Generalized Berlekamp–Massey Decoding of Algebraic-Geometric Codes up to Half the Feng–Rao Bound

Shojiro Sakata, *Senior Member, IEEE,* Helge Elbrønd Jensen, and Tom Høholdt, *Member, IEEE*

***Abstract*—We treat a general class of algebraic-geometric codes and show how to decode these up to half the Feng–Rao bound, using an extension and modification of the Sakata algorithm. The Sakata algorithm is a generalization to N dimensions of the classical Berlekamp–Massey algorithm.**

***Index Terms*—Decoding, algebraic-geometric codes.**

I. Introduction

EFFICIENT decoding of BCH- and Reed–Solomon codes can be done by using the Berlekamp–Massey algorithm [1], and it is natural to try to use the extension to N dimensions of Sakata [2] to decode algebraic-geometric codes. For codes from regular plane curves this was done in [3] and using the Feng–Rao majority scheme from [4], the procedure was extended in [5] and [6]. For a class of space curves the method of [3] was generalized in [7], but here the algorithm does not correct all errors up to half the minimum distance.

In this paper we treat a general class of algebraic-geometric codes, the so-called one-point codes, and show how to decode these up to half the Feng–Rao bound, using an extension and modification of the Sakata algorithm. The complexity of the decoding algorithm is also calculated.

The paper is organized as follows. In Section II we present the codes, and the special choice of a basis for the spaces needed, so that the decoding problem can be solved using the Sakata algorithm. In Section III we present the decoding algorithm, and Section IV contains calculation of the complexity.

II. The Codes

We assume some familarity with the basic concepts of algebraic-geometric codes, e.g., [8] or [9].

Let $P_1, P_2, \cdots, P_n, P_\infty$ be a set of $\mathbb{F}_q$-rational points on a nonsingular, irreducible curve χ of genus g defined over $\mathbb{F}_q$. We consider an algebraic-geometric code C of type $C_L(D, G)^\perp = C_\Omega(D, G)$, where

$$D = P_1 + P_2 + \cdots + P_n \qquad G = mP_\infty.$$

The code C has length n, and for any $\boldsymbol{y} \in \mathbb{F}_q^n$ we have

$$\boldsymbol{y} \in C \Leftrightarrow \sum_{j=1}^{n} f(P_j) y_j = 0, \qquad \text{for all } f \in L(mP_\infty). \quad (1)$$

When $2g - 2 < m < n$, the dimension of C is $k = n - m + g - 1$, and the minimum distance is lower-bounded by $d^* = m - 2g + 2$. When $m < 4g - 2$ this estimate is improved by the Feng–Rao bound d_{FR}, which we define later. One has $d_{\mathrm{FR}} \geq d^*$ with equality when $m \geq 4g - 2$.

Recall that a number o_i is a *nongap* for P_∞ if $L(o_i P_\infty) \neq L((o_i - 1)P_\infty)$. In this case, there exists a function $\varphi_i \in L(o_i P_\infty) \backslash L((o_i - 1)P_\infty)$, which means that φ_i has a pole of order o_i at P_∞ and no other poles. It is well known that the nongaps satisfy

$$0 = o_1 < o_2 < \cdots < o_g < o_{g+1} = 2g$$

$$o_i = i + g - 1, \quad \text{for } i \geq g + 1.$$

The functions φ_i, $i = 1, 2, \cdots, m - g + 1$ provide a basis for the space $L(mP_\infty)$.

The nongap sequence—that is, the possible pole orders at P_∞—forms a semigroup under addition. Let $a_1, a_2, \cdots, a_N$ be a minimal set of generators for this semigroup, and let ψ_j be a function with pole order a_j at P_∞ and no other poles. To any vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)$ of nonnegative integers corresponds the function

$$f_{\underline{\alpha}} = \prod_{j=1}^{N} \psi_j^{\alpha_j}. \quad (2)$$

This function has a pole only at P_∞. The order of this pole is denoted $O(\boldsymbol{\alpha})$, and we have

$$O(\boldsymbol{\alpha}) = \sum_{j=1}^{N} \alpha_j a_j. \quad (3)$$

The set of functions $f_{\underline{\alpha}}$ where $O(\boldsymbol{\alpha}) \leq m$, span the space $L(mP_\infty)$. These functions are, however, not independent, since if $O(\boldsymbol{\alpha}) = O(\boldsymbol{\alpha}')$ then

$$f_{\underline{\alpha}} = c f_{\underline{\alpha}'} + g, \qquad \text{where } c \in \mathbb{F}_q \quad \text{and} \quad O_{P_\infty}(g) < O(\boldsymbol{\alpha}). \quad (4)$$

Manuscript received September 13, 1994; revised May 22, 1995. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, June 27–July 1, 1994.

S. Sakata is with the Department of Computer Science and Information Mathematics, Faculty of Electro-Communications, The University of Electro-Communications, Chofu-ga-oka 151, Chofu-shi, Tokyo, Japan.

H. E. Jensen and T. Høholdt are with the Mathematical Institute, Technical University of Denmark, DK-2800 Lyngby, Denmark.

IEEE Log Number 9415548.

An important concept in decoding is the *syndrome* of a vector. Let $\boldsymbol{y} \in \mathbb{F}_q^n$. With each function $f_{\underline{\alpha}}$ we associate the syndrome $S_{\underline{\alpha}}(\boldsymbol{y})$ defined by

$$S_{\underline{\alpha}}(\boldsymbol{y}) = \sum_{j=1}^{n} f_{\underline{\alpha}}(P_j) y_j. \tag{5}$$

It follows from (1) and the remarks above that

$$\boldsymbol{y} \in C \Leftrightarrow S_{\underline{\alpha}}(\boldsymbol{y}) = 0, \qquad \text{for all } \boldsymbol{\alpha} \text{ with } O(\boldsymbol{\alpha}) \le m.$$

In the decoding situation we receive a vector $\boldsymbol{r}$, which is the sum of an unknown codeword $\boldsymbol{c}$ and an unknown error vector $\boldsymbol{e}$, that is $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. We therefore have $S_{\underline{\alpha}}(\boldsymbol{e}) = S_{\underline{\alpha}}(\boldsymbol{r})$ for all $\boldsymbol{\alpha}$ with $O(\boldsymbol{\alpha}) \le m$, and the decoding problem is then, from the known terms $S_{\underline{\alpha}}(\boldsymbol{e})$, to find the vector $\boldsymbol{e}$.

One way to do this, reasonably efficiently ([3]), is to consider recursions among the syndromes and from such equations determine an error locator, that is, a function which points out the positions where the coordinates in $\boldsymbol{e}$ are different from zero. This procedure, however, does not correct errors up to half the minimum distance. Another approach, which we use in this paper, is first to determine *all* syndromes $S_{\underline{\alpha}}(\boldsymbol{e})$, $0 \le \alpha_j \le q-1$, $i = 1, \cdots, N$. How this is done is explained in the next section, but let us here just suppose that we know all the syndromes. Then for each P_l we can form the sum

$$\sum_{\underline{\alpha}} S_{\underline{\alpha}}(\boldsymbol{e}) \prod_{s=1}^{N} \psi_s^{-\alpha_s}(P_l) \tag{6}$$

where the summation is over all vectors $\boldsymbol{\alpha}$ with $1 \le \alpha_s \le q-1$, $s = 1, \cdots, N$. By inserting (5) and (2) we get

$$\sum_{\underline{\alpha}} \sum_{j=1}^{n} \prod_{s=1}^{N} \psi_s^{\alpha_s}(P_j) \psi_s^{-\alpha_s}(P_l) e_j = \sum_{j=1}^{n} e_j \prod_{s=1}^{N} \sum_{\underline{\alpha}} \left[\frac{\psi_s(P_j)}{\psi_s(P_l)} \right]^{\alpha_s} = (-1)^N e_l \tag{7}$$

and hence e_l can be calculated. The last equality in (7) needs two comments. First, if $\psi_s(P_j) \ne \psi_s(P_l)$, then

$$\sum_{\alpha_s=1}^{q-1} \left[\frac{\psi_s(P_j)}{\psi_s(P_l)} \right]^{\alpha_s} = 0.$$

If $j \ne l$, then for at least one s we have $\psi_s(P_j) \ne \psi_s(P_l)$. Because otherwise $f_{\underline{\alpha}}(P_j) = f_{\underline{\alpha}}(P_l)$ for each α, and consequently there is a codeword with weight 2, and we will not consider such codes.

The second remark is that in the calculations we have supposed that $\psi_s(P_l) \ne 0$ for all $s = 1, \cdots, N$. If this is not the case the calculations should be slightly modified, which is done in Appendix I. In any case, knowing all syndromes we can find the error vector.

III. The Algorithm

The decoding algorithm is a modification of the Sakata algorithm [2], and in the following we will assume some familiarity with this algorithm and use some of the results from [2].

The algorithm takes as input an N-dimensional array of elements from $\mathbb{F}_q$, and produces as output a so-called minimal set of polynomials corresponding to linear recurring relations satisfied by the array. In order to describe the algorithm we have to introduce some notation from [2].

Let Σ_0 be defined as the set of all N-tuples of nonnegative integers, that is $\Sigma_0 = \mathbb{Z}_0^N$. For any subset $\Gamma \subseteq \Sigma_0$ an *array* over the field K is a mapping u: $\Gamma \to K$, which is written $u = (u_{\underline{x}})$ where $u_{\underline{x}} = u(\boldsymbol{x})$, $\boldsymbol{x} \in \Gamma$ is the "value" of u at the point $\boldsymbol{x}$.

We need a total ordering of the points of Σ_0, and here we choose—and that is an important choice—the ordering corresponding to the code described in Section II. This means that we define

$$\boldsymbol{p} = (p_1, \cdots, p_N) <_T \boldsymbol{q} = (q_1, \cdots, q_N)$$

if

$$O(\boldsymbol{p}) < O(\boldsymbol{q}) \quad \text{or} \quad O(\boldsymbol{p}) = O(\boldsymbol{q}) \qquad \text{and} \qquad p_l < q_l;\ p_i = q_i,\ i < l.$$

This, in turn, also gives an ordering of the functions $f_{\underline{\alpha}}$ and the syndromes $S_{\underline{\alpha}}(\boldsymbol{e})$. It should be mentioned that Sakatas algorithm works for any admissible ordering of Σ_0.

It is convenient to represent linear recurring relations by N-variate polynomials $\sigma \in \mathbb{F}_q[x] = \mathbb{F}_q[\psi_1, \psi_2, \cdots, \psi_N]$. Any such polynomial can be written as

$$\sigma = \sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} x^{\underline{q}} \tag{8}$$

where Γ_σ is a finite subset of Σ_0, such that $\sigma_{\underline{q}} \ne 0$ for $\boldsymbol{q} \in \Gamma_\sigma$. The maximum element in Γ_σ with respect to the total order $<_T$ is called the *degree* of σ and is written $\mathrm{Deg}\,(\sigma)$.

A polynomial σ is said to be valid at a point $\boldsymbol{p}$ for an array u, if $\boldsymbol{p} \ge \boldsymbol{s} = \mathrm{Deg}\,(\sigma)$ and

$$\sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} u_{\underline{q}+\underline{p}-\underline{s}} = 0 \tag{9}$$

Here $\ge$ is the natural partial order on Σ_0 defined by $\boldsymbol{p} \ge \boldsymbol{q}$ iff $\boldsymbol{p}_i \ge \boldsymbol{q}_i$ for all $i = 1, \cdots, N$. Moreover, here and in the following we assume that Γ is of the form

$$\Gamma = \{\boldsymbol{x} \in \Sigma_0 \mid \boldsymbol{x} <_T \boldsymbol{l}\}$$

and write $u = u^{\underline{l}}$ for the corresponding array.

A polynomial σ is said to be valid for the array $u = u^{\underline{l}}$ iff (9) holds for all points $\boldsymbol{p}$ where $\boldsymbol{s} \le \boldsymbol{p} <_T \boldsymbol{l}$.

To understand the whole setup better, let us consider the decoding situation where we look at the array of known syndromes $S_{\underline{\alpha}}(\boldsymbol{e})$ where $O(\boldsymbol{\alpha}) \le m$.

Inserting (5) in (9) and using (2) we get

$$\sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} S_{\underline{q}+\underline{p}-\underline{s}} = \sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} \sum_{j \in E} f_{\underline{q}+\underline{p}-\underline{s}}(P_j) e_j = \sum_{j \in E} e_j f_{\underline{p}-\underline{s}}(P_j) \sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} f_{\underline{q}}(P_j) \tag{10}$$

where $E = \{j_1, \cdots, j_t\}$ denotes the positions for which the error vector is $\neq 0$. It follows from this that if the function

$$f = \sum_{\underline{q} \in \Gamma_\sigma} \sigma_{\underline{q}} f_{\underline{q}} \tag{11}$$

is zero at all error points $P_{j_1}, \cdots, P_{j_t}$, then the polynomial σ in (8) satisfies all possible recurring relations (9) for that polynomial and the array considered.

In the ordinary decoding procedures for AG codes the basic idea is to find a function like (11) with the error positions as zeros, by considering all possible equations (9) and take the "smallest" nonzero solution. It turns out ([3]) that in this way you can only be sure to get an error locator if the number of errors is somewhat smaller than half the minimum distance. However, using recurring relations like (9), it is possible—for errors up to half the minimum distance—to predict the value of the unknown syndromes and then correct all the errors using (7). How this is done will be explained in the following.

Let us return to the general situation where we consider an array $u = u^{\underline{l}}$. The set of valid polynomials for this array is denoted VALPOL(u).

For an array u a *minimal polynomial set* is a finite subset F of $\mathbb{F}_q[x]$ such that

1)

$$F \subseteq \text{VALPOL}\,(u)$$

2) Let

$$S = \{s = \text{Deg}\,(\sigma) \mid \sigma \in F\}.$$

Then for any s and t

$$s \in S \wedge s < t \Rightarrow t \notin S.$$

3) Let

$$\Delta = \Delta(F) = \sum\nolimits_0 \Big\backslash \bigcup_{s \in S} \{t \in \Sigma_0 \mid s \leq t\}.$$

Then there exists no polynomial $g \in \text{VALPOL}\,(u)$ such that $\text{Deg}\,(g) \in \Delta$.

It follows that the word minimal in the term minimal polynomial set refers to the degrees of the polynomials in the set.

The algorithm of Sakata takes as input the elements of an array $u = u^{\underline{l}}$ and produces as output a minimal polynomial set for the array. The algorithm considers the elements of the array step by step. At each step, one has a minimal polynomial set F for the part of the array seen so far. When the next element of the array is taken into consideration, the algorithm starts to check if the polynomials $\sigma \in F$ are still valid for the new array. If this is not the case, they are updated and a new minimal polynomial set and a new Δ-set is produced.

The details of the algorithm can be found in [2]. Actually, we need a small modification of the algorithm, but before we explain this, we will emphasize the following result ([2, Lemma 2]), which is essential for the whole process.

Lemma 1: Let $\text{Deg}\,(\sigma) = \boldsymbol{s}$. If $\sigma \in \text{VALPOL}\,(u^{\underline{q}})$ and $\sigma \notin \text{VALPOL}\,(u^{\underline{q}+1})$, then there exists no polynomial $g \in \text{VALPOL}\,(u^{\underline{q}+1})$, such that $\text{Deg}\,(g) \leq \boldsymbol{q} - \boldsymbol{s}$.

Here $\boldsymbol{q}+1$ denotes the next point of $\boldsymbol{q}$ with respect to the total order.

Let us next go to the decoding situation where the array consists of the known syndromes $S_{\underline{\alpha}}(\boldsymbol{e})$. With t we denote the number of errors, and $P_{j_1}, \cdots, P_{j_t}$ corresponds to the positions where the errors occur. We assume that all syndromes $S_{\underline{\alpha}}(\boldsymbol{e})$, where $O(\boldsymbol{\alpha}) \leq m'$, are known, and we want to find $S_{\underline{\alpha}}(\boldsymbol{e})$ for $O(\boldsymbol{\alpha}) = m'$. Here $m' > m$. There can be many syndromes corresponding to the same pole order. But if $O(\boldsymbol{\alpha}) = O(\boldsymbol{\alpha}')$, then we have an identity (4) between $f_{\underline{\alpha}}$ and $f_{\underline{\alpha}'}$, and hence also an identity for the syndromes

$$S_{\underline{\alpha}} = cS_{\underline{\alpha}'} + \sum_{0(\underline{\beta}) < 0(\underline{\alpha})} c_{\underline{\beta}} S_{\underline{\beta}}. \tag{12}$$

We want to have a way to distinguish between functions or syndromes, which are dependent—in the sense of (4) or (12)—and those, which are independent.

To this end we choose a set $\Sigma' \subseteq \Sigma_0$ such that Σ' contains exactly one element $\boldsymbol{x}$ corresponding to each poleorder $O(\boldsymbol{x})$. Let T denote the total order on Σ_0 where $\boldsymbol{x}T\boldsymbol{y}$ iff $x_1 > y_1$ or $x_i = y_i$, $i = 1, \cdots, k$ and $x_{k+1} > y_{k+1}$. Corresponding to the pole order $O(\boldsymbol{x})$ we then take the element $\boldsymbol{x}'$, such that $\boldsymbol{x}'T\boldsymbol{y}$ for all other vectors $\boldsymbol{y}$ with $O(\boldsymbol{y}) = O(\boldsymbol{x}')$.

In the Sakata algorithm we now only consider polynomials, for which the degree belongs to Σ'. This is possible according to (4). As a consequence, we shall use Σ' instead of Σ_0 in the definition of $\Delta = \Delta(F)$, which means that different points in Δ corresponds to functions with different pole orders. And such functions are independent, a fact we use in the next lemma, which like Lemma 1 is essential for the whole setup.

Lemma 2: At each step of the algorithm the number of points in the Δ-set is at most t.

Proof: Let R denote the ring of functions, which have no poles outside P_∞, and let $I \subseteq R$ be the ideal of those functions, which are zero at the error points $P_{j_1}, \cdots, P_{j_t}$. Then the dimension of $R \backslash I$, as a vector space over $\mathbb{F}_q$, is equal to t. Now, for each $\boldsymbol{a} \in \Delta$ we take a polynomial $\sigma_{\underline{\alpha}}$ with $\text{Deg}\,(\sigma_{\underline{\alpha}}) = \boldsymbol{a}$, the corresponding function $g_{\underline{\alpha}} \in \mathbb{R}$ and the image $[g_{\underline{\alpha}}] \in \mathbb{R}/I$. Here $g_{\underline{\alpha}} \notin I$, because otherwise the expressions (10) were zero, and hence $\sigma_{\underline{\alpha}}$ was valid. The same holds for any linear combination of functions $g_{\underline{\alpha}}$. Therefore, the number of elements in Δ is at most the dimension of R/I, that is at most t. □

Let us return to the decoding situation as explained after Lemma 1. Let $\boldsymbol{\gamma} \in \Sigma'$ satisfy $O(\boldsymbol{\gamma}) = m'$. Put $\boldsymbol{\gamma}^{(0)} = \boldsymbol{\gamma}$ and let $\boldsymbol{\gamma}^{(1)}, \boldsymbol{\gamma}^{(2)}, \cdots$ be all the other elements of Σ_0 with pole order m'. With $F = \{\sigma^{(1)}, \cdots, \sigma^{(k)}\}$ we denote a minimal polynomial set for the array $S_{\underline{\beta}}$, $O(\boldsymbol{\beta}) < m'$, where $\text{Deg}\,(\sigma^{(i)}) \in \Sigma'$. We may suppose without loss of generality that all σ's have leading coefficient 1.

Now, take $\sigma^{(i)}$ with $\text{Deg}\,(\sigma^{(i)}) = s^{(i)}$ and suppose that $s^{(i)} \leq \boldsymbol{\gamma}^{(j)}$. Then we can test the polynomial $\sigma^{(i)}$ at the point $\boldsymbol{\gamma}^{(j)}$. We do not know $S_{\gamma(j)}$ yet, but there are two possibilities, either $\sigma^{(i)}$ is valid at $\boldsymbol{\gamma}^{(j)}$ or it is not. If it turns out that $\sigma^{(i)}$

is valid at $\boldsymbol{\gamma}^{(j)}$, then (9) holds, that is

$$S_{\underline{\gamma}(j)} + \sum_{\underline{q} \in \Gamma_{\sigma(i)} \backslash \underline{s}^{(i)}} \sigma_{\underline{q}} S_{\underline{q}+\underline{\gamma}^{(j)}-\underline{s}^{(i)}} = 0 \tag{13}$$

and from this equation we can calculate $S_{\underline{\gamma}(j)}$ and then $S_{\underline{\gamma}}$ is determined by (12).

If $\sigma^{(i)}$ is not valid at the point $\boldsymbol{\gamma}^{(j)}$, that is, if (13) does not hold for the correct value of $S_{\underline{\gamma}(j)}$, then $\sigma^{(i)}$ must be updated. This updating will increase the size of the Δ-set, and we can use Lemma 1 to estimate how much the Δ-set is increased. First, however, we will introduce some notation.

We put

$$K(\boldsymbol{\gamma}) = \{\boldsymbol{x} \in \Sigma' \mid \exists \boldsymbol{\gamma}^{(j)} : \boldsymbol{x} \leq \boldsymbol{\gamma}^{(j)} \wedge \boldsymbol{\gamma}^{(j)} - \boldsymbol{x} \in \Sigma'\}. \tag{14}$$

Note that since there is one-to-one correspondence between pole orders and elements in Σ', the elements of $K(\boldsymbol{\gamma})$ reflects the pole orders $r = O(\boldsymbol{x})$ for which there is a pole order s such that $r + s = O(\boldsymbol{\gamma}) = m'$.

Next, for each $\sigma^{(i)}$ with $\mathrm{Deg}\,(\sigma^{(i)}) = \boldsymbol{s}^{(i)}$, we check if there is a $\boldsymbol{\gamma}^{(j)}$ with $\boldsymbol{\gamma}^{(j)} \geq \boldsymbol{s}^{(i)}$ and $\boldsymbol{\gamma}^{(j)} - \boldsymbol{s}^{(i)} \in \Sigma'$. If such a $\boldsymbol{\gamma}^{(j)}$ exists, we use (13) and (12) to predict the value of $s_{\underline{\gamma}}$ and we put

$$K_i = \{\boldsymbol{x} \in K(\boldsymbol{\gamma}) \mid \boldsymbol{x} \leq \boldsymbol{\gamma}^{(j)} - \boldsymbol{s}^{(j)}\}. \tag{15}$$

If such a $\boldsymbol{\gamma}^{(j)}$ does not exist, then $\sigma^{(i)}$ is not used to find the correct value of $S_{\underline{\gamma}}$.

Let v_i denote the value of $S_{\underline{\gamma}}$ predicted by $\sigma^{(i)}$, if this situation occurs. If v_i turns out to be wrong, then according to Lemma 1, all the points in K_i belongs to the new Δ-set. Therefore, if we put

$$K_i' = K_i \backslash \Delta \tag{16}$$

then the Δ-set increases at least with K_i', if v_i is not the correct value. Of course, K_i' can be empty.

Let $w_1, \cdots, w_p$ be the *different* predictions v_i for $S_{\underline{\gamma}}$ obtained in the way described above, and for each $j = 1, \cdots, p$ let L_j denote the union of the sets (16) for which $v_i = w_j$.

We define the Feng–Rao distance, d_{FR}, for the code in question by

$$d_{\mathrm{FR}} = \min_{\substack{\underline{r} \in \Sigma' \\ 0(\underline{r}) > m}} |K(\boldsymbol{r})| \tag{17}$$

where $K(\boldsymbol{r})$ is defined in (14). It will be clear below, why d_{FR} is the relevant number to consider in this context. In Appendix II we prove that

$$d_{\mathrm{FR}} \geq m - 2g + 2 \tag{18}$$

with equality if $m \geq 4g - 2$. As it is well known, $m - 2g + 2$ is the designed distance for the code, but the true minimum distance might be larger. Now we are able to formulate the main result in the paper, which gives a very simple way to find the correct value of the next syndrome $S_{\underline{\gamma}}$.

Theorem 1: Suppose that the number t of errors satisfies

$$t \leq \left\lfloor \frac{d_{\mathrm{FR}} - 1}{2} \right\rfloor \tag{19}$$

and let $l \in \{1, \cdots, p\}$ be the number for which $|L_l|$ is maximal. Then for the syndrome $S_{\underline{\gamma}}$ we have

$$S_{\underline{\gamma}} = w_l. \tag{20}$$

This is a surprising result and we want to emphasize that the basic idea is due to Feng and Rao. The setup and the proofs are different from those in the Feng–Rao paper, but it is always much easier to prove theorems when you know what you should look for. To see why the result is true we state first the following result, which is proved in Appendix II.

Lemma 3: Let $K' = L_1 \cup L_2 \cup \cdots \cup L_p$ and put $K'' = K' \backslash \Delta$. Then

$$|K''| \geq |K(\boldsymbol{\gamma})| - 2|\Delta|. \tag{21}$$

Proof of Theorem 1: Suppose first that $S_{\underline{\gamma}}$ was different from all the values $w_1, \cdots, w_p$. Then the next Δ-set, which we denote Δ', is increased with at least K'' according to the arguments in relation with (16). But then, using Lemma 3 we have

$$|\Delta'| \geq |\Delta| + |K''| \geq |\Delta| + |K(\boldsymbol{\gamma})| - 2|\Delta|$$

and from this follows, using the definition (17), the assumption (19) and Lemma 2 that

$$|\Delta'| \geq |K(\boldsymbol{\gamma})| - \Delta \geq d_{\mathrm{FR}} - t > t.$$

But this is in contradiction with Lemma 2. So one of the values $w_1, \cdots, w_p$, say w_1, is the correct one.

Now, put $\overline{L}_1 = L_2 \cup \cdots \cup L_p$. Since $w_2, \cdots, w_p$ are different from $S_{\underline{\gamma}}$, the Δ-set will increase with at least $\overline{L}_1$. So by Lemma 2 we have

$$|\Delta| + |\overline{L}_1| \leq t$$

from which follows, using Lemma 3 and (17)

$$|\overline{L}_1| \leq t - |\Delta| < \frac{d_{\mathrm{FR}}}{2} - |\Delta| \leq \frac{1}{2}|K''|. \tag{22}$$

Since $\overline{L}_1 \cup L_1 = K''$, we must have $|L_1| > \frac{1}{2}|K''|$. But this gives us a general insight. Because if we, for any $j = 2, \cdots, p$, put

$$\overline{L}_j = \bigcup_{i \neq j} L_i$$

then $\overline{L}_j \supseteq L_1$ and therefore $|\overline{L}_j| > \frac{1}{2}|K''|$. So the conclusion is in general that the correct value w_l corresponds to the minimal value of $|\overline{L}_l|$, that is, the maximal value of $|L_l|$. This proves the theorem.

IV. The Complexity

The complete decoding algorithm can now be described as follows:

1) Calculate the syndromes $S_{\underline{a}}$, where $O(\boldsymbol{\alpha}) \leq m$, using (4) and (5).
2) Use Sakata's algorithm to find a reduced minimal polynomial set for the array of known syndromes where reduced means that the degrees of all polynomials belong to Σ'.
3) Use Theorem 1 to find $S_{\underline{\gamma}}$, where $O(\boldsymbol{\gamma}) = m+1$ and $\boldsymbol{\gamma} \in \Sigma'$.
4) Calculate all $S_{\underline{\gamma(i)}}$ using (12).

Repeat step 2) to step 4) until all syndromes $S_{\underline{\gamma}}$, where $O(\boldsymbol{\gamma}) \leq d_{\mathrm{FR}} + 4g$, are known, (which means that $2g$ new syndromes must be calculated).

5) Calculate the remaining symdromes using (12) and (13) with polynomials from the last minimal set.
6) Calculate the error values using (7).

We shall make a few comments to the steps above. Let us first consider a reduced minimal set

$$F = \{\sigma^{(1)}, \cdots, \sigma^{(l)}\}.$$

We claim that the number of elements in F is at most a_1, where a_1 is the lowest nonzero pole order. Suppose it is not like that. Then there are two polynomials, say $\sigma^{(i)}$ and $\sigma^{(j)}$ with $\boldsymbol{\alpha}^{(i)} = \mathrm{Deg}\,(\sigma^{(i)})$ and $\boldsymbol{\alpha}^{(j)} = \mathrm{Deg}\,(\sigma^{(j)})$, such that

$$O(\boldsymbol{\alpha}^{(i)}) \equiv O(\boldsymbol{\alpha}^{(j)}) \bmod a_1. \tag{23}$$

Now, $\boldsymbol{\alpha}^{(i)}$ and $\boldsymbol{\alpha}^{(j)}$ belong both to Σ', so if $\boldsymbol{\alpha}^{(i)} = (\alpha_1^{(i)}, \cdots, \alpha_N^{(i)})$, $\boldsymbol{\alpha}^{(j)} = (\alpha_1^{(j)}, \cdots, \alpha_N^{(j)})$ then $\alpha_1^{(i)}$ and $\alpha_1^{(j)}$ are as big as possible for the pole orders in question. Then (23) implies that $\alpha_l^{(i)} = \alpha_l^{(j)}$, $l = 2, \cdots, N$, and therefore $\boldsymbol{\alpha}^{(i)}$ and $\boldsymbol{\alpha}^{(j)}$ cannot both be minimal, which proves the claim.

The next comment is to emphasize that when we carry out step 3), each minimal polynomial is used at most once to find a candidate value for $S_{\underline{\gamma}}$. The last comment is connected with step 5). When we know the syndromes up to pole order $m - 2g + 2 + 4g$, which is $\leq d_{\mathrm{FR}} + 4g$, then all polynomials in the minimal set are valid for all the remaining syndromes (see, e.g., [10, Proposition 4.6] about this) and hence one can proceed as described.

Following these comments we will estimate the complexity of the decoding procedure by counting the number of $\mathbb{F}_q$ multiplications and additions in the different steps.

It is convenient to distinguish between independent syndromes and dependent syndromes. For $\boldsymbol{\alpha} \in \Sigma'$ we call $S_{\underline{\alpha}}$ an independent syndrome. All the dependent syndromes can be calculated from the independent syndromes by simple linear combinations (12). The number of terms on the right-hand side in (12) is at most $r = O(\boldsymbol{\alpha})$. So if $A(r)$ denotes the number of syndromes with order r, then the complexity of finding all the dependent syndromes of order r is

$$rA(r). \tag{24}$$

In the following, we first focus on the independent syndromes, and then later we find the complexity related to the dependent syndromes.

1) There are $m - g + 1$ independent syndromes $S_{\underline{\alpha}}$ with $O(\boldsymbol{\alpha}) \leq m$, and the calculation here costs $(m - g + 1) \cdot 2n$ operations.

2) The number of polynomials in a reduced minimal set is at most the smallest pole order, denoted a_1. Let σ be such a polynomial and let $h = O(\mathrm{Deg}\,(\sigma))$. The number of terms in σ is at most the number of pole orders smaller than or equal to h, and this number is $h - g + 1$. From ([2, p. 228]) follows that one iteration of Sakata's algorithm has complexity $O(a_1(r - g + 1))$, where r is the pole order in question. The complexity of finding a reduced minimal polynomial set for the array of known syndromes is $O(a_1(m - g + 1)^2)$.

3) To calculate the candidate values for $S_{\underline{\gamma}}$, where $O(\boldsymbol{\gamma}) = m+1$ costs at most $a_1(m-g)$ operations. Moreover, we must find the number of elements in the sets K_i', which costs at most $a_1 \cdot d$ operations where $d = d_{\mathrm{FR}}$.

We must repeat calculation of new syndromes and updating of the reduced minimal set up to pole order $d + 4g$.

The complexity of doing this is

$$O((d - 4g - m) \cdot a_1 \cdot (m - g)) + O((d - 4g - m) \cdot a_1 \cdot d) + O(a_1(d + 3g + 1)^2).$$

Using the upper bound n for both m and d, the complexity of the steps considered so far is at most $O(a_1 \cdot n^2)$.

4) and 5) Let us now consider the dependent syndromes. And further let us remark that when we use polynomials in the minimal set to find new syndromes directly—as stated in step 5)—then we consider linear expressions like (12). From the point of view of complexity we can therefore treat these syndromes in the same way as the dependent syndromes.

To calculate all dependent syndromes of order r costs $r(Ar)$ operations, as stated in (24). By summing up $rA(r)$ over all pole orders, we get an upper bound on the complexity we are looking for at this step. If $r = x_1a_1 + \cdots + x_Na_N$ then

$$\begin{aligned}\sum_r A_r \cdot r &= \sum_{\underline{x}} (x_1a_1 + \cdots + x_Na_N) \\ &= \sum_{x_1=0}^{q-1} \cdots \sum_{x_N=0}^{q-1} (x_1a_1 + \cdots + x_Na_N) \\ &= q^{N-1} \sum_{i=1}^{N} a_i \sum_{x_i=0}^{q-1} x_i\end{aligned}$$

and the magnitude of this is $q^{N+1}(a_1 + \cdots + a_N)$.

6) The magnitude for calculating the error values using (6) is $n \cdot q^N \cdot N$ operations. This process can often be speeded up by using a fast transform, but so far we have used the above expression as a measure for the complexity.

Altogether, the complexity for the whole decoding procedure is upper-bounded by

$$O(a_1 \cdot n^2) + O[q^{N+1}(a_1 + \cdots + a_N)] + O(n \cdot N \cdot q^N) \tag{25}$$

where $O(\ )$ in each case means that the number of operations is bounded by a fixed number multiplied by the term inside

the brackets. How good or small this complexity is depends on the special code construction. We illustrate by an example.

Example: Let us consider the curve in the affine 3-space over $\mathbb{F}_q$, $q = r^2$, defined by

$$y^{r+1} = x^r + x \qquad z^{r+1} = -xy^r - yx^r - 1.$$

It follows from [14] that if $r \equiv 1 \bmod 3$, then the curve has $(r^2-1)^2$ $\mathbb{F}_q$-rational points and has genus r^3+r^2-r. At P_∞, the common pole of x, y, and z, the functions x, y, and z have pole orders $(r+1)^2$, $r(r+1)$, and $r(r+2)$, respectively. If we express all the terms in (25) using the code length n, we get

$$O(n^{1/2} \cdot n^2) + O(n^2 \cdot 3 \cdot n^{1/2}) + O(n \cdot 3 \cdot n^{3/2})$$

so in this case the complexity is $O(n^{5/2})$.

Appendix I
Calculation of the Error Values Given $S_{\underline{\alpha}}$, $0 \le \alpha_i \le q-1$

In Section II we explained how to calculate the error value e_l at a point P_l, where $\psi_s(P_l) \neq 0$ for all $s = 1, \cdots, N$. Here we will first treat the case, where $\psi_s(P_l) = 0$ for some, but not all s. Among all points with this property we introduce a partial order given by $P < Q$, iff $\psi_i(P) = \psi_i(Q)$ for all i where $\psi_i(Q) \neq 0$.

Now look at P_l where $\psi_{i_1}(P_l) \neq 0, \cdots, \psi_{i_r}(P_l) \neq 0$ and $\psi_i(P_l) = 0$ for $i \in T = \{1, 2, \cdots, N\} \backslash \{i_1, \cdots, i_r\}$. We form the sum

$$\sum_{\underline{\alpha}} S_{\underline{\alpha}'} \prod_{s=1} \psi_{i_s}^{-\alpha_{i_s}}(P_l)$$

where the summation is over all vectors $\boldsymbol{\alpha}' = (\alpha'_j)$, where $\alpha'_j = 0$ if $j \in T$ and $1 \le \alpha'_j \le q-1$ if $j \notin T$. The sum equals

$$\sum_{j=1}^{n} e_j \prod_{s=1}^{r} \sum_{\alpha_{i_s}=1}^{q-1} \left[\frac{\psi_{i_s}(P_j)}{\psi_{i_s}(P_l)}\right]^{\alpha_{i_s}}$$

which we write as

$$\sum_{j=1}^{n} e_j c_j.$$

If $\psi_{i_s}(P_j) = \psi_{i_s}(P_l)$ for $i = 1, \cdots, r$ we have $c_j = (-1)^r$, and otherwise we have $c_j = 0$. Consequently, $c_j \neq 0$ iff $P_j < P_l$, and the sum is therefore

$$(-1)^r \left[e_l + \sum_{P_j < P_l} e_j \right].$$

Now, if the point P_l is minimal with respect to the partial order, we get in this way e_l directly. So, in general, if we do the calculations according to the partial order starting with minimal elements, the terms in the expression above are all known except e_l, which can therefore be calculated.

What is left is to consider the situation, where $\psi_s(P_l) = 0$ for all $s = 1, \cdots, N$. Clearly, there is at most one such point Q (otherwise the minimum distance is 2), and by the procedure described above all error values e_p, $P \neq Q$, has been calculated. Since

$$S_0 = \sum_{j=1}^{n} e_j$$

it is easy to calculate e_Q.

Appendix II

A. Proof of Lemma 3

We must prove that if $K'' = (\cup K_i) \backslash \Delta$, where K_i is given by (15), then $|K''| \ge |K(\boldsymbol{\gamma})| - 2|\Delta|$.

To see this, let $\boldsymbol{x} \in K(\boldsymbol{\gamma})$, $\boldsymbol{x} \notin \Delta$. Since $\boldsymbol{x} \in K(\boldsymbol{\gamma})$ there is a uniquely determined $\boldsymbol{\gamma}^{(j)}$, such that $\boldsymbol{x} \le \boldsymbol{\gamma}^{(j)}$ (and $\boldsymbol{\gamma}^{(j)} - \boldsymbol{x} \in \Sigma'$). Moreover, since $\boldsymbol{x} \notin \cup K_i$ we know that $\boldsymbol{x} \le \boldsymbol{\gamma}^{(j)} - \boldsymbol{s}^{(i)}$ is not satisfied for any i. Consequently, $\boldsymbol{\gamma}^{(j)} - \boldsymbol{x} \ge 0$ and for all i's the statement $\boldsymbol{\gamma}^{(j)} - \boldsymbol{x} \ge \boldsymbol{s}^{(i)}$ is false. This, however, tells us that $\boldsymbol{\gamma}^{(j)} - \boldsymbol{x} \in \Delta$. In this way, we construct a mapping from $K(\boldsymbol{\gamma}) \backslash (K'' \cup \Delta)$ into Δ. This mapping is injective, since if $\boldsymbol{\gamma}^{(j)} - \boldsymbol{x} = \boldsymbol{\gamma}^{(l)} - \boldsymbol{y}$ then $O(\boldsymbol{x}) = O(\boldsymbol{y})$ and, consequently, $\boldsymbol{x} = \boldsymbol{y}$, since $\boldsymbol{x}, \boldsymbol{y} \in \Sigma'$. But then $|\Delta| \ge |K(\boldsymbol{\gamma})| - |K'' \cup \Delta|$, and the lemma follows.

B. Proof of (18)

$$\begin{aligned} d_{\mathrm{FR}} &= \min_{\substack{\underline{r} \in \Sigma' \\ 0(\underline{r}) > m}} |\{\boldsymbol{x} \in \Sigma' | \exists \boldsymbol{r}^{(j)} : \boldsymbol{x} \le \boldsymbol{r}^{(j)} \wedge \boldsymbol{r}^{(j)} - \boldsymbol{x} \in \Sigma'\}| \\ &= \min_{\substack{\underline{r} \in \Sigma' \\ 0(\underline{r}) > m}} |\{\boldsymbol{x} \in \Sigma' | \exists \boldsymbol{r}^{(j)} \exists \boldsymbol{y} \in \Sigma' : \boldsymbol{x} + \boldsymbol{y} = \boldsymbol{r}^{(j)}\}| \\ &= \min_{r > m} |\{s \text{ is a nongap} | \exists \text{ nongap } t : s + t = r\}. \end{aligned}$$

Now, there are at most g gaps $\le r$, and for each nongap $s \le r$, the number $r - s$ is a gap in at most g cases. From this follows that $d_{\mathrm{FR}} \ge r+1-2g$ and therefore $d_{\mathrm{FR}} \ge m+2-2g$. Moreover, the above argument leads to equality if all gaps are "used," so to speak, and this is the case if $m + 1 - o_g \ge 2g$. Since $o_g \le 2g - 1$ (cf. the beginning of Section II) we have equality if $m \ge 4g - 2$.

References

[1] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory,* vol. IT-15, pp. 122–127, Jan. 1969.

[2] S. Sakata, "Extension of the Berlekamp–Massey algorithm to N dimensions," *Inform. Computat.,* vol. 84 no. 2, pp. 207–239, Feb. 1990.

[3] J. Justesen, K. J. Larsen, H. E. Jensen, and T. Høholdt, "Fast decoding of codes from algebraic plane curves," *IEEE Trans. Inform. Theory,* vol. 38, pp. 111–119, Jan. 1992.

[4] G. L. Feng and T. R. N. Rao, "Decoding algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inform. Theory,* vol. 39, pp. 37–45, Jan. 1993.

[5] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Høholdt, "Fast decoding of AG-codes up to the designed minimum distance," this issue, pp. 1672–1677.

[6] ——, "A fast decoding method of AG codes from Miura–Kamiya curves C_{ab} up to half the Feng–Rao bound," *Finite Fields Appl.,* vol. 1, pp. 83–101, Jan. 1995.

[7] C. Dahl, "Fast decoding of codes from algebraic curves," *IEEE Trans. Inform. Theory,* vol. 40, pp. 223–230, June 1994.

[8] J. H. van Lint and G. van der Geer, *Introduction to Coding Theory and Algebraic Geometry.* Basel, Switzerland: Birkhäuser-Verlag, 1988.

[9] H. Stichtenoth, *Algebraic Function Fields and Codes.* Berlin, Germany: Springer-Verlag, 1993.
[10] M. E. O'Sullivan, "Decoding of codes defined by a single point on a curve," this issue, pp. 1709–1719.
[11] R. Pellikaan, "On the efficient decoding of algebraic-geometric codes," in *Proc. Eurocode 1993*.
[12] C. Kirfel and R. Pellikaan, "The minimum distance of codes in an array coming from telescopic semigroups," this issue, pp. 1720–1732.
[13] I. Duursma, "Decoding codes from curves and cyclic codes," Ph.D. dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, Sept. 1993.
[14] C. Voss and T. Høholdt, "A family of Kummer extensions of the Hermitian function field," *Commun. Algebra,* vol. 23, no. 4, pp. 1551–1567.