lagrange Interpolation Formula (LIF) for these codes. They introduce a variation for the general error locator polynomial, which we may call fixed-weight locator. A *fixed-weight locator* is a polynomial able to correct all errors of a fixed weight via the evaluation of the corresponding syndromes. They develop a method to obtain a representation of the primary unknown syndrome in terms of the primary known syndrome and a representation of the coefficients of both fixed-weight locator and general error locator polynomial for these codes. These polynomials are explicitly obtained for the $(17, 9, 5)$, $(23, 12, 7)$, $(41, 21, 9)$ QR codes. In Table IV we treat these three codes one per column showing the number of terms relevant to the alternative representations. For each code, the second row deals with representation of the chosen primary unknown syndrome, while the last deal with two locators.
Note that, for all the three codes, the general error locator polynomials are sparse (even without using the rational representation) as forseen in Conjecture 11. In particular the $(41, 21, 9)$ code has correction capability $t = 4$ and the number of terms of its locator is less than $n^{\epsilon} = 41^3 = 68921$. Observe also that the evaluation of

TABLE IV
NUMBER OF TERMS OF UNKNOWN SYNDROME, FIXED-WEIGHT LOCATOR AND GENERAL ERROR LOCATOR

	$(17,9,5)$	$(23,12,7)$	$(41,21,9)$
Splitting field	$\mathbb{F}_{2^8}$	$\mathbb{F}_{2^{11}}$	$\mathbb{F}_{2^{20}}$
Unknown syndrome	5	17	1355
Fixed-weight locator	4	15	1270
General error locator	4	76	1380

the locators of the $(23,12,7)$ code in Table III and in [8] cost approximately the same.

In [31], Chang et al. propose to decode *binary* cyclic codes generated by irreducible polynomials using, as in [8], an interpolation formula in order to get the general error locator polynomial but in a slightly different way. The general error locators they obtain satisfy at least one congruence relation, and they are explicitly found for the $(17,9,5)$ QR code, the $(23,12,7)$ Golay code, and one $(43,29,6)$ cyclic code. Table V shows the maximum number of terms for the coefficients of these three polynomials. Also in this case, the locators are *sparse* for the three codes.

TABLE V
MAXIMUM NUMBER OF TERMS AMONG THE LOCATOR COEFFICIENTS σ_i

	$(17,9,5)$	$(23,12,7)$	$(43,29,6)$
Splitting field	$\mathbb{F}_{2^8}$	$\mathbb{F}_{2^{11}}$	$\mathbb{F}_{2^{14}}$
General error locator	9	203	25

In [32], Lee et al. extend the method proposed by Chang and Lee in [8] for finding fixed-weight locators and general error locators for binary cyclic codes generated by irreducible polynomials to the case of *ternary cyclic codes* generated by irreducible polynomials. These polynomials are presented for two *ternary* cyclic codes, one $(11,6,5)$ code and one $(23,12,8)$ code. In Table VI we report the maximum number of terms of the coefficients of the general error locator for these two codes.

TABLE VI
MAXIMUM NUMBER OF TERMS AMONG THE LOCATOR COEFFICIENTS σ_i

	$(11,6,5)$	$(23,12,8)$
Splitting field	$\mathbb{F}_{3^5}$	$\mathbb{F}_{3^{11}}$
General error locator	232	15204

To discuss the *sparsity* of these cases one would need to know $\epsilon(3)$ from Conjecture 11. Assuming an optimistic stance, let us compare their sparsity with $\epsilon(3) = 3$, that is, let us assume the polynomial exponent of the ternary codes to be the same as that of binary codes (reasonably $\epsilon(3) \geq \epsilon(2)$).
The first locator is definitely sparse, with $|\mathcal{L}| = 232 < 1331 = 11^3$. For the second locator we have $|\mathcal{L}| = 15204$ which compared to $n^3 = 23^3 = 12167$ show that the locator is not sparse (although the numbers are close) and indeed we believe much sparser locators exist for this code, still to be found.

In the same paper ([32]) the authors give also an interesting upper bound on $|\mathcal{L}|$ which holds for any irreducible ternary cyclic code, as follows.

Proposition 36 ([32]): Let C be a ternary cyclic code of length n with defining set $S_C = \{1\}$, and error correction capability t. Each coefficient of a general error locator polynomial can be expressed as a polynomial in terms of the known syndrome x_1 and the number of terms of this polynomial is less than $\lfloor \frac{\sum_{\nu=1}^{t} 2^\nu \binom{n}{\nu}}{n} \rfloor$.
Indeed, we can generalize their result to the following theorem holding over any finite field.

Theorem 37: Let C be any cyclic code over $\mathbb{F}_q$ of length n with defining set $S_C = \{1\}$, $\gcd(n,q) = 1$ and error correction capability t. Each coefficient of a general error locator polynomial can be expressed as a polynomial in terms of the known syndrome x_1 and the number of terms of this polynomial is less than $\lfloor \frac{\sum_{\nu=1}^{t} (q-1)^\nu \binom{n}{\nu}}{n} \rfloor$.

Proof: By considering Corollary 20 and the fact that to obtain any locator coefficient, one can use simply (univariate) Lagrange interpolation on the set of correctable syndromes, which are obviously $1 + \sum_{\nu=1}^{t} (q-1)^\nu \binom{n}{\nu}$. ■

With q fixed, the codes covered by the previous theorem are actually the component of our families $C_{1,\gamma}^q$ for $\gamma \geq 1$. Depending on the actual considered length we will have the correct determination of γ, since this value strongly depends on the size of the splitting field. By (14) case $r = 1$, the time complexity of the decoding method for codes in $C_{1,\gamma}^q$ is

$$O\left(n^2 + tn^{(\gamma-1)/2}\right). \qquad (23)$$

Using the estimation given by Proposition 36, the complexity of the same decoding approach for these codes becomes

$$O\left(n^2 + tn^{t-1}\right). \qquad (24)$$

We observe that which of the two estimations is better depends on the particular values of t and γ.

VIII. CONCLUSIONS

This paper provides additional theoretical arguments supporting the sparsity of the general error locator polynomial for infinite families of cyclic codes over $\mathbb{F}_q$. For infinite classes of binary codes with $t = 2$ and $t = 3$ a sparse general error locator polynomial is obtained. Furthermore, for all binary cyclic codes with length less than 63 and correction capability 3, we see that the number of monomials never exceeds five times the code length.

We provide some argument showing the link between the locators' sparsity and the bounded-distance decoding complexity of cyclic codes, which might turn out to be of interest.

Appendix A
Some tables

Table VII report the codes with $t = 3$ and $n < 63$ grouped according to increasing lengths, and, within the same length according to Theorem 27, i.e. if two codes with the same length are equivalent or one is a subcode of the other,then they are in the same group. For each group there is a code in bold, which is the one reported in Table III, i.e. the code for which we determined a general error locator polynomial and that can be used to obtain locators for all the codes of the group.

In Table VIII we show the coefficients b and c of a general error locator polynomial for binary cyclic codes with $t = 3$ and $n = 55$. For the sake of conciseness, both b and c are represented in the form described in Theorem 19, where y_1 stands for x_1^{55}.

Acknowledgements

Most results in this paper are from the last author's PHD thesis and so she would like to thank her supervisor, the third author. The authors would like to thank Matteo Bonini and Michela Ceria for helpful suggestions and discussions.

References

[1] J. L. Massey, "Shift-register synthesis and BCH decoding," *Information Theory, IEEE Transactions on*, vol. 15, no. 1, pp. 122–127, 1969.

[2] S. Lin and E. Weldon, "Long BCH codes are bad," *Information and Control*, vol. 11, no. 4, pp. 445–451, 1967.

[3] G.-L. Feng and K. K. Tzeng, "A new procedure for decoding cyclic and BCH codes up to actual minimum distance," *Information Theory, IEEE Transactions on*, vol. 40, no. 5, pp. 1364–1374, 1994.

[4] R. He, I. S. Reed, T.-K. Truong, and X. Chen, "Decoding the $(47, 24, 11)$ quadratic residue code," *Information Theory, IEEE Transactions on*, vol. 47, no. 3, pp. 1181–1186, 2001.

[5] Y. Chang, T.-K. Truong, I. S. Reed, H. Cheng, and C.-D. Lee, "Algebraic decoding of $(71, 36, 11)$,$(79, 40, 15)$, and $(97, 49, 15)$ quadratic residue codes," *IEEE transactions on communications*, vol. 51, no. 9, pp. 1463–1473, 2003.

[6] T.-K. Truong, P.-Y. Shih, W.-K. Su, C.-D. Lee, and Y. Chang, "Algebraic decoding of the $(89, 45, 17)$ quadratic residue code," *Information Theory, IEEE Transactions on*, vol. 54, no. 11, pp. 5005–5011, 2008.

[7] T.-K. Truong, Y. Chang, Y.-H. Chen, and C.-D. Lee, "Algebraic decoding of $(103, 52, 19)$ and $(113, 57, 15)$ quadratic residue codes," *IEEE transactions on communications*, vol. 53, no. 5, pp. 749–754, 2005.

[8] Y. Chang and C.-D. Lee, "Algebraic decoding of a class of binary cyclic codes via Lagrange interpolation formula," *Information Theory, IEEE Transactions on*, vol. 56, no. 1, pp. 130–139, 2010.

[9] Y. Chang, C.-D. Lee, and K. Feng, "Multivariate interpolation formula over finite fields and its applications in coding theory," *arXiv preprint arXiv:1209.1198*, 2012.

[10] C.-D. Lee, Y. Chang, M.-H. Jing, and J.-H. Miao, "New method of predetermining unified unknown syndrome representations for decoding binary cyclic codes," *IET Communications*, vol. 6, no. 18, pp. 3339–3349, 2012.

[11] M. Elia, "Algebraic decoding of the $(23, 12, 7)$ Golay code," *Information Theory, IEEE Transactions on*, vol. 33, no. 1, pp. 150–151, 1987.

[12] E. Orsini and M. Sala, "Correcting errors and erasures via the syndrome variety," *Journal of Pure and Applied Algebra*, vol. 200, no. 1, pp. 191–226, 2005.

[13] E. Orsini and M. Sala, "General error locator polynomials for binary cyclic codes with $t \leq 2$ and $n < 63$," *IEEE transactions on information theory*, vol. 53, no. 3, pp. 1095–1107, 2007.

[14] C. Marcolla, E. Orsini, and M. Sala, "Improved decoding of affine-variety codes," *Journal of Pure and Applied Algebra*, vol. 216, no. 7, pp. 1533–1565, 2012.

[15] C.-D. Lee, Y. Chang, H.-H. Chang, and J.-H. Chen, "Unusual general error locator polynomial for the $(23, 12, 7)$ Golay code," *IEEE Communications Letters*, vol. 14, no. 4, pp. 339–341, 2010.

[16] C.-D. Lee, "Weak general error locator polynomials for triple-error-correcting binary Golay code," *IEEE communications letters*, vol. 15, no. 8, pp. 857–859, 2011.

[17] D. Augot, M. Bardet, and J.-C. Faugere, "On the decoding of binary cyclic codes with the newton identities," *Journal of Symbolic Computation*, vol. 44, no. 12, pp. 1608–1625, 2009.

[18] F. J. MacWilliams and N. J. A. Sloane, *The theory of error correcting codes. I and II.* Elsevier, 1977.

[19] W. W. Peterson and E. J. Weldon, *Error-correcting codes.* MIT press, 1972.

[20] V. Pless, R. A. Brualdi, and W. C. Huffman, *Handbook of coding theory. Vol. I, II.* Elsevier Science Inc., 1998.

[21] T. Mora and E. Orsini, "Decoding cyclic codes: the Cooper philosophy," in *Gröbner Bases, Coding, and Cryptography*, pp. 69–91, Springer, 2009.

[22] R. T. Chien, "Cyclic decoding procedures for Bose-Chaudhuri-Hocquenghem codes," *Information Theory, IEEE Transactions on*, vol. 10, no. 4, pp. 357–363, 1964.

[23] D. Schipani, M. Elia, and J. Rosenthal, "On the decoding complexity of cyclic codes up to the BCH bound," in *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pp. 835–839, IEEE, 2011.

[24] J. Hong and M. Vetterli, "Simple algorithms for BCH decoding," *Communications, IEEE Transactions on*, vol. 43, no. 8, pp. 2324–2333, 1995.

[25] J. Bruck and M. Naor, "The hardness of decoding linear codes with preprocessing," *Information Theory, IEEE Transactions on*, vol. 36, no. 2, pp. 381–385, 1990.

[26] G. L. Mullen and D. Panario, *Handbook of Finite Fields*. CRC Press, 2013.

[27] E. Ballico, M. Elia, and M. Sala, "On the evaluation of multivariate polynomials over finite fields," *Journal of Symbolic Computation*, vol. 50, pp. 255–262, 2013.

[28] C.-L. Chen, *Some results on algebraically structured error-correcting codes*. Elsevier, 1969.

[29] G. Promhouse and S. E. Tavares, "The minimum distance of all binary cyclic codes of odd lengths from 69 to 99," *Information Theory, IEEE Transactions on*, vol. 24, no. 4, pp. 438–442, 1978.

[30] W. Bosma, J. Cannon, and C. Playoust, "The magma algebra system I: The user language," *Journal of Symbolic Computation*, vol. 24, no. 3, pp. 235–265, 1997.

[31] C.-D. Lee, Y. Chang, T.-K. Truong, and Y.-H. Chen, "More on general error locator polynomials for a class of binary cyclic codes," in *Information Theory and its Applications (ISITA), 2010 International Symposium on*, pp. 273–277, IEEE, 2010.

[32] C.-D. Lee, M.-H. Jing, and J.-H. Miao, “Algebraic decoding of a class of ternary cyclic codes,” in *Signal Processing, Communication and Computing (ICSPCC), 2012 IEEE International Conference on*, pp. 331–336, IEEE, 2012.

TABLE I

BINARY CYCLIC CODES WITH $t = 3$ AND LENGTH < 121 COVERED BY THEOREM 31

15, {1, 3, 5}	21, {1, 3, 5}	21, {1, 5, 9}	23, {0, 1}	31, {0, 1, 7, 15}	31, {1, 3, 5}
35, {1, 3, 5}	35, {1, 5, 7}	45, {1, 3, 5}	49, {1, 3}	63, {1, 3, 5}	63, {1, 3, 11, 23, 27, 31}
63, {1, 5, 9, 13, 21}	63, {1, 3, 11, 13, 23}	63, {1, 5, 11, 13, 15}	63, {1, 15, 23, 31}	63, {1, 5, 13, 15, 21}	63, {0, 1, 15, 31}
63, {1, 5, 9, 13, 15}	63, {1, 11, 13, 15, 23, 27}	69, {1, 3, 23}	69, {0, 1, 3}	75, {1, 3, 5}	75, {1, 3, 25}
77, {1, 3}	77, {1, 7, 33}	85, {1, 3, 5}	85, {1, 7, 13, 15, 17}	85, {1, 15, 29, 37}	85, {0, 1, 21, 37}
89, {0, 1, 3}	89, {0, 1, 11}	91, {1, 3}	91, {1, 9, 19}	91, {1, 7, 9, 11, 13}	93, {1, 5, 17, 33}
93, {1, 7, 9, 17}	93, {1, 15, 17, 31, 33}	93, {1, 11, 23, 45}	93, {1, 17, 23, 31, 33}	93, {1, 9, 17, 33}	93, {1, 3, 5}
105, {1, 3, 5}	105, {1, 3, , 13, 25}	105, {1, 5, 9, 17}	105, {1, 9, 13, 25}	105, {1, 5, 7, 9, 11}	105, {1, 3, 9, 17, 25}
105, {1, 3, 17, 21, 25}	105, {1, 3, 11, 17, 45}	105, {1, 5, 9, 49}	105, {1, 3, 17, 25, 49}	105, {1, 9, 11, 13, 15, 17}	105, {1, 9, 13, 45, 49}
105, {1, 9, 17, 25, 49}	105, {1, 9, 11, 13, 45}	105, {0, 1, 9, 13}	105, {1, 3, 17, 35}	113, {0, 1}	115, {1, 23, 25}
117, {0, 1, 3}	117, {0, 1, 21, 29}	119, {1, 3}	119, {1, 7, 17}	119, {1, 11, 13}	

TABLE II

BINARY CYCLIC CODES WITH $t = 3$ AND LENGTH < 121 COVERED BY THEOREM 33

15, {1, 3, 5}	21, {1, 3, 5}	21, {1, 3, 7, 9}	31, {1, 3, 5}	35, {1, 3, 5}	45, {1, 3, 5}	49, {1, 3}
63, {1, 3, 5}	75, {1, 3, 5}	77, {1, 3}	85, {1, 3, 5}	91, {1, 3}	93, {1, 3, 15, 31, 33}	93, {1, 3, 7, 9}
93, {1, 3, 5}	105, {1, 3, 5}	117, {1, 3, 7}	119, {1, 3}			

TABLE III

BINARY CYCLIC CODE WITH $t = 3$ AND $n < 55$

n	b	c	Case	Codes
15	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3, 5}
21	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3, 5}
	$\frac{x_2^2(x_1^3+x_2)+(x_1^9+x_4)}{x_3+x_1x_2^2}$	$\frac{x_3(x_1^3+x_2)+x_1(x_1^9+x_4)}{x_3+x_1x_2^2}$	3	{1, 3, 7, 9}
	$x_4x_1^2+x_3^3x_1^2+x_3^2x_2^3+x_3^2x_2^2x_1^3+x_3^2x_1^9+x_3x_2^3x_1^{28}+$ $x_3x_2^2x_1^{10}+x_3x_2x_1^{13}+x_3x_1^{37}+x_2^7x_1^{44}+x_2^7x_1^{23}+$ $x_2^6x_1^{47}+x_2^6x_1^5+x_2^5x_1^{50}+x_2^4x_1^{53}+x_2^4x_1^{32}+x_2^3x_1^{56}+$ $x_2^3x_1^{35}+x_2^2x_1^{59}+x_2^2x_1^{38}+x_2x_1^{41}+x_2x_1^{20}+x_1^{23}+x_1^2$	$x_1^3+x_2+x_1b$	5	{0, 1, 3, 7}
23	$\left(x_1^2+\left(\frac{x_9+x_1^9}{x_3+x_1^3}+(x_1^3+x_3)^2\right)^{1365}\right)\cdot$ $\cdot\frac{(x_1^3+x_3)}{(x_1x_2+x_3)}$	$(x_1^3+x_3+bx_1)\frac{(x_1^3+x_3)}{(x_1x_2+x_3)}$	4	{1}
31	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3, 5}
	$\frac{x_3^8(x_4+x_1x_3^2)+x_2^4(x_3^2+x_1x_3^4)}{(x_3^{12}+x_2^8)}$	$\frac{x_2^4(x_4+x_1x_3^2)+x_3^4(x_3^2+x_1x_3^4)}{(x_3^{12}+x_2^8)}$	2	{0, 1, 7, 15}
35	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3, 5}
	$\frac{x_3(x_1^{256}+x_1x_2^2)+x_1^8(x_2^2+x1^{1025})}{x_1^{16}+x_3x_1^{1024}}$	$\frac{x_1^8(x_1^{256}+x_1x_2^2)+x_1^{1024}(x_2^2+x1^{1025})}{x_1^{16}+x_3x_1^{1024}}$	2	{1, 5, 7}
45	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3, 5},
	$\frac{x_4(x_1x_3^2+x_1^{64})+x_1^{16}(x_3^2+x_1^{513})}{x_1^{512}x_4+x_1^{32}}$	$\frac{x_1^{16}(x_1x_3^2+x_1^{64})+x_1^{512}(x_3^2+x_1^{513})}{x_1^{512}x_4+x_1^{32}}$	2	{1, 5, 9, 15}
49	$\frac{(x_1^3x_2+x_1^6+x_2^2+x_1x_3)}{(x_1^3+x_2)}$	$\frac{(x_1^2x_2+x_3)}{(x_1^3+x_2)}$	1	{1, 3}
51	$x_1^2+(x_1^3+x_2)(\frac{x_3^2+x_5x_3}{q_1x_1}+(\frac{x_3+x_2^3}{x_5^4+x_2^3}+1)(\frac{x_1^2}{x_1^3+x_2}+$ $\frac{x_4+x_2^4x_1}{q_2}))$ $q_1=(x_3x_1^9+x_3x_2x_1^6+x_2^3x_1^9+x_3^2+x_3x_2^2x_1^3+x_2^4x_1^6+x_3x_2^3+$ $x_5x_1^3+x_5x_2+x_2^6)$ $q_2=(x_1^{16}+x_2^4x_1^4+x_4x_2+x_2^5x_1)$	$x_1^3+x_2+x_1b$	5	{1, 3, 9}

TABLE VII

BINARY CYCLIC CODES WITH $t = 3$ AND $n < 63$

n	Codes
15	{**1, 3, 5**}, {3, 5, 7}, {0, 3, 5, 7}, {0, 1, 3, 5}
21	{**1, 3, 5**}, {1, 5, 9}, {1, 3, 5, 9}
	{**1, 3, 7, 9**}, {3, 5, 7, 9}, {0, 3, 5, 7, 9}, {0, 1, 3, 7, 9}
	{**0, 1, 3, 7**}, {0, 5, 7, 9}, {0, 1, 3, 7, 9}, {0, 3, 5, 7, 9}
23	{**1**}, {5}, {0, 1}, {0, 5}
31	{**1, 3, 5**}, {1, 5, 7}, {3, 5, 15}, {3, 11, 15}, {0, 1, 5, 7}, {0, 3, 5, 15}, {0, 1, 3, 5}, {1, 3, 11}, {1, 7, 11}, {0, 1, 7, 11}, {0, 1, 3, 11}, {5, 7, 15}, {0, 5, 7, 15}, {7, 11, 15}, {0, 3, 11, 15}
	{**0, 1, 7, 15**}, {0, 1, 3, 15}, {0, 3, 7, 11}, {0, 5, 11, 15}, {0, 1, 5, 11}, {0, 3, 5, 7}
35	{**1, 3, 5**}, {1, 3, 15}, {1, 3, 5, 15}
	{**1, 5, 7**}, {3, 7, 15} {0, 3, 5, 7, 15}, {0, 1, 5, 7, 15}, {0, 3, 7, 15}, {0, 1, 5, 7}, {3, 5, 7, 15}, {1, 5, 7, 15}
45	{**1, 3, 5**}, {1, 5, 21}, {5, 7, 21}, {3, 5, 7}, {0, 1, 3, 5, 9, 21}, {3, 5, 7, 9, 15}, {1, 3, 5, 9, 21}, {1, 5, 9, 21}, {1, 5, 15, 21}, {0, 1, 5, 9, 21}, {0, 1, 5, 15, 21}, {0, 3, 5, 7, 9, 15}, {1, 3, 5, 9}, {0, 3, 5, 7, 15, 21}, {0, 1, 3, 5, 9}, {3, 5, 7, 15, 21}, {0, 3, 5, 7, 21}, {1, 5, 9, 15, 21}, {3, 5, 7, 21}, {0, 3, 5, 7, 9, 15, 21}, {0, 1, 3, 5}, {5, 7, 9, 21}, {1, 3, 5, 9, 15}, {0, 5, 7, 9, 21}, {0, 1, 5, 9, 15, 21}, {0, 1, 3, 5, 9, 15}, {0, 1, 3, 5, 15, 21}, {3, 5, 7, 9, 15, 21}, {1, 3, 5, 15, 21}, {3, 5, 7, 15}, {0, 3, 5, 7, 15},{0, 1, 3, 5, 9, 15, 21}, {5, 7, 15, 21}, {1, 3, 5, 9, 15, 21},{0, 5, 7, 15, 21}, {0, 3, 5, 7, 9, 21}, {0, 1, 3, 5, 21}, {1, 3, 5, 15}, {3, 5, 7, 9, 21}, {5, 7, 9, 15, 21}, {3, 5, 7, 9}, {0, 1, 3, 5, 15}, {1, 3, 5, 21}, {0, 5, 7, 9, 15, 21}, {0, 1, 5, 21}, {0, 3, 5, 7, 9}, {0, 3, 5, 7}, {5, 7, 21}, {0, 5, 7, 21}
	{**1, 5, 9, 15**}, {5, 7, 9, 15}, {3, 5, 7, 9, 15}, {0, 3, 5, 7, 9, 15}, {1, 5, 9, 15, 21}, {0, 3, 5, 7, 9, 15, 21}, {1, 3, 5, 9, 15}, {0, 1, 5, 9, 15, 21},{0, 5, 7, 9, 15}, {0, 1, 3, 5, 9, 15}, {3, 5, 7, 9, 15, 21}, {0, 1, 3, 5, 9, 15, 21}, {1, 3, 5, 9, 15, 21}, {0, 5, 7, 9, 15, 21}, {0, 1, 5, 9, 15}
49	{**1, 3**}
51	{**1, 3, 9**}, {3, 9, 11}, {3, 9, 19}, {3, 5, 9} {1, 3, 9, 17}, {0, 1, 3, 9}, {3, 5, 9, 17}, {0, 3, 5, 9, 17}, {3, 9, 11, 17}, {0, 3, 9, 11}, {3, 9, 17, 19}, {0, 3, 9, 11, 17}, {0, 3, 9, 17, 19}, {0, 3, 9, 19}, {0, 1, 3, 9, 17}, {0, 3, 5, 9}
55	{**0, 1**},{0, 3}

TABLE VIII

GENERAL ERROR LOCATOR FOR CYCLIC CODES WITH $t = 3$ AND $n = 55$

b	$x_1^2 \cdot (y_1^{475}+y_1^{472}+y_1^{470}+y_1^{469}+y_1^{468}+y_1^{463}+y_1^{462}+y_1^{461}+y_1^{460}+y_1^{458}+y_1^{457}+y_1^{455}+y_1^{454}+y_1^{452}+y_1^{449}+y_1^{448}+y_1^{446}+y_1^{444}+y_1^{443}+y_1^{440}+y_1^{436}+y_1^{434}+y_1^{427}+ y_1^{426}+y_1^{425}+y_1^{424}+y_1^{417}+y_1^{416}+y_1^{413}+y_1^{410}+y_1^{408}+y_1^{405}+y_1^{403}+y_1^{402}+y_1^{401}+y_1^{399}+y_1^{397}+y_1^{395}+y_1^{394}+y_1^{392}+y_1^{388}+y_1^{387}+y_1^{386}+y_1^{384}+y_1^{380}+y_1^{378}+ y_1^{377}+y_1^{376}+y_1^{375}+y_1^{374}+y_1^{372}+y_1^{370}+y_1^{369}+y_1^{368}+y_1^{364}+y_1^{363}+y_1^{361}+y_1^{360}+y_1^{359}+y_1^{358}+y_1^{357}+y_1^{355}+y_1^{350}+y_1^{347}+y_1^{345}+y_1^{343}+y_1^{340}+y_1^{338}+y_1^{336}+ y_1^{334}+y_1^{330}+y_1^{329}+y_1^{327}+y_1^{326}+y_1^{325}+y_1^{324}+y_1^{321}+y_1^{319}+y_1^{318}+y_1^{316}+y_1^{315}+y_1^{312}+y_1^{308}+y_1^{306}+y_1^{305}+y_1^{302}+y_1^{301}+y_1^{296}+y_1^{295}+y_1^{292}+y_1^{290}+y_1^{289}+y_1^{285}+ y_1^{284}+y_1^{278}+y_1^{277}+y_1^{276}+y_1^{275}+y_1^{274}+y_1^{273}+y_1^{272}+y_1^{271}+y_1^{265}+y_1^{261}+y_1^{260}+y_1^{256}+y_1^{255}+y_1^{250}+y_1^{249}+y_1^{248}+y_1^{247}+y_1^{243}+y_1^{242}+y_1^{240}+y_1^{239}+y_1^{235}+y_1^{234}+ y_1^{233}+y_1^{231}+y_1^{230}+y_1^{229}+y_1^{227}+y_1^{225}+y_1^{224}+y_1^{222}+y_1^{221}+y_1^{217}+y_1^{215}+y_1^{213}+y_1^{212}+y_1^{210}+y_1^{209}+y_1^{207}+y_1^{205}+y_1^{203}+y_1^{202}+y_1^{201}+y_1^{200}+y_1^{199}+y_1^{197}+y_1^{195}+ y_1^{189}+y_1^{187}+y_1^{183}+y_1^{182}+y_1^{181}+y_1^{180}+y_1^{179}+y_1^{178}+y_1^{175}+y_1^{172}+y_1^{169}+y_1^{167}+y_1^{165}+y_1^{164}+y_1^{163}+y_1^{160}+y_1^{159}+y_1^{157}+y_1^{155}+y_1^{154}+y_1^{145}+y_1^{141}+y_1^{137}+y_1^{133}+ y_1^{130}+y_1^{129}+y_1^{128}+y_1^{125}+y_1^{123}+y_1^{122}+y_1^{121}+y_1^{117}+y_1^{115}+y_1^{114}+y_1^{113}+y_1^{112}+y_1^{111}+y_1^{110}+y_1^{109}+y_1^{108}+y_1^{107}+y_1^{102}+y_1^{98}+y_1^{96}+y_1^{95}+y_1^{90}+y_1^{89}+y_1^{88}+y_1^{86}+ y_1^{84}+y_1^{83}+y_1^{81}+y_1^{80}+y_1^{78}+y_1^{77}+y_1^{76}+y_1^{74}+y_1^{72}+y_1^{70}+y_1^{68}+y_1^{67}+y_1^{65}+y_1^{63}+y_1^{62}+y_1^{61}+y_1^{55}+y_1^{54}+y_1^{53}+y_1^{52}+y_1^{51}+y_1^{50}+y_1^{49}+y_1^{47}+y_1^{46}+y_1^{45}+y_1^{43}+y_1^{42}+ y_1^{40}+y_1^{38}+y_1^{36}+y_1^{35}+y_1^{33}+y_1^{32}+y_1^{31}+y_1^{30}+y_1^{29}+y_1^{28}+y_1^{24}+y_1^{23}+y_1^{22}+y_1^{21}+y_1^{20}+y_1^{17}+y_1^{15}+y_1^{14}+y_1^{13}+y_1^{11}+y_1^{12}+y_1^{9}+y_1^{7}+y_1^{6}+y_1^{4}+y_1^{3}+y_1^{2}+y_1^{1}+1+ x_2 \cdot (y_1^{26}+y_1^{24}+y_1^{23}+y_1^{13}+y_1^{11}+y_1^{10}+y_1^{8}+y_1^{7}+y_1^{6}+y_1^{3}+y_1))$
c	$x_1^3 \cdot (y_1^{477}+y_1^{476}+y_1^{473}+y_1^{472}+y_1^{470}+y_1^{469}+y_1^{466}+y_1^{463}+y_1^{461}+y_1^{459}+y_1^{458}+y_1^{457}+y_1^{456}+y_1^{453}+y_1^{452}+y_1^{451}+y_1^{450}+y_1^{449}+y_1^{448}+y_1^{447}+y_1^{446}+y_1^{443}+y_1^{441}+ y_1^{440}+y_1^{439}+y_1^{438}+y_1^{436}+y_1^{433}+y_1^{431}+y_1^{428}+y_1^{422}+y_1^{420}+y_1^{419}+y_1^{414}+y_1^{413}+y_1^{410}+y_1^{409}+y_1^{407}+y_1^{406}+y_1^{403}+y_1^{402}+y_1^{400}+y_1^{399}+y_1^{394}+y_1^{391}+y_1^{388}+y_1^{385}+ y_1^{384}+y_1^{383}+y_1^{382}+y_1^{381}+y_1^{379}+y_1^{373}+y_1^{372}+y_1^{368}+y_1^{367}+y_1^{366}+y_1^{363}+y_1^{362}+y_1^{359}+y_1^{358}+y_1^{357}+y_1^{356}+y_1^{354}+y_1^{353}+y_1^{350}+y_1^{349}+y_1^{348}+y_1^{347}+y_1^{344}+y_1^{342}+ y_1^{341}+y_1^{340}+y_1^{339}+y_1^{337}+y_1^{335}+y_1^{334}+y_1^{333}+y_1^{332}+y_1^{331}+y_1^{330}+y_1^{328}+y_1^{325}+y_1^{324}+y_1^{323}+y_1^{322}+y_1^{321}+y_1^{320}+y_1^{319}+y_1^{313}+y_1^{312}+y_1^{310}+y_1^{307}+y_1^{305}+y_1^{304}+ y_1^{303}+y_1^{302}+y_1^{300}+y_1^{295}+y_1^{294}+y_1^{293}+y_1^{292}+y_1^{289}+y_1^{287}+y_1^{286}+y_1^{283}+y_1^{282}+y_1^{280}+y_1^{279}+y_1^{276}+y_1^{274}+y_1^{272}+y_1^{270}+y_1^{269}+y_1^{267}+y_1^{264}+y_1^{263}+y_1^{262}+y_1^{261}+ y_1^{259}+y_1^{256}+y_1^{255}+y_1^{254}+y_1^{253}+y_1^{251}+y_1^{246}+y_1^{244}+y_1^{243}+y_1^{242}+y_1^{241}+y_1^{238}+y_1^{237}+y_1^{235}+y_1^{234}+y_1^{233}+y_1^{231}+y_1^{230}+y_1^{225}+y_1^{222}+y_1^{221}+y_1^{220}+y_1^{212}+y_1^{210}+ y_1^{208}+y_1^{207}+y_1^{206}+y_1^{205}+y_1^{199}+y_1^{198}+y_1^{197}+y_1^{193}+y_1^{191}+y_1^{190}+y_1^{189}+y_1^{188}+y_1^{187}+y_1^{185}+y_1^{184}+y_1^{180}+y_1^{179}+y_1^{177}+y_1^{176}+y_1^{175}+y_1^{174}+y_1^{170}+y_1^{169}+y_1^{167}+ y_1^{166}+y_1^{165}+y_1^{162}+y_1^{160}+y_1^{159}+y_1^{158}+y_1^{156}+y_1^{155}+y_1^{154}+y_1^{148}+y_1^{146}+y_1^{142}+y_1^{141}+y_1^{139}+y_1^{138}+y_1^{137}+y_1^{135}+y_1^{131}+y_1^{130}+y_1^{129}+y_1^{128}+y_1^{126}+y_1^{125}+y_1^{123}+ y_1^{122}+y_1^{120}+y_1^{117}+y_1^{115}+y_1^{112}+y_1^{111}+y_1^{110}+y_1^{108}+y_1^{107}+y_1^{105}+y_1^{103}+y_1^{102}+y_1^{101}+y_1^{99}+y_1^{97}+y_1^{96}+y_1^{92}+y_1^{91}+y_1^{86}+y_1^{85}+y_1^{83}+y_1^{82}+y_1^{81}+y_1^{80}+y_1^{79}+y_1^{78}+ y_1^{77}+y_1^{76}+y_1^{75}+y_1^{74}+y_1^{72}+y_1^{70}+y_1^{68}+y_1^{67}+y_1^{65}+y_1^{62}+y_1^{61}+y_1^{59}+y_1^{58}+y_1^{57}+y_1^{56}+y_1^{55}+y_1^{54}+y_1^{53}+y_1^{52}+y_1^{51}+y_1^{50}+y_1^{49}+y_1^{48}+y_1^{45}+y_1^{44}+y_1^{43}+y_1^{41}+y_1^{40}+ y_1^{39}+y_1^{38}+y_1^{37}+y_1^{36}+y_1^{31}+y_1^{30}+y_1^{27}+y_1^{26}+y_1^{25}+y_1^{24}+y_1^{22}+y_1^{21}+y_1^{20}+y_1^{18}+y_1^{17}+y_1^{16}+y_1^{15}+y_1^{14}+y_1^{13}+y_1^{12}+y_1^{11}+y_1^{9}+y_1^{8}+y_1^{7}+y_1^{6}+y_1^{5}+y_1^{4}+y_1^{3}+y_1^{2}+y_1+ x_2 \cdot (y_1^{24}+y_1^{23}+y_1^{21}+y_1^{19}+y_1^{17}+y_1^{15}+y_1^{11}+y_1^{10}+y_1^{8}+y_1^{7}+y_1^{6}+y_1^{5}+y_1^{4}+y_1^{2}+1))$