

# A class of Sudan-decodable codes

**Nielsen, Rasmus Refslund**

*Published in:*
I E E E Transactions on Information Theory

*Link to article, DOI:*
10.1109/18.850696

*Publication date:*
2000

*Document Version*
Publisher's PDF, also known as Version of record

[3] J. D. Watson, N. H. Hopkins, J. W. Roberts, J. A. Seitz, and A. M. Weiner, *Molecular Biology of the Gene*, 4th ed. Menlo Park, NJ: Benjamin/Cummings, 1987, vol. I.
[4] K. J. Breslauer, R. Frank, H. Blöcker, and L. A. Marky, "Predicting DNA duplex stability from the base sequence," *Proc. Nat. Acad. Sci., USA*, vol. 83, pp. 3746–3750, 1986.
[5] M. S. Waterman, "Combinatorics in molecular biology," in *Handbook of Combinatorics*. Amsterdam, The Netherlands: Elsevier, 1995, pp. 1988–1990.
[6] W. Rychlik, *Oligo: Primer Analysis Software, Version 6.4* Cascade, CO, 1998, Molecular Biology Insights, Inc..
[7] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.
[8] Tables of the maximal Hamming distances for linear codes. [Online] Available: http://www.cw.nl/htbin/aeb/lincodbd/; http://www.win.tue.nl/\~aeb/voorlincod.html

# A Class of Sudan-Decodable Codes

R. Refslund Nielsen, *Student Member, IEEE*

***Abstract*—In this correspondence, Sudan's algorithm is modified into an efficient method to list-decode a class of codes which can be seen as a generalization of Reed–Solomon codes. The algorithm is specialized into a very efficient method for unique decoding. The code construction can be generalized based on algebraic-geometry codes and the decoding algorithms are generalized accordingly. Comparisons with Reed–Solomon and Hermitian codes are made.**

***Index Terms*—Algebraic-geometry codes, decoding, Reed–Solomon codes, Sudan's algorithm.**

## I. Introduction

Reed–Solomon codes are often used in practice due to the fact that they can be decoded efficiently and have the optimal minimum distance for the lengths and dimensions where a Reed–Solomon code exists. During the last decades much effort has been put into the construction of codes with lengths and dimensions not obtainable for Reed–Solomon codes while maintaining a good minimum distance. The study of algebraic-geometry (AG) codes has lead to very promising results.

However, the minimum distance is not the only measure of the usability of a code. For practical purposes it is important that there exists an efficient decoding method to make use of the error-correcting capability, and it is important that error patterns which are likely to occur in the actual application are usually corrected by the decoder.

For example, consider an $(n, k)$ Reed–Solomon code over $\mathbb{F}_{2^m}$. $\mathbb{F}_{2^m}$ can be seen as a vector space of dimension $m$ over $\mathbb{F}_2$, so the code can be seen as an $(mn, mk)$ code over $\mathbb{F}_2$. The minimum distance of the Reed–Solomon code is optimal over $\mathbb{F}_{2^m}$, but the minimum distance of the binary code could be considerably less than for other codes. This means that the Reed–Solomon codes might not correct as many random binary errors as for example a Bose–Chaudhuri–Hocquenghem (BCH) code. A reason why Reed–Solomon codes are still widely used even though the underlying communication channel is binary is that errors are often likely to happen in bursts, so a bit has higher probability of being erroneous if the previous bit was erroneous as well, and a code over a larger alphabet handles this situation better than a binary code.

In [1] a series of new distance functions on vectors over finite sets is introduced and some codes which are good with respect to this distance are constructed. However, decoding methods are not discussed. This correspondence provides efficient methods for unique decoding and for list decoding of the codes presented in [1] which are based on Reed–Solomon and algebraic-geometry codes.

This correspondence is organized as follows: Section II describes the construction based on Reed–Solomon codes and Section III introduces the so-called $r$-distance. In Section IV a list decoding algorithm based on Sudan's algorithm is presented and specialized into a simple algorithm for unique decoding. In Section V comparisons to Reed–Solomon codes are discussed and in Section VI it is shown how the codes can be encoded systematically. Section VII defines some notation on algebraic function fields and generalizes the code construction using this notation. In Section VIII the decoding algorithms are generalized and Section IX is the conclusion.

## II. Construction

Let $\mathbb{F}_q$ denote a finite field with $q$ elements and suppose that

$$P := \{P_1, \cdots, P_{n'}\} \subseteq \mathbb{F}_q, \qquad \text{with } |P| = n'. \tag{1}$$

Consider a polynomial $f \in \mathbb{F}_q[x]$ with $f = \sum_{j=0}^{\deg(f)} f_j x^j$. Given some $P_i \in P$ we can write

$$f = \sum_{j=0}^{\deg(f)} f_{j,i}(x - P_i)^j$$

and it is seen by direct calculation that

$$f_{j,i} = \sum_{j'=j}^{\deg(f)} f_{j'} P_i^{j'-j} \binom{j'}{j}. \tag{2}$$

It is useful to observe that

$$(x - P_i)^j \mid f \Leftrightarrow \forall j' < j (f_{j',i} = 0). \tag{3}$$

*Definition 1:* Let $r$ be a positive integer and let $0 < k \leq rn'$. Then define the following error-correcting code:

$$C(P, r, k) = \{f(P, r) \mid \deg(f) < k\}$$

with $P$ being as in (1) and

$$f(P, r) := (f_{0,1}, \cdots, f_{r-1,1}; f_{0,2}, \cdots, f_{r-1,2}; \cdots; f_{0,n'}, \cdots, f_{r-1,n'}).$$

Notice that for $r = 1$ a Reed–Solomon code is obtained.

Furthermore, it is useful to notice that if $f(P, r) = (c_0, \cdots, c_{n-1})$ then for any $i$ and $j$ with $1 \leq i \leq n'$ and $0 \leq j < r$ we have

$$f_{j,i} = c_{(i-1)r+j}.$$

*Theorem 2 ([1], Theorem 6):* $C(P, r, k)$ is an $\mathbb{F}_q$-linear code with length $n := rn'$ and dimension $k$.

*Proof:* The block length is $n$ by construction and that the code is linear follows from the fact that $(f + \alpha g)_{j,i} = f_{j,i} + \alpha g_{j,i}$ for $f, g \in \mathbb{F}_q[x]$ and $\alpha \in \mathbb{F}_q$. To prove that the dimension is $k$ consider a polynomial $f \in \mathbb{F}_q[x] \backslash \{0\}$ with $\deg(f) < k$. Suppose that $f(P, r)$ is the zero vector. This implies that $\prod_{i=1}^{n'}(x - P_i)^r$ divides $f$, but this

Manuscript received August 16, 1999; revised February 28, 2000. The material in this correspondence will be presented at the IEEE International Symposium on Information Theory, Sorrento, Italy, June 2000.

The author is with the Department of Mathematics, Technical University of Denmark, DK-2800 Lyngby, Denmark (e-mail: R.R.Nielsen@mat.dtu.dk).

Communicate by R. M. Roth, Associate Editor for Coding Theory.
Publisher Item Identifier S 0018-9448(00)05017-3.

is a polynomial of degree $rn'$, which contradicts the assumption that $f$ is nonzero and of degree less than $k \leq rn'$. □

Notice that the polynomial 1 gives a word of weight $n'$ so the minimum distance is at most $n'$ for all $k$. So for $k < (r-1)n'$, $C(P,r,k)$ is not a good error-correcting code in the traditional sense, however, as will be seen later, that does not prevent it from performing well in certain situations.

*Example 3:* Let $\omega$ be a primitive element of $\mathbb{F}_4$ with $\omega^2+\omega+1=0$ and let $P := \{0, 1, \omega, \omega^2\}$. Then $C(P,2,4)$ is an $(8,4)$ code over $\mathbb{F}_4$. Suppose that

$$\begin{aligned} f &= 1 + \omega x + \omega x^2 + x^3 \\ &= \omega^2(x-1) + \omega^2(x-1)^2 + (x-1)^3 \\ &= \omega + (x-\omega) + (x-\omega)^3 \\ &= \omega + (x-\omega^2)^2 + (x-\omega^2)^3 \end{aligned}$$

then

$$f(P,2) = (1, \omega; 0, \omega^2; \omega, 1; \omega, 0).$$

## III. $r$-Distance

As mentioned in Section II, the minimum distance of the code $C(P,r,k)$ is normally bad with respect to the usual Hamming distance. In this section, a distance which will be called $r$-distance is introduced and the properties of $C(P,r,k)$ with respect to $r$-distance are analyzed. The $r$-distance was first mentioned in [1].

In $C(P,r,k)$ codewords consist of $n'$ chunks of $r$ field elements where each chunk corresponds to an element in $P$. This structure is reflected in the following definition of $r$-distance.

*Definition 4:* Let $r$ be a positive integer and let $u, v \in \mathbb{F}_q^n$ with $n = rn'$ for some integer $n'$. For $i \in \{1, \cdots, n'\}$ define the $r$-similarity, $\boldsymbol{s}_r(u,v,i)$, and the $r$-distance, $\boldsymbol{d}_r(u,v,i)$, between $u$ and $v$ with respect to the $i$th chunk as follows:

$$\begin{aligned} \boldsymbol{s}_r(u,v,i) &:= \max\{j \in \{0, \cdots, r\} \mid u_{(i-1)r+j'} = v_{(i-1)r+j'}, \\ &\qquad \text{for all } j' \text{ with } 0 \leq j' < j\} \\ \boldsymbol{d}_r(u,v,i) &:= r - \boldsymbol{s}_r(u,v,i). \end{aligned}$$

Furthermore, define the $r$-similarity, $\boldsymbol{s}_r(u,v)$, and the $r$-distance, $\boldsymbol{d}_r(u,v)$, between $u$ and $v$

$$\begin{aligned} \boldsymbol{s}_r(u,v) &:= \sum_{i=1}^{n'} \boldsymbol{s}_r(u,v,i) \\ \boldsymbol{d}_r(u,v) &:= \sum_{i=1}^{n'} \boldsymbol{d}_r(u,v,i) \end{aligned}$$

Let $f \in \mathbb{F}_q[x]$ and $u \in F_q^n$. The following short notations will then be used:

$$\begin{aligned} \boldsymbol{s}_r(f,u,i) &:= \boldsymbol{s}_r(f(P,r),u,i) \\ \boldsymbol{s}_r(f,u) &:= \boldsymbol{s}_r(f(P,r),u) \\ \boldsymbol{d}_r(f,u,i) &:= \boldsymbol{d}_r(f(P,r),u,i) \\ \boldsymbol{d}_r(f,u) &:= \boldsymbol{d}_r(f(P,r),u). \end{aligned} \tag{4}$$

For example, for all $f \in F_q[x]$ we have $\boldsymbol{d}_r(f, f(P,r)) = 0$ and

$$\boldsymbol{d}_r(f, f(P,r) + (0,1,0,\cdots,0)) = r-1, \qquad \text{if } r > 1.$$

Furthermore, $\boldsymbol{d}_r(f,w) = n - \boldsymbol{s}_r(f,w)$. For $r=1$ the usual Hamming distance is obtained, so $\boldsymbol{d}_1 = \boldsymbol{d}$.

*Theorem 5:* $\boldsymbol{d}_r$ is a distance function on $\mathbb{F}_q^n$.

*Proof:* Let $u, v, w \in \mathbb{F}_q^n$. $\boldsymbol{d}_r(u,v) \geq 0$ and it is straightforward to see that $\boldsymbol{d}_r(u,v) = 0 \Leftrightarrow u = v$ and that $\boldsymbol{d}_r(u,v) = \boldsymbol{d}_r(v,u)$. Furthermore, notice that for $i \in \{1, \cdots, n'\}$, if $\boldsymbol{s}_r(u,w,i) = r$ then $\boldsymbol{d}_r(u,w,i) = 0$, so in this case it is trivial that

$$\boldsymbol{d}_r(u,w,i) \leq \boldsymbol{d}_r(u,v,i) + \boldsymbol{d}_r(v,w,i).$$

If $j := \boldsymbol{s}_r(u,w,i) < r$ then $u_{(i-1)r+j} \neq w_{(i-1)r+j}$ so $v_{(i-1)r+j} \neq u_{(i-1)r+j}$ or $v_{(i-1)r+j} \neq w_{(i-1)r+j}$ and, therefore, $\boldsymbol{s}_r(u,v,i) \leq j$ or $\boldsymbol{s}_r(v,w,i) \leq j$. This implies that

$$\boldsymbol{d}_r(u,v,i) \geq \boldsymbol{d}_r(u,w,i)$$

or

$$\boldsymbol{d}_r(v,w,i) \geq \boldsymbol{d}_r(u,w,i)$$

for all $i$, so

$$\boldsymbol{d}_r(u,w) \leq \boldsymbol{d}_r(u,v) + \boldsymbol{d}_r(v,w)$$ □

The following theorem (a special case of [1, Theorem 6]) gives the minimum $r$-distance of $C(P,r,k)$.

*Theorem 6:* If $u, v \in C(P,r,k)$ with $u \neq v$ then $\boldsymbol{d}_r(u,v) \geq n-k+1$ and $\boldsymbol{d}_r(w,0) = n-k+1$ for some $w \in C(P,r,k)$.

*Proof:* Let $f, g \in \mathbb{F}_q[x]$ be polynomials with degrees less than $k$ so that $u = f(P,r)$ and $v = g(P,r)$. Notice that for each $i \in \{1, \cdots, n'\}$ and $j = 0, \cdots, r-1$

$$(f-g)_{j,i} = f_{j,i} - g_{j,i} = u_{(i-1)r+j} - v_{(i-1)r+j}$$

so $(f-g)_{j,i} = 0$ if $j < \boldsymbol{s}_r(u,v,i)$ and, therefore, (3) gives

$$(x - P_i)^{\boldsymbol{s}_r(u,v,i)} \mid (f-g) \tag{5}$$

which means that a polynomial of degree $\boldsymbol{s}_r(u,v)$ divides $f-g$. Suppose that $\boldsymbol{d}_r(u,v) \leq n-k$. Then $\boldsymbol{s}_r(u,v) \geq k$ implying that $f = g$ and consequently $u = v$ which is false by assumption so $\boldsymbol{d}_r(u,v) \geq n-k+1$.

Let $h \in \mathbb{F}_q[x] \backslash \{0\}$ be an arbitrary nonzero polynomial of degree at most $k-1$. For any $i \in \{1, \cdots, n'\}$ and $j = 0, \cdots, r-1$ the equation $h_{j,i} = 0$ is a homogeneous linear equation in the $k$ coefficients of $h$ by (2). So if $j_1, \cdots, j_{n'} \in \{0, \cdots, r\}$ with $\sum_{i=1}^{n'} j_i = k-1$ then $h$ can be constructed so that $h_{j_i', i} = 0$ for all $i \in \{1, \cdots, n'\}$ and $j_i' < j_i$. By Definition 4, $\boldsymbol{d}_r(h,0) \leq n-k+1$ and by the above $\boldsymbol{d}_r(h,0) = n-k+1$. □

It can be shown (see [1]) that the minimum $r$-distance in the above theorem is the greatest possible given the code length and number of codewords.

## IV. Decoding

In [2], V. Guruswami and M. Sudan presented an algorithm to decode Reed–Solomon codes beyond half the minimum distance by allowing the output to be a (small) list of codewords closest to the received word. In this section, the method in [2] will be generalized to a list decoding method for $C(P,r,k)$. However, first some notation is needed.

Let

$$M := \{x^\alpha y^\beta \in \mathbb{F}_q[x,y] \mid (\alpha, \beta) \in \mathbb{N}^2\}$$

be the set of monomials in $\mathbb{F}_q[x,y]$. A *monomial ordering* is a binary relation, $<_m$, on $M$, which satisfies the following:

- $<_m$ is a total ordering on $M$,
- $\forall f, g, h \in M (f <_m g \Rightarrow fh <_m gh)$,
- $<_m$ is a well-ordering.

One monomial ordering is the *lexicographic order*. The lexicographic order with $y < x$ is defined by

$$x^{\alpha}y^{\beta} <_l x^{\alpha'}y^{\beta'} \Leftrightarrow \alpha < \alpha' \vee (\alpha = \alpha' \wedge \beta < \beta').$$

The lexicographic order with $x < y$ is defined by exchanging $x$ and $y$ in the above definition.

Let

$$f(x,y) \in \mathbb{F}_q[x,y]\backslash\{0\}$$

with $f = \sum_{\alpha,\beta} f_{\alpha}^{(\beta)} x^{\alpha} y^{\beta}$. Then the $(a,b)$-*weighted degree* of $f(x,y)$ is given by

$$\deg^{(a,b)}(f) := \max\left\{\alpha a + \beta b \,|\, f_{\alpha}^{(\beta)} \neq 0\right\}$$

where $a \in \mathbb{N}$ is called the weight of $x$ and $b \in \mathbb{N}$ is called the weight of $y$. For any choice of $a$ and $b$ we may define $\deg^{(a,b)}(0) := -\infty$. In general, $\deg^{(a,b)}(f)$ is called the weight of $f$.

Given a weighted degree $\deg^{(a,b)}$ and a lexicographic order $<_l$, a corresponding *weighted degree lexicographic order* can be defined on $M$ by

$$f <_w g \Leftrightarrow \deg^{(a,b)}(f) < \deg^{(a,b)}(g) \vee (\deg^{(a,b)}(f) = \deg^{(a,b)}(g) \wedge f <_l g)$$

for all $f, g \in M$.

The following lemma describes the weight of the monomials.

*Lemma 7:* Consider the polynomial ring $\mathbb{F}_q[x,y]$ with the weighted degree $\deg^{(1,k-1)}$ for some integer $k > 1$ and let the monomials be ordered by a corresponding $<_w$ order. Suppose that

$$m_0 <_w m_1 <_w m_2 <_w \cdots$$

is an increasing list of all the monomials in $F_q[x,y]$. Then

$$\deg^{(1,k-1)}(m_j) = \left\lfloor \frac{j}{b} + \frac{(b-1)(k-1)}{2} \right\rfloor \tag{6}$$

where $b$ satisfies

$$\binom{b}{2} \leq \frac{j}{k-1} < \binom{b+1}{2}.$$

*Proof:* Group the monomials into the disjoint sets, $M_1, M_2, \cdots$, where

$$M_c = \{m_{j'} \,|\, (c-1)(k-1) \leq \deg^{(1,k-1)}(m_{j'}) < c(k-1)\}.$$

Then $|M_c| = c(k-1)$ so

$$|M_1| + |M_2| + \cdots + |M_{c-1}| = \binom{c}{2}(k-1).$$

Since

$$\binom{b}{2}(k-1) \leq j < \binom{b+1}{2}(k-1)$$

we have $m_j \in M_b$. The smallest monomial in $M_b$ has weighted degree $(b-1)(k-1)$ and for each $a$ with $(b-1)(k-1) \leq a < b(k-1)$ there are exactly $b$ monomials with weighted degree $a$ in $M_b$. If the monomials of $M_b$ are listed increasingly with respect to $<_w$ then $m_j$ is monomial number $j - \binom{b}{2}(k-1)$ so the weighted degree of $m_j$ must be

$$\deg^{(1,k-1)}(m_j) = (b-1)(k-1) + \left\lfloor \frac{j - \binom{b}{2}(k-1)}{b} \right\rfloor = \left\lfloor \frac{j}{b} + \frac{(b-1)(k-1)}{2} \right\rfloor \qquad \square$$

The following definition turns out to be useful:

*Definition 8:* Let $w = (w_0, \cdots, w_{n-1}) \in \mathbb{F}_q^n$ with $n = rn'$. Then define

$$w^{(i)}(x) := \sum_{j=0}^{r-1} w_{(i-1)r+j}(x - P_i)^j.$$

Notice that for any $f \in \mathbb{F}_q[x]$

$$(x - P_i)^{\boldsymbol{s}_r(f,w,i)} \,|\, (f - w^{(i)}(x)). \tag{7}$$

Furthermore, if

$$Q(x,y) = \sum_{\alpha=0}^{\deg_y(Q)} Q^{(\alpha)} y^{\alpha}$$

then $Q$ can be written as follows:

$$Q(x,y) = \sum_{\alpha=0}^{\deg_y(Q)} Q^{(\alpha,i)}(y - w^{(i)}(x))^{\alpha}, \qquad Q^{(\alpha,i)} \in \mathbb{F}_q[x].$$

*Algorithm 9:* As input take the code $C(P,r,k)$, a received word $w$, and a parameter $s \geq 1$.

Let

$$\ell_s := \left\lfloor \frac{n\binom{s+1}{2}}{b_s} + \frac{(b_s - 1)(k-1)}{2} \right\rfloor$$

where $b_s$ satisfies

$$\binom{b_s}{2} \leq \frac{n\binom{s+1}{2}}{k-1} < \binom{b_s+1}{2}.$$

Determine $Q(x,y) \in \mathbb{F}_q[x,y]\backslash\{0\}$ so that

$$\deg^{(1,k-1)}(Q) \leq \ell_s$$

and, furthermore, for $i \in \{1, \cdots, n'\}$, $\alpha \in \{0, \cdots, s-1\}$, and $j \in \{0, \cdots, r(s-\alpha)-1\}$

$$Q_{j,i}^{(\alpha,i)} = 0. \tag{8}$$

Next, let

$$\tau_s := n - \left\lfloor \frac{\ell_s}{s} \right\rfloor - 1$$

and find all factors of $Q$ of the form $y - f$ with $\deg(f) < k$. If $\boldsymbol{d}_r(f,w) \leq \tau_s$ then include $f$ in the output list.

The claim is now that the output list contains all codewords $f(P,r)$ in $C(P,r,k)$, where $\boldsymbol{d}_r(f,w) \leq \tau_s$. To prove that the algorithm works, it must be proven that the polynomial $Q$ exists and that it has the right factors.

*Theorem 10:* $Q(x,y)$ satisfying the conditions above exists.

*Proof:* Equation (8) gives

$$n'(rs^2 - r(0 + 1 + \cdots + s - 1)) = n\binom{s+1}{2}$$

conditions on the polynomial $Q$. Each of these conditions is a homogeneous linear equation in the coefficients of $Q$. By Lemma 7 there are at least $n\binom{s+1}{2} + 1$ monomials in $\mathbb{F}_q[x,y]$ with weighted degree at most $\ell_s$, so there are $n\binom{s+1}{2} + 1$ unknown coefficients. It is a well-known fact from linear algebra that such a system of equations has a nonzero solution. $\square$

*Lemma 11:* If $f \in \mathbb{F}_q[x]$ with $\deg(f) < k$ then $(x - P_i)^{s\boldsymbol{s}_r(f,w,i)}$ divides $Q(x,f)$.

*Proof:* Let

$$R(x) := Q(x,f) = \sum_{\alpha=0}^{b_s-1} (f - w^{(i)}(x))^{\alpha} Q^{(\alpha,i)}(x).$$

By (7) we have that

$$(x - P_i)^{\alpha \boldsymbol{s}_r(f,w,i)} \,|\, (f - w^{(i)}(x))^{\alpha}.$$

For $\alpha \in \{0, \cdots, s-1\}$ (8) ensures that

$$(x - P_i)^{r(s-\alpha)} \,|\, Q^{(\alpha,i)}(x).$$

Therefore,

$$(x - P_i)^{r(s-\alpha)+\alpha \boldsymbol{s}_r(f,w,i)} \,|\, (f - w^{(i)}(x))^{\alpha} Q^{(\alpha,i)}(x).$$

This proves the lemma since

$$r(s-\alpha) + \alpha \boldsymbol{s}_r(f,w,i) \geq \boldsymbol{s}_r(f,w,i)(s-\alpha) + \alpha \boldsymbol{s}_r(f,w,i) = s \boldsymbol{s}_r(f,w,i). \quad \square$$

*Theorem 12:* If a codeword $f(P,r) \in C(P,r,k)$ has $\boldsymbol{d}_r(f,w) \leq \tau_s$ then $(y-f) \,|\, Q$.

*Proof:* By Lemma 11, a polynomial of degree $s\boldsymbol{s}_r(f,w)$ divides $Q(x,f)$, but

$$\boldsymbol{d}_r(f,w) \leq n - \left\lfloor \frac{\ell_s}{s} \right\rfloor - 1 \Rightarrow \boldsymbol{s}_r(f,w) \geq \left\lfloor \frac{\ell_s}{s} \right\rfloor + 1$$

and $\deg(Q(x,f)) \leq \ell_s < s\boldsymbol{s}_r(f,w)$. So $Q(x,f) = 0$ and $y - f$ is, therefore, a factor of $Q$. $\square$

The theorem below gives an idea of the size of $\tau_s$ in Algorithm 9 and corresponds to the result of [2]. The proof, which is omitted here, is similar to [3, Proof of Theorem 3.31].

*Theorem 13:* If $n$ and $s$ are sufficiently large then

$$\frac{\tau_s}{n} \approx 1 - \sqrt{\frac{k}{n}}.$$

The following theorem gives an upper bound on the size of the output list.

*Theorem 14:* The number of codewords returned by Algorithm 9 is less than $b_s$.

*Proof:* By the proof of Lemma 7, the maximal degree in $y$ of $Q$ is $b_s - 1$. Therefore, $Q$ can have at most $b_s - 1$ factors of the form $y - f$, so the number of codewords returned by the algorithm is at most $b_s - 1$. $\square$

The list decoding algorithm can easily be modified into an efficient algorithm for unique decoding of the code $C(P,r,k)$ up to half the minimum $r$-distance. This algorithm, which can be seen as a generalization of the Welch–Berlekamp algorithm for decoding Reed–Solomon codes (see, for example, [4] or [5]), is described in the following.

To modify Algorithm 9 into an algorithm for unique decoding, the parameter $s$ is set to 1, and instead of calculating $b_s$ as described in the algorithm, $b_s$ is set to the constant value 2. Furthermore, the $Q$-polynomial is not allowed to hold terms of degree greater than 1 in $y$. This gives the following algorithm.

*Algorithm 15:* As input take the code $C(P,r,k)$ and the received word, $w \in \mathbb{F}_q^n$.

Let $Q(x,y) = Q^{(0)}(x) + yQ^{(1)}(x) \in \mathbb{F}_q[x,y] \backslash \{0\}$ satisfy that

$$\deg(Q^{(0)}) \leq \left\lceil \frac{n+k}{2} \right\rceil - 1 \text{ and } \deg(Q^{(1)}) \leq \left\lfloor \frac{n-k}{2} \right\rfloor$$

and, furthermore, for each $1 \leq i \leq n'$ and $j < r$

$$Q_{j,i}^{(0)} + \sum_{j'=0}^{j} w_{(i-1)r+j'} Q_{j-j',i}^{(1)} = 0. \tag{9}$$

If there exists a codeword

$$f(P,r) \in C(P,r,k)$$

with $\boldsymbol{d}_r(f,w) \leq \lfloor (n-k)/2 \rfloor$ then $f = -Q^{(0)}/Q^{(1)}$.

The proof that this method indeed works as promised will be omitted here since it is very similar to the proof of Algorithm 9. However, an example of using the above algorithm is given below.

*Example 16:* This continues Example 3. Suppose that the codeword

$$f(P,2) = (1, \omega; 0, \omega^2; \omega, 1; \omega, 0)$$

is sent but

$$w = (1, \omega^2; 0, \omega^2; \omega, \omega^2; \omega, 0)$$

is received. Then $\boldsymbol{d}_2(f,w) = 2$ and since $\lfloor 8 - 4/2 \rfloor = 2$ Algorithm 15 should be able to reconstruct $f$ from $w$.

Solving the system of linear equations in (9) gives the following polynomial $Q$:

$$Q(x,y) = \omega x + \omega x^2 + x^3 + x^5 + (\omega x + x^2)y$$

and

$$\frac{\omega x + \omega x^2 + x^3 + x^5}{\omega x + x^2} = x^3 + \omega x + \omega x + 1 = f.$$

So Algorithm 15 indeed corrects the two errors successfully.

## V. Comparing with Reed–Solomon Codes

$\mathbb{F}_{q^r}$ can be seen as an $r$-dimensional vector space over $\mathbb{F}_q$. So suppose that $k = rk'$ for some integer $k'$ and consider each chunk of $r$ elements in a codeword of $C(P,r,k)$ as an element in $\mathbb{F}_{q^r}$. The following theorem gives the main parameters of this code:

*Theorem 17:* $C(P,r,rk')$ is a code over $\mathbb{F}_{q^r}$ of length $n'$, having $q^{rk'}$ codewords, and minimum distance $n' - k' + 1$.

*Proof:* The code length is given by the construction. The number of codewords is equal to the number of codewords in the code seen as a $\mathbb{F}_q$ code, namely, $q^{rk'}$, however, it should be noticed that the code is not necessarily $\mathbb{F}_{q^r}$-linear. To find the minimum distance consider two polynomials $f, g \in \mathbb{F}_q[x]$, both with degree less than $k$. Let $(F_1, \cdots, F_{n'}) = f(P,r)$ with $F_i \in \mathbb{F}_{q^r}$ for all $i$ and, similarly, $(G_1, \cdots, G_{n'}) = g(P,r)$. Suppose that $F_i = G_i$. Then $f_{0,i} = g_{0,i}, \cdots, f_{r-1,i} = g_{r-1,i}$, so $(x - P_i)^r \,|\, f - g$. This means that if $F_i = G_i$ for $k'$ values of $i$, then a polynomial of degree $k = rk'$ divides $f - g$, but this implies that $f = g$. Therefore, two different codewords can be equal on at most $k' - 1$ positions, which shows that the minimum distance is at least $n' - k' + 1$. Equality holds by the Singleton bound. $\square$

The theorem shows that $C(P,r,rk')$ has the same main parameters as a $(n',k')$ Reed–Solomon code over $\mathbb{F}_{q^r}$, but what about the error correcting capability of the two codes when using Algorithm 15 for unique decoding of $C(P,r,rk')$ and some decoder to decode the Reed–Solomon code up to half the minimum distance? In the Reed–Solomon code, errors will be $\mathbb{F}_{q^r}$-errors, each one corresponding to $r$ $\mathbb{F}_q$-errors. However, in the code $C(P,r,rk')$, error correcting starts from the point in the affected $\mathbb{F}_{q^r}$ symbol where the error actually starts. The effect of this is that some "fractional" $\mathbb{F}_{q^r}$-errors can be recognized, namely, errors which only affect the last part of a $\mathbb{F}_{q_r}$ symbol. For example, on average, random bit-errors will

only count as half an $\mathbb{F}_{q^r}$ error where a full error must be corrected by the Reed–Solomon code. Burst errors of length slightly greater than one $\mathbb{F}_{q^r}$-symbol will on average only count as around $\frac{3}{2}$ errors compared to 2 errors in the Reed–Solomon code. However, it should be noted that usually $n' \ll q^r$, so the Reed–Solomon code considered here is very short.

Now compare using an $(n', k')$ Reed–Solomon code over $\mathbb{F}_q$ with using $C(P, r, rk')$, and suppose that $rk'$ information symbols are to be transmitted. $r$ RS codewords or one $C(P, r, rk')$ codeword will be needed. Let $t := \lfloor (n' - k')/2 \rfloor$ and suppose that $rt$ errors occur. Which code has the highest probability of correcting the errors? There are $\binom{rn'}{rt}$ error patterns in total, but none of the codes will correct all of them if $r > 1$.

The RS code will require the errors to be located so that exactly $t$ errors occur in each chunk of $n'$ elements. The number of error patterns with that property is $\binom{n'}{t}^r$ (in each chunk $t$ errors can occur in $\binom{n'}{t}$ different patterns).

The $C(P, r, rk')$ code will require the errors to happen so that only the last part of each chunk of $r$ elements is affected. To see in how many ways this can happen, suppose that $rt$ errors are added one by one, each time one of the $n'$ chunks is selected to receive the error and the error will have to be located at the last correct position in the chunk. The number of ways to do this is at most $\binom{n'+rt-1}{rt}$ and equality holds only if $t \leq 1$ because if a chunk receives $r$ errors, it will not be able to contain more errors. Experiments indicate that $\binom{n'+rt-1}{rt}$ is normally smaller than $\binom{n'}{t}^r$ and when that is the case, using $r$ RS codewords gives a higher probability of decoding $rt$ randomly positioned errors than using a $C(P, r, rk')$ codeword.

However, it should be emphasized that the error-correcting profile is significantly different for the two codes. Consider an example where $n' = 4$, $k' = 2$, and $r = 2$ (and $q \geq 4$). So four $\mathbb{F}_q$ symbols of information can either be sent as two codewords of a $(4, 2)$ RS code or as one codeword of $C(P, 2, 4)$. Let the two RS codewords be denoted by $a = (a_0, \cdots, a_3)$ and $b = (b_0, \cdots, b_3)$, and the $C(P, 2, 4)$ codeword be denoted by $c = (c_0, \cdots, c_7)$. In the codeword $c$ it is possible to select four elements ($c_1$, $c_3$, $c_5$, and $c_7$) with the property that any pattern of two errors happening among these four symbols can be corrected. It is not possible to select four elements of $a$ and $b$ which has this property, because at least two of the elements will always be from the same codeword. On the other hand, if $a$ and $b$ are interleaved so that $(a_0, b_0, a_1, b_1, a_2, b_2, a_3, b_3)$ is sent then two consecutive errors will always be corrected. It is not possible to arrange the elements of the word $c$ in such a way that the same property is achieved, because if an error happens in $c_j$ with $j \in \{0, 2, 4, 6\}$ then a second error is only guaranteed to be corrected if it occurs in $c_{j+1}$. Therefore, if an error happens at one of these symbols, then a second error in the previous or in the following symbol will give an error pattern which is not guaranteed to be corrected.

In the above, all error patterns were assumed to occur with a probability only depending on the weight of the error pattern. However, this may not always be the case. Consider the following example.

*Example 18:* Let $\mathbb{F}_q = \{\omega_0, \cdots, \omega_{q-1}\}$ and suppose that a vector in $\mathbb{F}_q^{rn'}$ is transmitted as $n'$ integers where a chunk $(\omega_{j_0}, \cdots, \omega_{j_{r-1}})$ is transmitted as

$$c := \sum_{\ell=0}^{r-1} j_\ell q^{r-1-\ell}.$$

What is received is $c + e$ where $|e|$ is a small integer. In this case, errors are most likely to affect the rightmost element, and in general if an element is erroneous then all the elements to the right of that element in the same chunk are almost always erroneous as well. This means that the $r$-distance and the Hamming distance between a codeword and a received word are usually the same. So in this case using a $C(P, r, k)$ code corresponds to using a minimum-distance separable (MDS) code of length $rn'$ which, however, does not exist if $n' = q$ and $r > 1$.

Finally, a word about complexity. Suppose that we have an implementation of Sudan's algorithm which runs in time $O(n^2)$ where $n$ is the code length. Then decoding a $C(P, r, k)$ codeword will be $O(r^2 n^2)$ while decoding $r$ RS codewords will be $O(rn^2)$. So decoding the $C(P, r, k)$ code is generally slower than the similar RS code.

## VI. Systematic Encoding

When using an error-correcting code in practice, it is often desired that encoding can be done systematically. That is, if the code has dimension $k$ then $k$ fixed positions in a codeword contain the information word and the rest of the positions contain check values. In this section, a method to encode systematically for the code $C(P, r, k)$ is described. The main part is the following lemma. Notice that the proof is constructive.

*Lemma 19:* Let $j_1, \cdots, j_{n'} \in \{0, \cdots, r-1\}$ be chosen so that $\sum_{i=1}^{n'} j_i = k$. For each $i \in 1, \cdots, n'$ and $j \in 0, \cdots, j_i - 1$ there exists a polynomial $F^{(i,j)} \in \mathbb{F}_q[x]$ with $\deg(F^{(i,j)}) < k$ so that for all $i' \in 1, \cdots, n'$ and $j' \in 0, \cdots, j_{i'} - 1$

$$F^{(i,j)}_{i',j'} = \begin{cases} 0, & \text{if } i \neq i' \vee j \neq j' \\ 1, & \text{if } i = i' \wedge j = j'. \end{cases}$$

*Proof:* Define

$$B^{(i)} := \frac{\prod_{i'=1}^{n'} (x - P_{i'})^{j_{i'}}}{(x - P_i)^{j_i}}$$

and for $j \in \{0, \cdots, j_i - 1\}$ let

$$B^{(j,i)} := \frac{(x - P_i)^j B^{(i)}}{B^{(i)}(P_i)}.$$

Now $\deg(B^{(j,i)}) < k$ and for $i' \neq i$ and $j' < j_{i'}$, $B^{(j,i)}_{j',i'} = 0$. Furthermore, $B^{(j,i)}_{j',i} = 0$ for $j' < j$ and $B^{(j,i)}_{j,i} = 1$. For $j = j_i - 1, \cdots, 0$ define inductively

$$F^{(j,i)} := B^{(j,i)} - \sum_{j'=j+1}^{j_i - 1} B^{(j,i)}_{j',i} F^{(j',i)}.$$

Then $F^{(j,i)}$ has the properties stated above by construction. □

Letting $j_1, \cdots, j_{n'}$ and $F^{(i,j)}$ be as in the lemma above, encoding can now be done systematically. Let $m \in \mathbb{F}_q^k$ denote the information word (the message) with

$$m = (m_{0,1}, \cdots, m_{j_1,1}; \cdots; m_{0,n'}, \cdots, m_{j_{n'},n'}).$$

If

$$f = \sum_{i=1}^{n'} \sum_{j=0}^{j_i - 1} m_{j,i} F_{j,i}$$

then $\deg(f) < k$ and $f(P, r)$ holds the information word on the $k$ positions determined by the $j_i$'s.

## VII. Construction Based on AG Codes

Let $\chi$ be a nonsingular absolutely irreducible curve over $\mathbb{F}_q$ and let $P_1, \cdots, P_{n'}, P_\infty$ be $\mathbb{F}_q$-rational points on $\chi$. The curve defines an algebraic function field $\mathbb{F}_q(\chi)$ with a discrete valuation $\boldsymbol{v}_{P_i} : \mathbb{F}_q(\chi) \to \mathbb{Z} \cup \{\infty\}$, corresponding to each point $(i = 1, \cdots, n', \infty)$.

Recall that a function $f \in \mathbb{F}_q(\chi)$ is called regular in the point $P_i$ if $\boldsymbol{v}_{P_i}(f) \geq 0$. The functions which are regular in a point form a ring $\mathcal{O}_{P_i}$ which has a unique maximal principal ideal

$$\mathcal{M}_{P_i} = \{f \in \mathbb{F}_q(\chi) \,|\, \boldsymbol{v}_{P_i}(f) > 0\} = \langle t_i \rangle$$

where $t_i$ satisfies $\boldsymbol{v}_{P_i}(t_i) = 1$. $t_i$ is then called a local parameter in $P_i$. Furthermore, the group of units of $\mathcal{O}_{P_i}$ is given by

$$\mathcal{O}_{P_i} \backslash \mathcal{M}_{P_i} = \{f \in \mathbb{F}_q(\chi) \,|\, \boldsymbol{v}_{P_i}(f) = 0\}.$$

Any nonzero function $f \in \mathbb{F}_q(\chi)$ can be written uniquely up to the choice of local parameter as follows:

$$f = t_i^{\boldsymbol{v}_{P_i}(f)} u_f$$

where $u_f$ is a unit, that is, $u_f \in \mathcal{O}_{P_i} \backslash \mathcal{M}_{P_i}$. This will be called the standard representation of $f$ (with respect to the local parameter $t_i$). More details can be found in [6].

A class of algebraic-geometry codes is given by

$$C_{\mathcal{L}}(P, \ell P_\infty) = \{(f(P_1), \cdots, f(P_{n'})) \,|\, f \in \mathcal{L}(\ell P_\infty)\}, \qquad \ell < n'$$

where $P = \{P_1, \cdots, P_{n'}\}$ and

$$\mathcal{L}(\ell P_\infty) = \{f \in \mathbb{F}_q(\chi) \,|\, \boldsymbol{v}_{P_\infty}(f^{-1}) \leq \ell \wedge \boldsymbol{v}_Q(f) \geq 0 \text{ for all } Q \neq P_\infty\}$$

The length of this code is $n'$, and if $g$ denotes the genus of $\chi$ and $2g - 1 \leq \ell < n'$ then the dimension of the code is $k' = \ell - g + 1$ and the minimum distance is lower-bounded by $d^* = n' - \ell$ since the number of zeroes of a nonzero function cannot exceed the number of poles.

$\mathcal{L}(\ell P_\infty)$ is a vector space over $\mathbb{F}_q$ and for $\ell \geq 2g - 1$ the dimension is $\ell - g + 1$. Recall that the nonnegative integers $\mathbb{N}$ are divided into gaps and nongaps by calling $\ell \in \mathbb{N}$ a gap if and only if

$$\mathcal{L}(\ell P_\infty) \backslash \mathcal{L}((\ell - 1)P_\infty) = \emptyset.$$

The number of gaps equals the genus $g$ of the curve defining the function field. For $\ell \in \mathbb{N}$, let $\boldsymbol{g}(\ell)$ denote the number of gaps less than or equal to $\ell$. That is,

$$\boldsymbol{g}(\ell) := \ell - \dim(\mathcal{L}(\ell P_\infty)) + 1. \tag{10}$$

If $\ell \geq 2g - 1$ then $\boldsymbol{g}(\ell) = \ell - (\ell - g + 1) + 1 = g$.

It is well known that $\mathcal{L}(\ell P_\infty)$ has a basis $\phi_0, \cdots, \phi_{\ell - \boldsymbol{g}(\ell)}$, where the pole order at $P_\infty$ is increasing

$$\boldsymbol{v}_{P_\infty}(\phi_0^{-1}) < \boldsymbol{v}_{P_\infty}(\phi_1^{-1}) < \cdots < \boldsymbol{v}_{P_\infty}\left(\phi_{\ell - \boldsymbol{g}(\ell)}^{-1}\right). \tag{11}$$

And conversely, any set of $\ell - \boldsymbol{g}(\ell) + 1$ functions having increasing pole order is a basis of $\mathcal{L}(\ell P_\infty)$. However, the following theorem (from [2]) shows the existence of increasing pole bases where also the zero multiplicity of a given point—different from $P_\infty$—is increasing for some permutation of the basis functions. Furthermore, the proof of the theorem describes a strategy to find these bases.

*Theorem 20:* Let $P_i$ ($i \in \{1, \cdots, n'\}$) be a point. Then there exist functions $\phi_{0,i}, \cdots, \phi_{\ell - \boldsymbol{g}(\ell),i}$ with mutually different pole orders at $P_\infty$ such that

$$\mathcal{L}(\ell P_\infty) = \operatorname{span}\{\phi_{0,i}, \cdots, \phi_{\ell - \boldsymbol{g}(\ell),i}\}$$

and, furthermore,

$$\boldsymbol{v}_{P_i}(\phi_{0,i}) < \boldsymbol{v}_{P_i}(\phi_{1,i}) < \cdots < \boldsymbol{v}_{P_i}(\phi_{\ell - \boldsymbol{g}(\ell),i}).$$

In the following, such a basis will be called an increasing zero basis with respect to the point $P_i$.

*Proof:* Suppose that some increasing pole basis

$$B = \{\phi_0, \cdots, \phi_{\ell - \boldsymbol{g}(\ell)}\}$$

of $\mathcal{L}(\ell P_\infty)$ is given (as in (11)). Let $B_i := \emptyset$ and do the following:

Let $e := \min\{\boldsymbol{v}_{P_i}(f) \,|\, f \in B\}$ and $A := \{f \in B \,|\, \boldsymbol{v}_{P_i}(f) = e\}$.

If $|A| = 1$ then let

$$B := B \backslash A$$
$$B_i := B_i \cup A$$

If $|A| > 1$ then let $a_1, \cdots, a_{|A|}$ be an enumeration of the elements in $A$ such that $\boldsymbol{v}_{P_\infty}(a_1^{-1}) < \boldsymbol{v}_{P_\infty}(a_j^{-1})$ for $j \neq 1$. Write each $a_j$ as its standard representation

$$a_j = t_i^e u_j, \qquad j = 1, \cdots, |A|$$

where $\boldsymbol{v}_{P_i}(u_j) = 0$ and $\boldsymbol{v}_{P_i}(t_i) = 1$. Now let

$$B := (B \backslash A) \cup \{u_1(P_i)a_j - u_j(P_i)a_1 \,|\, j = 2, \cdots, |A|\}$$
$$B_i := B_i \cup \{a_1\}.$$

This ensures that $v_{P_i}(f) > e$ for all $f \in B$ and that the pole orders are unchanged.

The process above is repeated $\ell - \boldsymbol{g}(\ell) + 1$ times until $B = \emptyset$. After this, $B_i$ holds an increasing zero basis of $\mathcal{L}(\ell P_\infty)$ with respect to $P_i$, since $B_i$ is constructed so that two elements cannot have the same valuation in $P_i$. □

Notice that $\boldsymbol{v}_{P_i}(\phi_{0,i}) \geq 0$, so in general $\boldsymbol{v}_{P_i}(\phi_{j,i}) \geq j$. Furthermore, each $\phi_{j,i}$ is in $\operatorname{span}\{\phi_0, \cdots, \phi_{\ell - \boldsymbol{g}(\ell)}\}$ and can be written as

$$\phi_{j,i} = \sum_{j'=0}^{\ell - \boldsymbol{g}(\ell)} \alpha_{j,i,j'} \phi_{j'}, \qquad \alpha_{j,i,j'} \in \mathbb{F}_q. \tag{12}$$

Furthermore, notice that the requirement that an increasing zero basis has different pole orders implies that if $\phi_{0,i}, \cdots, \phi_{\ell - \boldsymbol{g}(\ell),i}$ is an increasing zero basis of $\mathcal{L}(\ell P_\infty)$, then for any $\ell' \leq \ell$ a subset of this increasing zero basis can be used as an increasing zero basis of $\mathcal{L}(\ell' P_\infty)$. This subset will be denoted by

$$\phi_{0,i}^{(\ell')}, \cdots, \phi_{\ell' - \boldsymbol{g}(\ell'),i}^{(\ell')}.$$

Generally, it cannot be assumed that $\phi_{j,i}^{(\ell')} = \phi_{j,i}$ for all $j \leq \ell' - \boldsymbol{g}(\ell')$, because an increasing zero basis may need to be permuted to have increasing pole orders (see Example 21).

Let $f \in \mathcal{L}(\ell P_\infty)$ then $f$ can be written as

$$f = \sum_{j=0}^{\ell - \boldsymbol{g}(\ell)} f_{j,i} \phi_{j,i}.$$

Notice that $\boldsymbol{v}_{P_i}(f) \geq j'$ if $f_{j,i} = 0$ for all $j < j'$. If $f \in \mathcal{L}(\ell' P_\infty)$ for some $\ell' \leq \ell$ then $f$ can be written as

$$f = \sum_{j=0}^{\ell' - \boldsymbol{g}(\ell')} f_{j,i}^{(\ell')} \phi_{j,i}^{(\ell')}.$$

The following gives an example of some of the concepts introduced above.

*Example 21:* An example of a function field is the so-called Hermitian function field defined by the Hermitian curve over $\mathbb{F}_{q_1^2}$

$$X^{q_1+1} - Y^{q_1} - Y = 0.$$

It is well known that this curve indeed is nonsingular and absolutely irreducible. Furthermore, the curve contains $q_1^3$ affine $\mathbb{F}_{q_1^2}$-rational points and has genus $(q_1(q_1 - 1))/2$. In this case, the point $P_\infty$ corresponds to the (unique) point at infinity on the homogenization of the Hermitian curve.

Consider the Hermitian function field over $\mathbb{F}_{16}$. Then $g = 6$ and the gaps are $1, 2, 3, 6, 7, 11$. Furthermore, $1, x, y, x^2, xy, y^2$ is a basis of

$\mathcal{L}(10P_\infty)$ with pole orders $0,\ 4,\ 5,\ 8,\ 9,\ 10$. An increasing zero basis of $\mathcal{L}(10P_\infty)$ with respect to the point $(0,0)$ is

$$1, x, x^2, y, xy, y^2.$$

The zero orders of these functions are $0,\ 1,\ 2,\ 5,\ 6,\ 10$. An increasing zero basis of $\mathcal{L}(5P_\infty)$ is $1, x, y$.

*Definition 22:* Let $r$ be a positive integer and let $k$ satisfy $g \le k \le rn' - g$. Define the following error-correcting code:

$$C_{P_\infty}(P, r, k) := \{f(P, r) \mid f \in \mathcal{L}(mP_\infty)\}$$

where $P = \{P_1, \cdots, P_{n'}\}$, $m := k + g - 1$, and

$$f(P,r) := \Big(f_{0,1}^{(m)}, \cdots, f_{r-1,1}^{(m)}; f_{0,2}^{(m)}, \cdots, f_{r-1,2}^{(m)}; \cdots; f_{0,n'}^{(m)}, \cdots, f_{r-1,n'}^{(m)}\Big)$$

Notice that this definition differs slightly from the definition in [1]. As illustrated in Example 21 there can be "holes" in the zero-order sequence of an increasing zero basis. If there is such a hole among the first $r$ functions of the increasing zero basis at a point $P_i$, then taking the $i$th chunk of the codewords to be the evaluation of the $r$ first coefficients of the Taylor series with respect a local parameter at $P_i$ gives codewords which are always 0 at some position. This is avoided by the use of increasing zero bases.

Just as the codes $C_{\mathcal{L}}(P, \ell P_\infty)$ can be seen as a generalization of Reed–Solomon codes, the codes $C_{P_\infty}(P, r, k)$ can be seen as a generalization of the codes of Definition 1. This is reflected in the following where the notation and most of the results on $C(P, r, k)$ codes presented in the previous sections are generalized to $C_{P_\infty}(P, r, k)$ codes.

The following theorems (from [1, Theorem 6]) give the length, dimension, and minimum $r$-distance of the code $C_{P_\infty}(P, r, k)$.

*Theorem 23:* $C_{P_\infty}(P, r, k)$ is an $\mathbb{F}_q$-linear code with length $n := rn'$ and dimension $k$.

*Proof:* The length is $n$ by construction and the linearity is straightforward. Consider $f \in \mathcal{L}((k+g-1)P_\infty)\backslash\{0\}$. Suppose that $f(P, r)$ is the zero vector. Then

$$\sum_{i=1}^{n'} \boldsymbol{v}_{P_i}(f) \ge rn' > k + g - 1.$$

So the total zero order of $f$ is greater than the pole order, contradicting the assumption that $f \neq 0$. □

*Theorem 24:* If $u, v \in C_{P_\infty}(P, r, k)$ with $u \neq v$ then

$$\boldsymbol{d}_r(u, v) \ge n - k - g + 1.$$

*Proof:* Let $f, h \in \mathcal{L}((k+g-1)P_\infty)$ such that $u = f(P, r)$ and $v = h(P, r)$. For each $i \in \{1, \cdots, n'\}$ and $j = 0, \cdots, r-1$

$$(f-h)_{j,i} = f_{j,i} - h_{j,i} = u_{(i-1)r+j} - v_{(i-1)r+j}$$

so $(f-h)_{j,i} = 0$ if $j < \boldsymbol{s}_r(u, v, i)$ and, therefore,

$$\boldsymbol{v}_{P_i}(f-h) \ge \boldsymbol{s}_r(u, v, i). \tag{13}$$

Now

$$\boldsymbol{s}_r(u, v) = \sum_{i=1}^{n'} \boldsymbol{s}_r(u, v, i) \le \sum_{i=1}^{n'} \boldsymbol{v}_{P_i}(f-h).$$

Since $f - h \in \mathcal{L}((k+g-1)P_\infty)$, the sum of zero orders is at most $k + g - 1$ so $\boldsymbol{s}_r(u, v) \le k + g - 1$ which implies

$$\boldsymbol{d}_r(u, v) = n - \boldsymbol{s}_r(u, v) \ge n - k - g + 1.$$ □

*Example 25:* Let $\omega$ be a primitive element of $\mathbb{F}_4$ with $\omega^2 + \omega + 1 = 0$. Consider the Hermitian function field over $\mathbb{F}_4$ defined by the curve $X^3 + Y^2 + Y = 0$. The genus is $g = 1$ and the curve contains eight points:

$$P := \{(0,0), (0,1), (1,\omega), (1,\omega^2), (\omega,\omega), (\omega,\omega^2), (\omega^2,\omega), (\omega^2,\omega^2)\}.$$

$P_\infty$ corresponds to the (unique) point at infinity on the homogenization of the Hermitian curve. An increasing zero basis of $\mathcal{L}(14P_\infty)$ with respect to the point $P_i = (x_i, y_i)$ for $i \in \{1, \cdots, 8\}$ is given by

$$\begin{aligned}
\phi_{0,i} &= 1\\
\phi_{1,i} &= U\\
\phi_{2,i} &= U^2\\
\phi_{3,i} &= x_i^2 U + V\\
\phi_{4,i} &= x_i^2 U^2 + UV\\
\phi_{5,i} &= x_i U + x_i^2 V + U^2 V\\
\phi_{6,i} &= x_i U + V^2\\
\phi_{7,i} &= x_i^3 U + x_i V + x_i^2 U^2 + x_i V^2 + UV^2\\
\phi_{8,i} &= x_i U + x_i^2 V + x_i UV + x_i^2 V^2 + x_i UV^2 + U^2 V^2\\
\phi_{9,i} &= x_i^2 U + x_i^3 V + x_i U^2 + x_i^3 V^2 + x_i U^2 V + x_i^2 UV^2 + V^3\\
\phi_{10,i} &= x_i^3 U + x_i V + x_i^2 U^2 V + x_i^3 UV^2 + x_i V^3 + x_i^2 U^2 V^2 + UV^3\\
\phi_{11,i} &= x_i U + x_i^2 V + x_i^3 U^2 + x_i UV + x_i^3 U^2 V + x_i UV^3 + U^2 V^3\\
\phi_{12,i} &= x_i^2 U + x_i^3 V + x_i^2 UV + x_i^3 V^2 + x_i^2 UV^2 + V^4\\
\phi_{13,i} &= x_i^2 U^2 + x_i^3 UV + x_i^2 U^2 V + x_i^3 UV^2 + x_i^2 U^2 V^2 + UV^4
\end{aligned}$$

where $U := x - x_i$ and $V := y - y_i$. Consider the code $C_{P_\infty}(P, 2, 11)$. For any $i$, $\mathcal{L}(11P_\infty) = \operatorname{span}\{\phi_{0,i}, \cdots, \phi_{10,i}\}$ so the function

$$f = \omega + \omega x + \omega y + \omega^2 xy + xy^2 + \omega^2 xy^3$$

is encoded as

$$f(P, 2) = (\omega, \omega, 0, \omega^2, \omega, 0, \omega, 1, 0, \omega^2, 1, \omega, \omega^2, 1, 0, 0).$$

## VIII. AG Decoding

The list decoding algorithm for Reed–Solomon codes in [2] by V. Guruswami and M. Sudan is generalized in the same paper to work for a broad class of algebraic-geometry codes. Here the method of Section IV will be generalized to a list decoding method for the code $C_{P_\infty}(P, r, k)$.

For $f \in \mathcal{L}((k+g-1)P_\infty)$ and $u \in F_q^n$ with $n = rn'$ short notations are defined as in (4).

Let $R$ denote the following vector space:

$$R := \bigcup_{\ell=0}^{\infty} \mathcal{L}(\ell P_\infty).$$

Suppose that $R = \operatorname{span}\{\phi_\ell \mid \ell \ge 1\}$ with the pole orders of the $\phi_\ell$'s being strictly increasing. Then $R[z] = \operatorname{span}\{\phi_\ell z^j \mid \ell \ge 1 \wedge j \ge 0\}$ (where $z$ is transcendental over $\mathbb{F}_q(\chi)$). A total ordering on these basis functions will be defined by associating a nonnegative integer—called the weight—to each function. The ordering will be parameterized by the number associated with $z$. Let this be denoted by $\rho(z)$. Then the weight of the basis function $\phi_\ell z^j$ is given by

$$\rho(\phi_\ell z^j) = \boldsymbol{v}_{P_\infty}(\phi_\ell^{-1}) + j\rho(z). \tag{14}$$

An ordering can now be defined using some lexicographic rule to break ties, for example,

$$\begin{aligned}
&\phi_\ell z^j < \phi_a z^b \Leftrightarrow\\
&\rho(\phi_\ell z^j) < \rho(\phi_a z^b) \vee (\rho(\phi_\ell z^j) = \rho(\phi_a z^b) \wedge j < b).
\end{aligned} \tag{15}$$

However, in this context only the weighting is important.

$\rho$ is extended to any nonzero function in $R[z]$ by the following definition:

*Definition 26:* Let $f \in R[z]\backslash\{0\}$. Suppose that $f = \sum_{\ell,j} f_{\ell,j}\phi_\ell z^j$ and that $\rho(z)$ is given. Then the weight of $f$ is defined as

$$\rho(f) = \max\{\rho(\phi_\ell z^j) \,|\, f_{\ell,j} \neq 0\}$$

with $\rho$ given by (14).

The following lemma describes the weight of the basis functions:

*Lemma 27:* Suppose that the basis functions, $\psi_0, \psi_1, \cdots$, of $R[z]$ are enumerated increasingly with respect to the ordering induced by the weighting $\rho(z) \geq 2g-1$

$$\rho(\psi_0) \leq \rho(\psi_1) \leq \cdots.$$

Let $j \in \mathbb{N}$ be given and let $b$ and $t$ satisfy

$$\binom{b}{2}\rho(z) - (b-1)g \leq j < \binom{b+1}{2}\rho(z) - bg$$

$$tb - \boldsymbol{g}(t) \leq j - \left(\binom{b}{2}\rho(z) - (b-1)g\right) < (t+1)b - \boldsymbol{g}(t+1)$$

where $\boldsymbol{g}(t)$ is given by (10).

The weight of $\psi_j$ is now given by

$$\rho(\psi_j) = (b-1)\rho(z) + t.$$

*Proof:* Group the basis functions into disjoint sets, $M_1, M_2, \cdots$, where

$$M_c = \{\psi_\ell \,|\, (c-1)\rho(z) \leq \rho(\psi_\ell) < c\rho(z)\}.$$

Observe that for each $t'$ with $0 \leq t' < \rho(z)$ the number of functions in $M_c$ with weight $t' + (c-1)\rho(z)$ is exactly $c$ if $t'$ is a nongap and $c-1$ if $t'$ is a gap. So

$$|M_c| = c(\rho(z) - g) + (c-1)g = c\rho(z) - g$$

and

$$|M_1| + |M_2| + \cdots + |M_{c-1}| = \binom{c}{2}\rho(z) - (c-1)g.$$

By the definition of $b$ it is seen that $\psi_j \in M_b$ and by the definition of $t$, $\rho(\psi_j) = (b-1)\rho(z) + t$. □

Definition 8 is generalized as follows.

*Definition 28:* Let $w = (w_0, \cdots, w_{n-1}) \in \mathbb{F}_q^n$ with $n = rn'$. Then define

$$w^{(i)} = \sum_{j=0}^{r-1} w_{(i-1)r+j}\phi_{j,i}^{(k+g-1)}.$$

Notice that for any $f \in \mathcal{L}(\ell P_\infty)$

$$\boldsymbol{v}_{P_i}(f - w^{(i)}) \geq \boldsymbol{s}_r(f, w, i)$$

Furthermore, if

$$Q(z) = \sum_{\alpha=0}^{\deg(Q)} Q^{(\alpha)} z^\alpha \in R[z]$$

then $Q$ can be written as follows:

$$Q(z) = \sum_{\alpha=0}^{\deg(Q)} Q^{(\alpha,i)}(z - w^{(i)})^\alpha, \qquad Q^{(\alpha,i)} \in R.$$

Now the algorithm can be stated.

*Algorithm 29:* As input take the code $C_{P_\infty}(P, r, k)$, a received word $w$, and a parameter $s \geq 1$.

Let $\rho(z) := k + g - 1$ and

$$\ell_s := (b_s - 1)\rho(z) + t$$

where $b_s$ and $t$ satisfy

$$\binom{b_s}{2}\rho(z) - (b_s-1)g \leq n\binom{s+1}{2} < \binom{b_s+1}{2}\rho(z) - b_s g$$

$$tb_s - \boldsymbol{g}(t) \leq n\binom{s+1}{2} - \left(\binom{b_s}{2}\rho(z) - (b_s-1)g\right) < (t+1)b_s - \boldsymbol{g}(t+1).$$

Determine $Q(z) \in R[z]\backslash\{0\}$ so that

$$\rho(Q) \leq \ell_s$$

and, furthermore, for $i \in \{1, \cdots, n'\}$, $\alpha \in \{0, \cdots, s-1\}$, and $j \in \{0, \cdots, r(s-\alpha)-1\}$

$$Q_{j,i}^{(\alpha,i)} = 0. \tag{16}$$

Next, let

$$\tau_s := n - \left\lfloor \frac{\ell_s}{s} \right\rfloor - 1$$

and find all factors of $Q$ of the form $z - f$ with $f \in \mathcal{L}((k+g-1)P_\infty)$. If $\boldsymbol{d}_r(f, w) \leq \tau_s$ then include $f$ in the output list.

To prove that the output list contains all codewords $f(P, r) \in C(P, r, k)$ with $\boldsymbol{d}_r(f, w) \leq \tau_s$, it must be proven that the polynomial $Q$ exists and that it has the right factors.

*Theorem 30:* $Q(z)$ satisfying the conditions above exists.

*Proof:* Equation (16) states

$$n'(rs^2 - r(0 + 1 + \cdots + (s-1))) = n\binom{s+1}{2}$$

conditions on the polynomial $Q$. Each of these conditions is a homogeneous linear equation in the coefficients of $Q$. By Lemma 27 there are at least $n\binom{s+1}{2} + 1$ basis functions of $R[z]$ with weight at most $\ell_s$, so there are $n\binom{s+1}{2} + 1$ unknown coefficients. Therefore, a nonzero solution exists. □

*Lemma 31:* If $f \in \mathcal{L}((k+g-1)P_\infty)$ then $\boldsymbol{v}_{P_i}(Q(f)) \geq s\boldsymbol{s}_r(f, w, i)$.

*Proof:*

$$Q(f) = \sum_{\alpha=0}^{b_s-1}(f - w^{(i)})^\alpha Q^{(\alpha,i)}.$$

Since $\boldsymbol{v}_{P_i}(f - w^{(i)}) \geq \boldsymbol{s}_r(f, w, i)$ we have that

$$\boldsymbol{v}_{P_i}((f - w^{(i)})^\alpha) \geq \alpha\boldsymbol{s}_r(f, w, i).$$

For $\alpha \in \{0, \cdots, s-1\}$ (16) ensures that

$$\boldsymbol{v}_{P_i}(Q^{(\alpha,i)}) \geq r(s-\alpha).$$

Therefore,

$$\begin{aligned}\boldsymbol{v}_{P_i}((f - w^{(i)})^\alpha Q^{(\alpha,i)}) &\geq r(s-\alpha) + \alpha\boldsymbol{s}_r(f, w, i)\\ &\geq \boldsymbol{s}_r(f, w, i)(s-\alpha) + \alpha\boldsymbol{s}_r(f, w, i)\\ &= s\boldsymbol{s}_r(f, w, i)\end{aligned}$$ □

*Theorem 32:* If a codeword $f(P, r) \in C_{P_\infty}(P, r, k)$ has $\boldsymbol{d}_r(f, w) \leq \tau_s$ then $(z - f) \,|\, Q$.

*Proof:* By Lemma 31, $\sum_{i=1}^{n'} \boldsymbol{v}_{P_i}(Q(f)) \geq s\boldsymbol{s}_r(f, w)$, but

$$\boldsymbol{d}_r(f, w) \leq n - \left\lfloor \frac{\ell_s}{s} \right\rfloor - 1 \Rightarrow \boldsymbol{s}_r(f, w) \geq \left\lfloor \frac{\ell_s}{s} \right\rfloor + 1$$

and $\boldsymbol{v}_{P_\infty}(Q(f)^{-1}) \leq \ell_s < s\boldsymbol{s}_r(f, w)$. So $Q(f) = 0$ and $z - f$ is, therefore, a factor of $Q$. □

An upper bound on the size of the output list is given by the following theorem:

*Theorem 33:* The number of codewords returned by Algorithm 29 is less than $b_s$.

*Proof:* By the proof of Lemma 27 the degree of $Q$ is at most $b_s$–1. Therefore, $Q$ can have at most $b_s - 1$ factors of the form $z - f$ so the number of codewords returned by the algorithm is at most $b_s - 1$. □

A simple modification of Algorithm 29 (generalizing Algorithm 15) gives an efficient and simple algorithm for unique decoding of the code $C_{P_\infty}(P, r, k)$. However, this algorithm which is described in the following, is only guaranteed to correct up to $r$-distance $\lfloor (n-k-g)/2 \rfloor - g$, which is $g$ less than half the minimum $r$-distance. To be guaranteed to correct up to (and beyond) $r$-distance $\lfloor (n-k-g)/2 \rfloor$, Algorithm 29 must be used for a sufficiently large value of the parameter $s$.

Let $\alpha + \beta \leq \ell$. For any $a \in \{0, \cdots, \alpha - \boldsymbol{g}(\alpha)\}$, and $b \in \{0, \cdots, \beta - \boldsymbol{g}(\beta)\}$, $\phi_{a,i}^{(\alpha)} \phi_{b,i}^{(\beta)} \in \mathcal{L}(\ell P_\infty)$. Define $c(a, b, j)$ by

$$\phi_{a,i}^{(\alpha)} \phi_{b,i}^{(\beta)} = \sum_{j=a+b}^{\ell - \boldsymbol{g}(\ell)} c(a, b, j) \phi_{j,i}.$$

This makes sense since $\boldsymbol{v}_{P_i}(\phi_{a,i}^{(\alpha)} \phi_{b,i}^{(\beta)}) \geq a + b$. Notice that $c(a, b, j)$ depends on $\ell$, $\alpha$, and $\beta$ as well, however, in the following, these numbers will be given by the context.

*Algorithm 34:* Let $Q = Q^{(0)} + zQ^{(1)} \in \mathbb{F}_q(\chi)[z] \backslash \{0\}$ where $z$ is transcendental over $\mathbb{F}_q(\chi)$, and with

$$Q^{(0)} \in \mathcal{L}\left(\left(\left\lfloor \frac{n+k+g}{2} \right\rfloor + g - 1\right) P_\infty\right) \quad \text{and}$$
$$Q^{(1)} \in \mathcal{L}\left(\left(\left\lceil \frac{n-k-g+2}{2} \right\rceil + g - 1\right) P_\infty\right)$$

and, furthermore, for $1 \leq i \leq n'$ and $j < r$

$$Q_{j,i}^{(0)} + \sum_{a=0}^{j} \sum_{b=0}^{j-a} c(a, b, j) w_{(i-1)r+a} Q_{b,i}^{(1)} = 0. \tag{17}$$

If there exists a codeword $f(P, r) \in C_{P_\infty}(P, r, k)$ with $\boldsymbol{d}_r(f, w) \leq \lfloor (n-k-g)/2 \rfloor - g$ then $f = -Q^{(0)}/Q^{(1)}$.

Notice that for an $(n', k')$ AG code it is normally possible to correct up to $\lfloor (n'-k'-g)/2 \rfloor$ errors using a relatively sophisticated algorithm, see, for example, [7]. If a Welch–Berlekamp type algorithm is used (the above method for $r = 1$) only $\lfloor (n'-k'-g)/2 - g \rfloor$ errors are guaranteed to be corrected.

For example, consider using a Hermitian code over $\mathbb{F}_{16}$ with length $64$ and dimension $48$, and compare this to using $C_{P_\infty}(P, 4, 192)$ based on the same curve. Four codewords in the Hermitian code will be able to correct some error patterns of weight up to $4\lfloor (64-48-6)/2 \rfloor = 20$; however, a codeword of $C_{P_\infty}(P, 4, 192)$ will be able to correct some error patterns with weight up to $\lfloor (256-192-6)/2 \rfloor - 6 = 23$.

The following example shows the use of Algorithm 34.

*Example 35:* This is a continuation of Example 25. Suppose that $f(P, 2)$ is sent, but the following word is received:

$$w = (\omega, \omega, 0, \omega^2, \omega, 1, \omega, 1, 0, \omega^2, 1, \omega, \omega^2, 1, 0, \omega)$$

which means that two errors happened, on positions $5$ and $15$, respectively, with the leftmost position being number $0$. So $w$ has 2-distance $2$ to $f(P, 2)$.

$C_{P_\infty}(P, 2, 11)$ is a $(16, 11)$ code with minimum $r$-distance $5$. In this case, the method in the beginning of this section should be able to correct errors only up to 2-distance $\lfloor (16-11-1)/2 \rfloor - 1 = 1$. But proceeding as described, the polynomial $Q$ is determined as

$$Q = (1 + \omega x + \omega^2 x^2 + y^2 + x^2 y + xy^2 + \omega x^2 y^2 + xy^3 + x^2 y^3 + \omega xy^4) + (\omega^2 + \omega x + \omega^2 y) z$$

and it can be verified that indeed

$$(\omega^2 + \omega x + \omega^2 y) f = 1 + \omega x + \omega^2 x^2 + y^2 + x^2 y + xy^2 + \omega x^2 y^2 + xy^3 + x^2 y^3 + \omega xy^4$$

So the method corrects the two errors in this case.

Experiments indicate that the algorithm often corrects up to $r$-distance $\lfloor (n-k-g)/2 \rfloor$.

Notice that if a $C_{P_\infty}(P, r, k)$ code is used in the situation of Example 18 the effect will be close to that of using a very long MDS code. For example, the $(256, 192)$ code $C_{P_\infty}(P, 4, 192)$ mentioned above will in practice usually correct up to 29 errors.

## IX. Conclusion

In this correspondence, efficient list-decoding methods have been presented for the codes introduced in [1]. The codes are generalizations of Reed–Solomon and one point algebraic-geometry codes. The decoding algorithms are generalizations of decodings algorithms presented in [2] for Reed–Solomon and algebraic-geometry codes, and results analogous to the ones obtained in [2] are obtained here with respect to error-correcting capability and upper bounds on the number of codewords in the output.

When comparing the performance of Reed–Solomon and Hermitian codes with the performance of their $r$-distance counterparts it is clear that the $r$-distance codes—which are longer—perform better provided that the error patterns can be assumed to follow the $r$-distance. If error patterns are distributed according to the Hamming distance, the performance seems to be at the same order of magnitude, but with slower decoding for the $r$-distance codes. However, a more precise comparison is still to be made.

## References


[1] M. Y. Rosenbloom and M. A. Tsfasman, "Codes for the $m$-metric," *Probl. Inform. Transm.*, vol. 33, no. 1, pp. 45–52, 1997.

[2] V. Guruswami and M. Sudan, "Improved decoding of Reed–Solomon and algebraic-geometric codes," in *Proc. 39th IEEE Symp. Foundations of Computer Science*, 1998.

[3] R. R. Nielsen, "Decoding AG-Codes Beyond Half the Minimum Distance," Master's Thesis, Tech. Univ. Denmark, Lyngby, Aug. 1998.

[4] M. Morii and M. Kasahara, "Generalized key-equation of remainder decoding algorithm for Reed–Solomon codes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1801–1807, Nov. 1992.

[5] S. B. Wicker and V. K. Bhargava, *Reed–Solomon Codes and their Applications*. Piscataway, NJ: IEEE Press, 1994, sec. 5.3, p. 87.

[6] H. Stichtenoth, *Algebraic Function Fields and Codes*. Berlin, Germany: Springer-Verlag, 1993.

[7] S. Sakata, H. E. Jensen, and T. Høholdt, "Generalized Berlekamp–Massey decoding of algebraic-geometric codes up to half the Feng–Rao bound," *IEEE Trans. Inform. Theory*, vol. 41, pp. 1762–1768, Nov. 1995.