
Peer reviewed version

Link to published version (if available):
10.1109/TIT.2017.2692213

Link to publication record in Explore Bristol Research
PDF-document

This is the author accepted manuscript (AAM). The final published version (version of record) is available online via IEEE at http://ieeexplore.ieee.org/document/7894179/. Please refer to any applicable terms of use of the publisher.

University of Bristol - Explore Bristol Research
General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available: http://www.bristol.ac.uk/pure/about/ebr-terms
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 (11) designed for the BCH codes. Although BCH codes can be decoded efficiently, it is known that their decoding performance degrades as the length increases (12). 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 13 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 14 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 15, 16, 17 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 semi-algebraic decoding algorithm for the Golay code of length 23. Orsini and Sala 12 introduced
the general error locator polynomials and presented
an algebraic decoder which permits to determine the
rectifiable 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 to
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
correctable 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 contribu-
tions 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 loca-
tor 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
tor polynomial, justifying theoretically the results
for all cases except three. For the remaining three
cases, the general error locator polynomial can be
computed explicitly.

In Section VI we provide a general error locator polynomials for
all binary cyclic codes with \( t = 3 \) and \( n < 63 \),
that is, with a total of only 952 codes. In Section
these conclusions.

VII we analyze more deeply the
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 \( f, g \in \mathbb{F}_q[x_1, \ldots, x_r] \) for some \( \ell \geq 1 \). When we express 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 usefulness in algebraic properties. We only consider \([n, k, d]_q\) cyclic codes with \((n, q) = 1\), that is \( n \) and \( q \) are coprime. Let \( R = \mathbb{F}_q[y]/(y^n-1) \), each vector \( \mathbf{c} \in (\mathbb{F}_q)^n \) is associated to a polynomial \( c_0 + c_1 y + \cdots + c_{n-1} y^{n-1} \in R \), 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 \( y^n-1 \) over \( \mathbb{F}_q \), and let \( \alpha \) be a primitive \( n \)-th root of unity over \( \mathbb{F}_q \), then it holds \( y^n-1 = \prod_{i=0}^{n-1} (y-\alpha^i) \). For the rest of the paper, we assume that, given \( q \) and \( n \), the primitive root \( \alpha \) 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 \( y^n-1 \) and the set \( \hat{S}_C = \{ i_1, \ldots, i_{n-k} \mid g(\alpha^{i_j}) = 0, \; j = 1, \ldots, n-k \} \) is called the complete defining set of \( C \). Also, the roots of unity \( \{ \alpha^i \mid i \in \hat{S}_C \} \) 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 \((n-k) \times n \) matrix \( H = [h_{ij}]_{i,j} \) over \( \mathbb{F}_q[m] \) such that \( h_{ij} = \alpha^{i_j} \), where \( i_j \in \hat{S}_C \) and \( j = 0, \ldots, n-1 \). This \( H \) is called the standard parity-check matrix. As the complete defining set is partitioned into cyclotomic classes, any subset of \( \hat{S}_C \) 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 \([x] \) 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

\[
\mathbf{e} = (0, \ldots, 0, e_1, 0, \ldots, 0, e_k, 0, \ldots, 0, e_{n-k}, 0, \ldots, 0) \in (\mathbb{F}_q)^n.
\]

We say that the set \( L = \{ i_1, \ldots, i_p \} \subset \{ 0, \ldots, n-1 \} \) is the set of the error positions, the set \( \{ \alpha^i \mid i \in L \} \) is the set of the error locations, and \( \{ s_1, \ldots, s_{n-k} \} \) is the set of the error values.

With this notation, the relation \( H \mathbf{e} = \mathbf{s} \) becomes the well-known equations

\[
s_j = \sum_{h=0}^{\mu} e_h (\alpha^{i_j})^h = \sum_{i \in L} e_i (\alpha^{i_j})^i, \quad 1 \leq j \leq n-k.
\]

(1)

The classical error locator polynomial associated to the error \( \mathbf{e} \) is the polynomial \( \sigma_e(z) = \prod_{i \in L} (1-z \alpha^i) \), i.e. the polynomial having as zeros the inverses of the error locations; whereas the plain error locator polynomial is the polynomial \( \mathbf{L}_e(z) = \prod_{i \in L} (z - \alpha^i) \), 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 \( \sigma_e(z) \) 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 \( \mu \) (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 $Z = (z_1, \ldots, z_n)$ to the error locations, $X = (x_1, \ldots, x_{n-k})$ to the syndromes $\{s_i\}_{1 \leq i \leq n-k}$, and $Y = (y_1, \ldots, y_l)$ 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

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

A first problem here is that obviously we do not know $\mu$ 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

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

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

- $x_j = s_j, \quad 1 \leq j \leq n-k$
- $y_h = e_h, \quad z_h = a^h, \quad 1 \leq h \leq \mu, \quad (3)$
- $y_h = 1, \ z_h = 0, \ \mu + 1 \leq h \leq t.$

\textbf{Remark 1:} We note that, at least in the case of affine-variety codes (constructed by evaluating multivariate polynomials at the rational points 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 \textit{ghost error location}.

If we want to use \([2]\) to decode, once a vector is received we would compute the syndromes and substitute 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 $\mathbb{F}_q$, that the valid error locations are powers of $a$, and thus are $n$-th roots of unity, and that the error values lie in $\mathbb{F}_q$ but are non-zero, so that we can safely consider the following equations

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

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

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

where

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

and they guarantee that the locations (if not zero) are all distinct. We can finally consider the system obtained by putting together \([2], [3], [5]\). This system can be used to unambiguously decode, by evaluating the $X$ variables at the syndromes and obtaining our solution \([5]\), 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 \textit{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 \([2], [3], [5]\) (without evaluating any syndrome) and use advanced commutative algebra to prove the existence of a special polynomial, called \textit{general error locator polynomial}, for every cyclic code (with $(q,n) = 1$).

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

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

\textbf{Remark 2:} The above second property is equivalent to $L(s,z) = z^{\rho(t)}L_\rho(z)$, where $\rho$ is the error associated to syndrome $s$.

Note that the general error locator polynomial $L$ 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 $L$, performs the following steps:

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

This approach is efficient as long as the evaluation of $L$ 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 \( L_C, L_{C'} \) and \( L_{C''} \) denote their respective general error locator polynomials.

If \( C \) is a subcode of \( C' \), then we can assume \( L_C = L_{C'} \).

If \( C \) is equivalent to \( C'' \) via the coordinate permutation function \( \phi : (\mathbb{F}_2)^n \rightarrow (\mathbb{F}_2)^n \), then we can decode \( C \) using \( L_{C''} \) (via \( \phi \)).

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 relatively 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 \( l \in S_C \), then its corresponding syndrome \( x_l \) satisfies the relation \( x_l = \hat{z}_1 + \hat{z}_2 \). The second observation requires more explanation, provided below. Let \( C \) be a code with error capability \( t = 2 \), a correctable syndrome, and \( \hat{z}_1 \) and \( \hat{z}_2 \) the (possibly ghost) error locations corresponding to the syndrome \( s \). Then, by definition we know that

\[
\mathcal{L}(X, z) = z^2 + a z + b = (z - \hat{z}_1)(z - \hat{z}_2),
\]

where \( a, b \in \mathbb{F}_2[X] \), and \( b(s) = \hat{z}_1 \hat{z}_2, a(s) = \hat{z}_1 + \hat{z}_2 \).

Moreover, there are exactly two errors if and only if \( b(s) \neq 0 \), and there is exactly one error if and only if \( b(s) = 0 \) and \( a(s) \neq 0 \) (in this case the error location is \( a(s) \)). Therefore, if \( l \in S_C \), we can write \( a = x_1 \) and so only \( b \) needs to be found, that is,

\[
\mathcal{L}(X, z) = z^2 + x_1 z + b, \quad \text{if } l \in S_C.
\]

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

**Definition 4:** We denote by \( \mathcal{V}^\mu \) the set of syndrome vectors corresponding to \( \mu \) 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, \ldots, \, t \, \text{errors, respectively}), i.e. \( \mathcal{V} = \mathcal{V}^0 \cup \mathcal{V}^1 \cup \cdots \cup \mathcal{V}^t \).

**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 \( \mathcal{V} \subset (\mathbb{F}_2^m)^r \) and \( X = (x_1, \ldots, x_r) \).

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 III 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 in $\mathbb{K}$, that is, $f \in \mathbb{K}[a_1, \ldots, a_N]$ for a variable set $A = \{a_1, \ldots, a_N\}$. We will denote by $|f|$ the number of terms (monomials) of $f$.

**Definition 7:** Let $A = \{a_1, \ldots, a_N\}$ and $B = \{b_1, \ldots, b_M\}$ be two variable sets. Let $\mathbb{K}$ be a field and let $\mathcal{F}$ be a rational function in $\mathbb{K}(A)$. Let $F \in \mathbb{K}[B]$, $f_1, \ldots, f_M \in \mathbb{K}[A]$ and $g_1, \ldots, g_M \in \mathbb{K}[A]$. We say that the triple $(F, \{f_1, \ldots, f_M\}, \{g_1, \ldots, g_M\})$ is a rational representation of $\mathcal{F}$ if

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

We say that the number

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

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

**Theorem 9:** Let $A = \{a_1, \ldots, a_N\}$. If $\mathcal{F}$ is a polynomial, i.e. $\mathcal{F} \in \mathbb{K}[A]$, then

$$ ||\mathcal{F}|| \leq |\mathcal{F}|. $$

Moreover, if $\mathcal{F} = a_1 + a_2$ then $||\mathcal{F}|| = |\mathcal{F}| = 2$.

**Remark 10:** The apparently-trivial result $||a_1 + a_2|| = |a_1 + a_2|$ 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 $||\mathcal{F}|| \leq |\mathcal{F}|$ for any $\mathcal{F}$ but which will give $||a_1 + a_2|| = 1$. 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 \( e = e(q) \) such that, for any cyclic code \( C \) over the field \( \mathbb{F}_q \) with \( n \geq q^2 - 1 \), \( \gcd(n, q) = 1 \), \( 3 \leq d \leq n - 1 \),
\( C \) admits a general error locator polynomial \( L_c \) whose functional density is bounded by
\[
\|L_c\| \leq n^e.
\]
Moreover, for binary codes we have \( e = 3 \), that is, \( e(2) = 3 \).

**Remark 12:** There are some indications that \( e(q) \) may grow with \( q \) and so we do not believe in just one \( e \) 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 \( L_C \) be a general error locator polynomial of \( C \).

**Definition 13:** If \( L_C \in \mathbb{F}_q[x_1, \ldots, x_r] \), then we say that \( L_C \) is sparse if \( \|L_C\| \leq n^3 \).

If Conjecture 11 holds and \( L_C \in \mathbb{F}_q[x_1, \ldots, x_r] \), then we say that \( L_C \) is sparse if \( \|L_C\| \leq n^e \).

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 \( L_C(x_1, \ldots, x_r, z) \) at \( s = (s_1, \ldots, s_r) \).
3) Computation of the roots of \( L_C(s, z) \), which are the valid locations and the ghost locations. The number of valid locations immediately gives \( \mu \).
4) Computations of the error positions from the (valid) locations.
5) Computation of the error values \( e_1, \ldots, e_{\mu} \).

By analyzing the above decoding algorithm, we observe that the main computational cost is the evaluation of the polynomial \( L_C(x_1, \ldots, x_r, z) \) at \( s \), 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 \( L_C(s, z) \) 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 \( n < q^2 - 1 \), 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]
\[
T_{n,t} = \{(\alpha^{i_1}, \ldots, \alpha^{i_r}, \alpha^{l_1}, \ldots, \alpha^{l_l}) \mid 0 \leq i_1 < \cdots < i_r < n,
0 \leq l_j \leq t \} \subset (R_n \cup \{0\})^r.
\]

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 \( z_1, \ldots, z_\mu \) and zero (with multiplicity \( t - \mu \)). By definition of elementary symmetric functions, they can then be expressed as elementary symmetric polynomials in \( \mu \) variables on the \( z_1, \ldots, z_\mu \). 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** (Z. p. 26): Let \( f : (\mathbb{F}_q)^n \to \mathbb{F}_q \). Then \( f \) can be represented by a polynomial in \( \mathbb{F}_q[x_1, \ldots, x_n] \), i.e. there is a polynomial \( P \in \mathbb{F}_q[x_1, \ldots, x_n] \) such that \( P(b_1, \ldots, b_n) = f(b_1, \ldots, b_n) \) for all \((b_1, \ldots, b_n) \in (\mathbb{F}_q)^n\). In particular, the polynomial

\[
\sum_{\alpha \in (\mathbb{F}_q^*)^n} f(\alpha_1, \ldots, \alpha_n) \prod_{i=1}^n (1 - (x_i - \alpha_i)^{q-1})
\]

where \( \alpha = (\alpha_1, \ldots, \alpha_n) \), 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 \( \sigma \in \mathbb{F}_q[y_1, \ldots, y_r] \) be a symmetric function. Then there exists \( a \in \mathbb{F}_q[X] \) such that for \((\bar{x}_1, \ldots, \bar{x}_r) \in \mathcal{V}^\mu \)

\[
a(\bar{x}_1, \ldots, \bar{x}_r) = \sigma(z_1, \ldots, z_\mu, 0, \ldots, 0)
\]

with \( z_1, \ldots, z_\mu \) the error locations corresponding to \( \bar{x}_1, \ldots, \bar{x}_r \).

**Proof:** We claim that the statement is obvious for elementary symmetric functions, as follows. Let \( \sigma_1, \ldots, \sigma_r \) be the elementary symmetric functions in \( \mathbb{F}_q[y_1, \ldots, y_r] \). The existence of a general error locator polynomial for any cyclic code guarantees that, for any \( 1 \leq i \leq t \), for any \( \sigma_\ell \) there is \( a_\ell \in \mathbb{F}_q[x_1, \ldots, x_r] \) such that for \((\bar{x}_1, \ldots, \bar{x}_r) \in \mathcal{V}^\mu \)

\[
a_\ell(\bar{x}_1, \ldots, \bar{x}_r) = \sigma_\ell(z_1, \ldots, z_\mu, 0, \ldots, 0)
\]

with \( z_1, \ldots, z_\mu \) the error locations corresponding to \( \bar{x}_1, \ldots, \bar{x}_r \).

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

\[
\sigma(z_1, \ldots, z_\mu, 0, \ldots, 0) = \sigma(\bar{x}_1, \ldots, \bar{x}_r, 0, \ldots, 0) = \sigma(\bar{x}_1, \ldots, \bar{x}_r, \bar{x}_1, \ldots, \bar{x}_r) = a(\bar{x}_1, \ldots, \bar{x}_r).
\]

**Lemma 18:** Let \( h \in \mathbb{F}_q[X] \) with \( \deg_h h < q \) for all \( l = 1, \ldots, r \) and \( h(x_1, \bar{x}_2, \ldots, \bar{x}_r) = 0 \) for all \((x_1, \ldots, \bar{x}_r) \in (\mathbb{F}_q)^r \) with \( \bar{x}_1 \neq \bar{x}_1 \). If \( l \in \mathbb{F}_q[X] \) and \( g(x_2, \ldots, x_r) \in \mathbb{F}_q[x_2, \ldots, x_r] \) such that

\[
h(X) = x_1 l(x_1, x_2, \ldots, r) + g(x_2, \ldots, x_r).
\]

Then \( h = 0 \) or \( h = (1 - x_1^{q-1}) \cdot g(x_2, \ldots, x_r) \).

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

If \( h(X) \notin \mathbb{F}_q[x_2, \ldots, x_r] \) and \( \deg(h) \leq q - 1 \) then we have that \( h(X) \) is a multiple of \( \mathbb{F}_q[X] \). Note that \( \deg_h h < q \) for all \( i = 1, \ldots, r \).

We claim that \( h = h \). Since the degree w.r.t. each variable \( x_i \) of both the polynomials \( h \) and \( \hat{h} \) is less than \( q \), to prove our claim it is sufficient to show that \( h(X) = \hat{h}(X) \) for all \( X \in (\mathbb{F}_q)^r \). Let us distinguish the cases \( \bar{x}_1 = 0 \) and \( \bar{x}_1 \neq 0 \).

If \( \bar{x}_1 = 0 \), then \( h(X) = 0 \cdot l(0, \bar{x}_2, \ldots, \bar{x}_r) + g(x_2, \ldots, x_r) \) and \( \hat{h}(X) = (1 - 0) \cdot g(x_2, \ldots, x_r) \). So, in this case \( h(X) = \hat{h}(X) \).

Otherwise, \( \bar{x}_1 \neq 0 \). By hypothesis, \( h(X) = 0 \). On the other hand, \( h(X) = (1 - 1) \cdot g(x_2, \ldots, x_r) \). So, in this case \( h(X) = \hat{h}(X) \).

**Theorem 19:** Let \( C \) be a cyclic code over \( \mathbb{F}_q \), with length \( n \) and correction capability \( t \), defined by \( S_C = \{i_1, \ldots, i_r\} \) and let \( x_j \) be the syndrome corresponding to \( i_j \) for \( j \in \{1, \ldots, r\} \). Let \( \sigma \in \mathbb{F}_q[y_1, \ldots, y_r] \) be a symmetric homogeneous function of total degree \( \delta \), with \( \delta \) a multiple of \( i_1 \), and let \( \lambda \) be a divisor of \( n \). Then there exist polynomials \( \sigma \in \mathbb{F}_q[y_1, \ldots, y_r] \), \( g \in \mathbb{F}_q[x_2, \ldots, x_r] \), some non-negative integers \( \delta_2, \ldots, \delta_r \) and some univariate polynomials \( F_{h_0, \ldots, h_r} \in \mathbb{F}_q[y] \) such that for any \( 0 \leq \mu \leq t \), for any \((\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_r) \in \mathcal{V}^\mu \) and the corresponding error locations \( z_1, \ldots, z_\mu \), we have

\[
a(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_r) = \sigma(z_1, \ldots, z_\mu, 0, \ldots, 0)
\]

and

\[
a(X) = \bar{x}_1^{\delta_1} \\
\cdot \sum_{h_0=0}^{\delta_r} \left(\frac{x}{x_1^{\frac{\delta}{\lambda}}} \right)^{h_0} \\
\cdot \sum_{h_1=0}^{\delta_2} \left(\frac{x_2}{x_1^{\frac{\delta}{\lambda}}} \right)^{h_1} \\
\cdot \sum_{h_r=0}^{\delta_r} \left(\frac{x_r}{x_1^{\frac{\delta}{\lambda}}} \right)^{h_r} \\
+ (1 - x_1^{q_0-1}) \cdot g(x_2, \ldots, x_r).
\]

**Proof:** We observe that (6) is immediate by Lemma 16. To prove (7) we first show the case \( \bar{x}_1 = 0 \) and then the general case.
Let us consider the following map $A : \{ X \in \mathcal{V} \mid x_1 \neq 0 \} \rightarrow \mathbb{P}_m^m$ defined by

$$A(x_1, x_2, \ldots, x_r) = \frac{\sigma(z_{1,0}, \ldots, z_{m,0}, 0, \ldots, 0)}{x_1^{\delta_1/m}}, \quad (8)$$

where $(z_1, \ldots, z_m, 0, \ldots, 0)$ is the element of $T_{n,r}$ associated to the syndrome vector $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_r)$. We claim that $A$ depends only on $(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_r/\tilde{x}_1^{\delta_1})$. If our claim is true, then we have

$$A(x_1, x_2, \ldots, x_r) = f(x_1^{\delta_1}, x_2/\tilde{x}_1^{\delta_1}, \ldots, x_r/\tilde{x}_1^{\delta_1}) \quad (9)$$

for a function $f : (\mathbb{P}_m^m)^r \rightarrow \mathbb{P}_m^m$, and so, by Lemma 16, we can view $f$ as a polynomial in $\mathbb{P}_m^m[x_1, \ldots, x_r]$. Since $\mathcal{V} \subset (\mathbb{P}_m^m)^r$, we can also view $A$ as a (non-uniform) polynomial $A(X) \in \mathbb{P}_m^m[X]$. On the other hand, (8) and Lemma 17 show that $A(x_1, x_2, \ldots, x_r) \in \mathbb{P}_m^m[X]$ is a polynomial $a \in \mathbb{P}_m^m$ and so also $A(X)$ can be chosen in $\mathbb{P}_m^m[X]$. Therefore, by (9) we have $f$ can be chosen in $\mathbb{P}_m^m[X]$. Let $\delta_2 = \deg_{s,y}(f), \ldots, \delta_r = \deg_y(f)$. Then, by collecting the powers of $x_i$ in $f$ we have $f = \sum_{s=0}^{\delta_2} \sum_{h=0}^{\delta_1} \cdots \sum_{h=0}^{\delta_r} h_2 F_{h_2, h_1, \ldots, h_r}(x_1), \quad \text{where any } F_{h_2, h_1, \ldots, h_r} \text{ is a univariate polynomial in } \mathbb{P}_m[x_1].$

From (8), (9) and (10) we directly obtain the restriction of (7) to the case $x_1 \neq 0$, considering that $x_1^{\delta_1/m} \equiv 0$, $\delta_1 \leq q^m - 1$ and $\deg F_{h_2, h_1, \ldots, h_r} \leq (q^m - 1)/\delta_1$.

We now prove our claim that gives (9). Let us take $(\tilde{x}_1, \ldots, \tilde{x}_r)$ such that $\tilde{x}_1 = x_1^{\delta_1}$, and $\tilde{x}_j/\tilde{x}_1^{\delta_1} = \tilde{x}_j/\tilde{x}_1^{\delta_1}$, for $j = 2, \ldots, r$. The first relation implies

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

for some $\beta$ such that $\beta^{\delta_1} = 1$. Substituting $\tilde{x}_1$ for $\beta \tilde{x}_1$ in the second relation for $j = 2, \ldots, r$, we obtain

$$\tilde{x}_j = \tilde{x}_j/\beta^{\delta_1} = \tilde{x}_j/\tilde{x}_1^{\delta_1} \quad \Rightarrow \quad \tilde{x}_j = \beta^{\delta_1} \tilde{x}_j. \quad (12)$$

Suppose that $(\tilde{x}_1, \ldots, \tilde{x}_r) \in \mathcal{V}^m$, and $(\tilde{x}_1, \ldots, \tilde{x}_r) \in \mathcal{V}^m$, with $\mu, \mu' \leq t$. From (11), we get $\tilde{y}_1^{\delta_1} + \cdots + \tilde{y}_m^{\delta_1} = \beta\tilde{y}_1^{\delta_1} + \cdots + \beta\tilde{y}_m^{\delta_1}$, for $j = 2, \ldots, r$. Let us now take $\tilde{x}_j = \tilde{x}_j$ for $j = 1, \ldots, t$. Since the $\tilde{x}_j$ are distinct valid error locations (i.e. $\tilde{x}_j = 1$, for $j = 1, \ldots, t$), we have that their syndromes are

$$\tilde{x}_j = \tilde{x}_1^{\delta_1} + \cdots + \tilde{x}_m^{\delta_1} = \tilde{x}_1^{\delta_1} + \cdots + \tilde{x}_m^{\delta_1} = \tilde{x}_j, \quad \text{for } j = 1, \ldots, r.$$
Let $C$ be a cyclic code over $\mathbb{F}_q$ as in Theorem 17. Then the coefficients of the general error locator polynomial can be written in the form given by the previous theorem.

**Corollary 20:** Let $C$ be a code with $t = 2$ defined by $S_C = \{i_1, \ldots, i_t\}$, with $i_1 = 1$, and let $L_C = z^t + x_1z + b$ be a general error locator polynomial for $C$. If $C$ is a primitive code, i.e. $n = q^m - 1$, then $b = x_1^r A$ with $A \in \mathbb{F}_q[x_2/x_1^t, \ldots, x_r/x_1^t]$.

**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 $b = x_1^r A$ with $A \in \mathbb{F}_q[x_2/x_1^t, \ldots, x_r/x_1^t]$. On the other hand, since $C$ is primitive, $x_1^t$ is zero when $x_1$ is zero, and it is 1 when $x_1$ is not zero. So for $\mu \in \{1, 2\}$, $x_1^\mu = 1$ and $b = x_1^r A$ with $A = A|x_1^t = 1$. We claim that $b^r = x_1^r A$ 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 $(1-x_1^{q^m-1})y$ does not appear in the expression of the locator coefficients.

**A. Complexity of the proposed decoding approach**

Now that we have Theorem 17 and Corollary 20, we can estimate the cost of evaluating the polynomial $L_C$ (at the syndrome vector $s$) in the more general case.

First, we observe that we can always choose $\lambda = n$, neglect the cost of computing the values $\frac{s_h}{x_1^h}$ 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 $\mathbb{F}_q[x_1, \ldots, x_r]$ of total degree $\delta$ with a complexity $O(\delta^{r/2})$.

Finally, we can estimate our $\delta$ by observing that, thanks to Corollary 20, we have a bound on the degree of each $z$-coefficient of $L_C$ in any new variable, so that its total degree is easily shown to be at most

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

then, using the method in [27], the evaluation of the $z$-coefficients of $L_C$ at $s$ costs at most

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

(13)

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

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

(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 $r$, let $\gamma$ be an integer $\gamma \geq 1$. Let $C_{\gamma}^n$ 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 $q^\gamma - 1 = O(n^\gamma)$ (and gcd$(n, q) = 1$). For codes in $C_{\gamma}^n$, the complexity (14) of this decoding is at most

$$r \geq 2, \quad O\left(rn^{r/2}\right), \quad r = 1, \quad O\left(n^2 + tn^{r-1}\right).$$

(15)

So, any family $C_{\gamma}^n$ provides a class containing infinite codes which can be decoded in polynomial time, with infinite values of distance and length ($r$ and $\gamma$ 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 $n \leq 63$ 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 $X_j$. In other words,

$$X_0 = x_1^h + z_2^h,$$

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

$$X_{s-\lambda}X_l + X_0 = (z_1z_2)^l(X_{s-\lambda-l}X_{l-t} + X_{s-2l})$$

(16)
Proof: Let us start considering the left-hand side of the equality:

\[ X_{s-1}X_4 + X_s = (z_1^s - z_2^s)(z_1^4 + z_2^4) + z_1^s + z_2^s = z_1^{s-1}z_2^4 + z_1^4z_2^{s-1} = (z_1z_2)^s(z_1^{s-1}z_2^4 + z_1^4z_2^{s-1}) \]

Now we consider the right-hand side:

\[ X_{s-1}X_4 + X_s = (z_1^{s-1} + z_2^{s-1})(z_1^4 + z_2^4) + z_1^2 - z_2^2 = z_1^{s-1}z_2^4 + z_1^4z_2^{s-1} \]

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 \( t = 2 \), \( \{1, l, s, s - \lambda, s - \lambda - 1, s - 2l, s - 2l - 1\} \subset \mathcal{S}_C \) for \( l = 1 \) and \( s - 2l = n = 1 \). Let \( \mu \) be the inverse of \( l \) modulo \( n \). Then the locator polynomial is

\[ \mathcal{L} = z^2 + X_1z + \left( \frac{X_{s-1}X_4 + X_s}{X_{s-1}X_1 + X_{s-2l}} \right)^\mu. \]  \( (17) \)

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

\( \mu = 2 \). From \[ (16) \] we have

\[ (z_1z_2)^s = \frac{X_{s-1}X_4 + X_s}{X_{s-1}X_1 + X_{s-2l}} \]

and from the fact that \( (l, n) = 1 \) we immediately have (passing to the evaluations at \( \text{mathe} \)bfs)

\[ \bar{z}_1\bar{z}_2 = \left( \frac{X_{s-1}X_4 + X_s}{X_{s-1}X_1 + X_{s-2l}} \right)^\mu. \]

However, the numerator or the denominator might be zero (once evaluated at \( s \)). Since the evaluation of their ratio is \( \bar{z}_1\bar{z}_2 \), which is non-zero when \( \mu = 2 \), then in this case \( X_{s-1}X_4 + X_s \neq 0 \) if and only if \( X_{s-1}X_1 + X_{s-2l} \neq 0 \). Therefore, it is enough to prove \( X_{s-1}X_4 + X_s \neq 0 \) (when evaluated at a syndrome vector corresponding to an error of weight 2). We must then prove that

\[ (\bar{z}_1^{\lambda s} + \bar{z}_2^{\lambda s})(\bar{z}_1^4 + \bar{z}_2^4) + (\bar{z}_1^s + \bar{z}_2^s) \neq 0 \]

which is \( (\bar{z}_1\bar{z}_2)^{(\lambda s)}(\bar{z}_1^4 + \bar{z}_2^4) + (\bar{z}_1^s + \bar{z}_2^s) \neq 0 \). Since \( \bar{z}_1\bar{z}_2 \neq 0 \), we must only show that \( \bar{z}_1^{\lambda s} \neq \bar{z}_2^{\lambda s} \). Since \( \bar{z}_1^n = \bar{z}_2^n = 1 \), if by contradiction we have \( \bar{z}_1^{\lambda s} = \bar{z}_2^{\lambda s} \), we would also have \( \bar{z}_1^{(s-2)\lambda n} = \bar{z}_1^{(s-2)\lambda n} \), i.e. \( \bar{z}_1 = \bar{z}_2 \), which is impossible (\( \mu = 2 \)).

\( \mu = 1 \). When there is only one error, the evaluation of \( X_{s-1}X_4 + X_s \) zero, as well as the evaluation of \( X_{s-1}X_1 + X_{s-2l} \), and so by our convention their ratio is zero, that is, \( b(s) = 0 \) when \( \mu = 1 \), as claimed.

The code family defined in the previous corollary contains nearly all codes with \( t = 2 \) and \( s \leq n < 105 \), 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-\( \lambda \) 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-\( \lambda \) code, then

- either its defining set is one of the following list
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 \( 15 \leq n < 63 \) \((n \text{ odd})\). Then there are only three cases.

1) Either \( C \) is one of the following:

\[
\begin{align*}
n &= 15, S_C = \{1, 3, 5\}, n = 21, S_C = \{1, 3, 5\}, S_C = \{1, 3, 5, 7, 9\}, S_C = \{0, 1, 3, 7\}; \\
n &= 23, S_C = \{1\}, n = 31, S_C = \{1, 3, 5\}, S_C = \{0, 1, 7, 15\}; \\
n &= 35, S_C = \{1, 3, 5\}, S_C = \{1, 5, 7\}, n = 45, S_C = \{1, 3, 5, 7, 9, 15\}; \\
n &= 49, S_C = \{1, 3\}, n = 51, S_C = \{1, 3, 9\}, n = 55, S_C = \{0, 1\}.
\end{align*}
\]

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. 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 shown in the following theorem.

Theorem 28: Let \( C \) be an \([n, k, d]\) code with \( d \in \{7, 8\} \) and \( 15 \leq n < 63 \) \((n \text{ odd})\). 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 \( \{1, 3, 2^i + 2^j, 2^i - 2^j, 2^i + 2^{i+1}\} \) 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:

\[
\begin{align*}
n &= 21, S_C = \{0, 1, 3, 7\}; \\
n &= 51, S_C = \{1, 3, 9\}; \\
n &= 55, S_C = \{0, 1\}.
\end{align*}
\]

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 \( t = 3 \), \( s \) a correctable syndrome and \( \bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3 \) the error locations. Then \( \mathcal{L}(X, z) = z^3 + a \bar{z}^2 + bz + c \), where \( a, b, c \in \mathbb{F}_2[X] \), and \( a(s) = \bar{\xi}_1 + \bar{\xi}_2 + \bar{\xi}_3 \), \( b(s) = \bar{\xi}_1 \bar{\xi}_2 + \bar{\xi}_1 \bar{\xi}_3 + \bar{\xi}_2 \bar{\xi}_3 \), \( c(s) = \bar{\xi}_1 \bar{\xi}_2 \bar{\xi}_3 \). Moreover, there are three errors if and only if \( c(s) \neq 0 \), there are two errors if and only if \( c(s) = 0 \) and \( b(s) \neq 0 \), and there is one error if and only if \( c(s) = b(s) = 0 \) and \( a(s) \neq 0 \). Note that from the previous corollary any code with \( t = 3 \) and \( n < 63 \) is equivalent to a code in 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 \( z_j^i x_m^i, i, j \in \{1, 2, 3\} \), and \( \Sigma_{l,m,r} \) denotes all the six terms of the type \( z_j^i z_k^i \), \( i, j, k \in \{1, 2, 3\} \).

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

\[
\mathcal{L}(X, z) = z^3 + x_1 z^2 + bz + c \quad \text{with} \quad b = \frac{(x_1^2 x_3 + x_5)}{(x_1^3 + x_5)}, \quad c = \frac{(x_1^2 x_3 + x_5^2 + x_2^2 + x_1 x_5)}{(x_1^3 + x_5)}.
\]

Proof: It is 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

\[
\mathcal{L}(X, z) = z^3 + x_1 z^2 + bz + c \quad \text{with} \quad b = \frac{x_i U + x_{i+1} V}{W}, \quad c = \frac{x_{i+1} U + x_{i+2} V}{W}
\]

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, \( e \) has weight three, and let \( b \) be its syndrome vector. It is a simple computation to show, using the Newton’s identities

\[
\begin{align*}
x_{i+4} &= x_1 x_{i+3} + b x_{i+2} + c x_{i+1} \\
x_{i+3} &= x_1 x_{i+2} + b x_i + c x_i
\end{align*}
\]

that \( b = \frac{x_i U + x_{i+1} V}{W}, \ c = \frac{x_{i+1} U + x_{i+2} V}{W} \), 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 \( \mu = 3 \), \( \mathcal{L}(s, z) \) is the error locator polynomial for \( C \).
Let us show that it is actually a general error locator polynomial for $C$. We have that
\[ x_i U + x_{i+1} V = (c_i^1 + z_i^1 + z_i^2) \cdot \left( \frac{z_i^{i+1} + z_i^{i+2} + z_i^{i+3}}{z_i^{i+1}} \right) \cdot \left( z_i^{i+1} + z_i^{i+2} + z_i^{i+3} + (z_i + z_2 + z_3) z_i^{i+3} \right) + (z_i^{i+2} + z_i^{i+2} + z_i^{i+3}) \cdot \left( z_i^{i+1} + z_i^{i+2} + z_i^{i+3} + (z_i + z_2 + z_3) z_i^{i+3} \right) \]
and
\[ x_i U + x_{i+1} V = \left( z_i^1 + z_i^2 + z_i^3 \right) \cdot \left( \frac{z_i^{i+1} + z_i^{i+2} + z_i^{i+3}}{z_i^{i+1}} \right) \cdot \left( z_i^{i+1} + z_i^{i+2} + z_i^{i+3} + (z_i + z_2 + z_3) z_i^{i+3} \right) \]

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

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

**Lemma 32:** Let $s_k = \Sigma_{i=1}^{\lfloor k \rfloor} \ldots \lfloor k \rfloor \geq 0 \geq k$ be the $k$th elementary symmetric polynomial in the variables $z_1, z_2, z_3$ over $\mathbb{F}_2$, where $k \in \{1, 2, 3\}$, and let $x_k = \Sigma_{i=1}^{\lfloor k \rfloor} \notin \mathbb{F}_2 [z_1, z_2, z_3]$ be the power sum polynomial of degree $k$, with $h \geq 0$. Then, for $i \geq 0$ and $j \geq i+2$,
\[ x_1^{2i+1} + x_2 x_3 + x_1^{2j+2} = x_1^{2i+1} + x_2 x_3 + x_1^{2j+2} \]

**Proof:** Since the syndrome $x_1$ is a known syndrome, that is, $i \in S_1$, we have that $a = x_1$. From the Newton identity $c = x^1_i + x_3 + x_1 b$ we get that
\[ c^{2i} = x_1^{2i+1} + x_3^{2i+1} + x_1 b^{2i} \]

On the other hand, by the previous lemma, we have that
\[ x_1^{2i+2} + x_2 x_3 + b^{2i} x_3^{2i+1} = b^{2i} x_3^{2i+1} + x_2 x_1^{2i+1} + c^{2i} x_3^{2i+1} \]

Taking into account (20) and (21), a few computations lead to the equalities $b^{2i} = \left( \frac{x_3^{2i+1} U + V}{W} \right)$ and $c^{2i} = \left( \frac{x_3^{2i+1} U + V}{W} \right)$.

Finally, let us consider the codes in 4 of Theorem 28 in Elia presents an algebraic decoding for the (23, 12, 7) Golay code providing the error locator polynomials for $\mu$ errors, for $\mu$ from one to three. In 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 $t = 3$, $S_1$ 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 $S_1$ containing $\{1, 3, 9\}$ with $(n, 3) = 1$. Then $L(x, z) = z^3 + x_1 z^2 + b x + c$ with
\[ b = (x_1^2 + D^t) h, \quad c = (x_3 + x_1 D^t) h, \]
where $D = \left( \frac{x_3 + x_1}{x_3 + x_1} \right)$ + (x_3 + x_1) \frac{x_3 + x_1}{x_3 + x_1}$, $l = 3$ and $t'$ is the inverse of $l$ modulo $2^{m-1}$ with $F_{2^m}$ the splitting field of $x^1 - 1$ over $F_2$.

**Proof:** Since $i \in S_1$, we have that $a = x_1$. From the following Newton identities
\[
\begin{align*}
x_9 &= x_1 x_8 + b x_7 + c x_6 \\
x_7 &= x_1 x_6 + b x_5 + c x_4 \\
x_5 &= x_1 x_4 + b x_3 + c x_2 \\
x_3 &= x_1 x_2 + b x_1 + c
\end{align*}
\]
using the equalities $x_6 = x_3^2$, and $x_2 = x_1^{2i}$ for $i \geq 0$, we get
\[ \left( \frac{x_9 + x_9}{x_3 + x_4} \right) + (x_1^3 + x_3)^2 = (b + x_1^3)^3 \]  
(22)

So \( b = x_1^3 + D \). From \( x_3 = x_1 x_2 + b x_1 + c \), we find \( c = x_3 + x_3 D \). Let us prove that \( L \) is a general error locator polynomial. By Lemma 1 and Lemma 2 in \[15\], 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 \( n \leq 55 \). 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 \( n = 51 \), \( x_1, x_2, x_3, x_4, x_5 \) 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 inspection 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 \( L \) costs \( O(1) \), and so the decoding process costs \( O(n^2) \). For \( t = 2 \) and \( t = 3 \) exhaustive searching method costs, respectively, \( O(n^3) \) and \( O(n^2) \). 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 foreseen 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 \( n^3 = 41^3 = 68921 \). Observe also that the evaluation of
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 IV 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.

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 V we report the maximum number of terms for the coefficients of the general error locator for these two codes.

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 $|L| = 232 < 1331 = 11^3$. For the second locator we have $|L| = 15204$ which compared to $n^t = 23^3 = 12167$ 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 $|L|$ 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 $x_i$ and the number of terms of this polynomial is less than $\lfloor \frac{\sum_{r=1}^{t} 2^r \binom{q}{r}}{n} \rfloor$.

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 $x_i$ and the number of terms of this polynomial is less than $\lfloor \frac{\sum_{r=1}^{t} q^{r-1} \binom{q}{r}}{n} \rfloor$.

**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 $1 + \sum_{r=1}^{t} (q-1)^r \binom{q}{r}$.

With $q$ fixed, the codes covered by the previous theorem are actually the component of our families $C_{1,y}$ for $y \geq 1$. Depending on the actual considered length we will have the correct determination of $y$, 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 $C_{1,y}$ is

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

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

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

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

**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

### Table I

<table>
<thead>
<tr>
<th>$n$</th>
<th>Binary cyclic codes with $i = 3$ and length &lt; 121 covered by Theorem 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>21</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>23</td>
<td>$(0, 1, 7, 15)$</td>
</tr>
<tr>
<td>31</td>
<td>$(0, 1, 7, 15)$</td>
</tr>
<tr>
<td>35</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>63</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>65</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>70</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>77</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>89</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>93</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>105</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>117</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>$n$</th>
<th>Binary cyclic codes with $i = 3$ and length &lt; 121 covered by Theorem 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>21</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>23</td>
<td>$(0, 1, 7, 15)$</td>
</tr>
<tr>
<td>31</td>
<td>$(0, 1, 7, 15)$</td>
</tr>
<tr>
<td>35</td>
<td>$(1, 3, 5)$</td>
</tr>
<tr>
<td>63</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>65</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>70</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>77</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>89</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>93</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>105</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
<tr>
<td>117</td>
<td>$(1, 3, 11, 23, 27)$</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>$n$</th>
<th>Binary cyclic code with $i = 3$ and $n &lt; 55$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>$(1, 3, 5)$ (50)</td>
</tr>
<tr>
<td>21</td>
<td>$(1, 3, 5)$ (50)</td>
</tr>
<tr>
<td>23</td>
<td>$(0, 1, 7, 15)$ (10)</td>
</tr>
<tr>
<td>31</td>
<td>$(0, 1, 7, 15)$ (10)</td>
</tr>
<tr>
<td>35</td>
<td>$(1, 3, 5)$ (50)</td>
</tr>
<tr>
<td>63</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>65</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>70</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>77</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>89</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>93</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>105</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
<tr>
<td>117</td>
<td>$(1, 3, 11, 23, 27)$ (10)</td>
</tr>
</tbody>
</table>

---

### Notes

1. **Table I** contains all binary cyclic codes with $i = 3$ and length less than 121 covered by Theorem 21.
2. **Table II** includes additional binary cyclic codes with $i = 3$ and length less than 121 covered by Theorem 23.
3. **Table III** lists all binary cyclic codes with $i = 3$ and $n < 55$.

---

**References**: 
- Theorem 21
- Theorem 23
- Various mathematical formulae and calculations related to binary cyclic codes.
### TABLE VII
Binary cyclic codes with $t = 3$ and $n < 63$

<table>
<thead>
<tr>
<th>$n$</th>
<th>Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>${3, 5}, {3, 5, 7}, {0, 3, 5, 7}, {0, 1, 3, 5}$</td>
</tr>
<tr>
<td>21</td>
<td>${1, 3, 5}, {1, 3, 9}, {1, 3, 5, 9}$</td>
</tr>
</tbody>
</table>
| 23     | $\{1, 3, 5, 9\}, \{3, 5, 7, 9\}, \{0, 3, 5, 7, 9\}, \{0, 1, 3, 7, 9\}$,  
|         | $\{0, 1, 3, 7\}, \{0, 3, 5, 7, 9\}, \{0, 3, 5, 7, 9\}$             |
| 31     | $\{1, 3, 5, 7, 15\}, \{3, 5, 15\}, \{3, 11, 15\}, \{0, 1, 5, 7, 15\}$,  
|         | $\{0, 1, 3, 5\}, \{1, 7, 11\}, \{0, 1, 3, 11\}, \{5, 7, 15\}$,  
|         | $\{0, 5, 7, 15\}, \{7, 11, 15\}, \{0, 3, 11, 15\}$                 |
| 35     | $\{1, 3, 5\}, \{1, 3, 15\}, \{1, 3, 5, 15\}$                       |
| 45     | $\{1, 3, 5, 15\}, \{1, 5, 21\}, \{3, 5, 7, 21\}, \{3, 5, 9, 21\}$,  
|         | $\{3, 5, 7, 9, 15\}, \{1, 3, 5, 9, 21\}, \{1, 3, 9, 21\}$,  
|         | $\{1, 5, 15\}, \{0, 1, 5, 9, 21\}, \{0, 3, 5, 7, 15\}, \{1, 3, 5, 9\}$,  
|         | $\{0, 3, 5, 7, 15\}, \{3, 5, 7, 15\}, \{3, 5, 7, 21\}$,  
|         | $\{0, 1, 3, 5, 9, 15\}, \{0, 1, 3, 5, 9, 21\}, \{0, 1, 3, 5, 9, 15\}, \{0, 1, 3, 5, 9, 21\}$,  
|         | $\{0, 1, 3, 5, 15, 21\}, \{3, 5, 7, 9, 15\}, \{3, 5, 7, 15\}, \{3, 5, 7, 9\}, \{0, 1, 3, 5\}$,  
|         | $\{1, 3, 5, 9, 15\}, \{1, 3, 5, 9, 21\}$, $\{1, 3, 5, 9\}$,  
|         | $\{5, 7, 15\}, \{0, 5, 7, 15\}, \{0, 3, 5, 7, 15\}, \{0, 1, 3, 5\}$,  
|         | $\{0, 1, 3, 5, 15\}, \{0, 3, 5, 7, 15\}$, $\{5, 7, 21\}$, $\{0, 5, 7\}$,  |
| 49     | $\{1, 3, 9\}, \{3, 9, 11\}, \{3, 9, 19\}, \{3, 5, 9\}, \{1, 3, 9\}$,  
|         | $\{3, 5, 9, 17\}, \{0, 1, 3, 9\}, \{3, 5, 9, 17\}, \{3, 9, 11, 17\}$,  
|         | $\{0, 3, 9, 11, 17\}, \{0, 3, 9, 17, 19\}, \{0, 3, 9, 19\}, \{0, 1, 3, 9, 17\}$,  
|         | $\{0, 3, 9\}$                                                       |
| 51     | $\{0, 1\}, \{0, 3\}$                                               |
| 55     |                                                                      |

### TABLE VIII
General error locator for cyclic codes with $t = 3$ and $n = 55$

```
<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```