Alternative Approaches to the Computation of Error Values for Hermitian Codes*

Michael E. O'Sullivan

Mathematics Dept.

National University of Ireland

Cork, Ireland

Michael.osullivan@ucc.ie

Abstract

Kötter's algorithm for decoding one-point codes produces a set of error locator polynomials. The advantages of the algorithm are that the logical control is extremely simple and that it uses several Berlekamp-Massey machines working in parallel. In this article we show that minor modifications to the algorithm will produce error evaluator polynomials as well. We also generalize an error evaluation formula for Reed-Solomon codes due to Kötter which involves the derivatives of the locator polynomials and the updating polynomials.

1 Introduction

Kötter's algoithm for decoding one-point codes uses several Berlekamp-Massey type algorithms running in parallel and is the most efficient algorithm for hardware implementation currently available. We show here that with only minor modifications Kötter's algorithm for decoding Hermitian codes can be used to compute error evaluator polynomials in addition to the locator polynomials. The coefficients of the evaluator polynomials are actually already computed in the current formulation of the algorithm, but the information is not used.

Unfortunately, the extra memory required to store the evaluator polynomials makes their computation less practical then simply computing one evaluator after the iterative algorithm is complete. There is an alternative method for error evaluation that uses

*This research is supported in part by NSF Grant No. CCR-98-05080

the update polynomials (which failed at a previous iteration) which may prove to be computationally more efficient than current methods.

This article will survey results in three main subjects which are treated more extensively in the referenced articles, (in particular [2, 4, 7]): the key equation, Kötter's algorithm, and the use of update polynomials in error evaluation.

2 Hermitian Codes

Let $\mathbb{F}_q$ denote the finite field with q elements. We will consider codes constructed from the Hermitian curve $\mathcal{H}$ over $\mathbb{F}_{q^2}$ defined in the xy-plane by

$$y^q + y = x^{q+1}$$

We now review some important facts about Hermitian curves [8, 9]. There is a unique point Q of $\mathcal{H}$ on the line at infinity. The ring

$$R = \frac{\mathbb{F}_{q^2}[x, y]}{< y^q + y - x^{q+1} >}$$

is the ring of functions on $\mathcal{H}$ which have poles only at Q. Let$K$ be the function field of $\mathcal{H}$. For $f \in K$, let $\nu_Q(f)$ be the valuation of f at Q. We will also use the pole order function $o(f) = -\nu_Q(f)$. Let Λ be the set of pole orders of the elements of R—the nongaps. Then

$$\Lambda = \{aq + b(q+1) : a, b \geq 0\}$$

The Hermitian curve has $n = q^3$ rational points, $P_k = (x_k, y_k)$. For $m \in \mathbb{Z}$ we consider the codes

$$C_m = \{v \in \mathbb{F}_{q^2}^n : \sum v_k f(P_k) = 0 \text{ for all } f \in L(mQ)\}$$

where $L(mQ) = \{f \in R : o(f) \leq m\}$. If $e \in \mathbb{F}_{q^2}^n$ is an error vector, the syndrome map defined by e is

$$\begin{aligned} S_e : R &\longrightarrow \mathbb{F}_{q^2} \\ f &\longmapsto \sum e_k f(P_k) \end{aligned}$$

When encoding with C_m, we receive a vector $v + e$ with $v \in C_m$, so we only know the action of the syndrome map on the space $L(mQ)$. The decoding question becomes how to process the knowledge of S_e on $L(mQ)$ to obtain the vector e. For the purpose of exposition we will just focus on the question of computing e from S_e (on all of R). In practice the Feng-Rao majority vote is used to compute $S_e|_{L((m+1)Q)}$ from $S_e|_{L(mQ)}$. Thus if e is correctable, S_e is computable.

3 Two Sets of Bases for the Function Field

We now identify two different ways to write elements of the function field K which are dual to each other. The first basis extends the basis for R used in Kötter's algorithm to all of K.

For any $u \in \mathbb{Z}$ there exist unique integers a, b such that $u = aq + b(q+1)$ and $0 \leq b < q$. We define

$$z_u = x^a y^b$$

Then $o(z_u) = u$ and

$$u \in \Lambda \Longleftrightarrow a \geq 0 \Longleftrightarrow z_u \in R$$

Furthermore, any element $f \in K$ has a unique expression as a series

$$f = \sum_{u \geq \nu_Q(f)} a_u z_{-u} \tag{1}$$

For the other basis we again write $u = aq + b(q+1)$ with $0 \leq b < q$. We define

$$\zeta_u = \begin{cases} x^{-a-1}(y^{q-1} + 1) & \text{if } b = 0 \\ x^{-a-1} y^{q-1-b} & \text{if } b \neq 0 \end{cases} \tag{2}$$

In this case

$$\begin{aligned} \nu_Q(\zeta_u) &= (-a-1)(-q) + (q-1-b)(-q-1) \\ &= aq + b(q+1) - q^2 + q + 1 \\ &= u - (2g - 1) \end{aligned}$$

where g is the genus of $\mathcal{H}$. Furthermore,

$$u \in \Lambda^c \Longleftrightarrow a < 0 \Longleftrightarrow \zeta_u \in R$$

As with the z_u, any $f \in K$ has a unique expansion in terms of the ζ_u.

Let dx be the differential of x. The divisor (dx) is $(2g-2)Q$ so $\nu_Q(\zeta_u dx) = u - 1$. Since $\nu_Q(z_u) = -u$, we have that $\nu_Q(z_u \zeta_u dx) = -1$ and therefore its residue at Q is nonzero. The following theorem show that the two basis $\{z_u\}$ and $\{\zeta_u\}$ are dual with respect to this operation of multiplying by dx and taking the residue at Q.

Theorem 3.1 ([4, Ex. 6.5]). *For any $u, v \in \mathbb{Z}$,*

$$\operatorname{res}_Q z_u \zeta_v dx = \begin{cases} -1 & \textit{if } v = u \\ 0 & \textit{otherwise} \end{cases}$$

4 The Syndrome Series and the Key Equation

For u a nongap, let $s_u = S_e(z_u)$. Then if $f = \sum_{u\in\Lambda} a_u z_u$, by linearity we have $S_e(f) = \sum_{u\in\Lambda} a_u s_u$.

Define the syndrome series as

$$h_e = \sum_{u\in\Lambda} s_u \zeta_u$$

It is clear from Theorem 3.1, that $s_u = \mathrm{res}_Q(z_u h_e dx)$ so the action of $S_e(\star)$ and $\mathrm{res}_Q((\star)dx)$ on R are the same. Having the syndrome expressed as a series now lets us identify the product of the syndrome with any $f \in R$. The coefficients of the product are related to syndromes of multiples of f as shown in the following proposition.

Definition 4.1. Let $f \in K$. Re-indexing in (1), we may write $f = \sum_{u\le o(f)} a_u z_u$. We define the R-part of f to be $\sum_{u\in\Lambda} a_u z_u$.

Proposition 4.2. *For $f \in R$, let*

$$f h_e = \sum_{v\in\mathbb{Z}} t_v \zeta_v \tag{3}$$

Then $t_v = S_e(\tilde{f})$ where $\tilde{f}$ is the R-part of $f z_v$.

PROOF: Write $f z_v = \sum_{u\in\mathbb{Z}} b_u z_u$. Then $\tilde{f} = \sum_{u\in\Lambda} b_u z_u$ and $f z_v = \tilde{f} + \sum_{u\in\Lambda^c} b_u z_u$. From Theorem 3.1, $t_v = -\mathrm{res}_Q\, z_v(f h_e)dx$. But,

$$\mathrm{res}_Q(f z_v) h_e dx = \mathrm{res}_Q\, \tilde{f} h_e dx + \mathrm{res}_Q \sum_{u\in\Lambda^c} b_u z_u h_e dx$$

Since the non-zero terms in h_e are indexed by nongaps, the second term is zero. Thus,

$$\begin{aligned} t_v &= -\mathrm{res}_Q\, \tilde{f} h_e dx \\ &= S_e(\tilde{f}) \end{aligned}$$

□

The following formula provides some additional justification for the use of the series h_e to represent the syndrome. Let h_k be the syndrome series for the kth basis vector which is 1 in the kth place and 0 elsewhere. Then $h_e = \sum e_k h_k$.

$$\begin{aligned} h_k &= \sum_{u\in\Lambda} z_u(P_k)\zeta_u \\ &= \sum_{\substack{0\le a \\ 0\le b<q}} x_k^a y_k^b x^{-a-1}(y^{q-b-1} + \delta_b) \end{aligned}$$

where δ_b is 1 if $b = 0$ and 0 otherwise,

$$\begin{aligned}
&= \frac{1}{x}\sum_{0\le a}(\frac{x_k}{x})^a + \frac{y^{q-1}}{x}\sum_{0\le a}(\frac{x_k}{x})^a \sum_{0\le b<q} \frac{y_k}{y})^b \\
&= \frac{1}{x-x_k} + \frac{1}{x-x_k}\cdot\frac{y^q-y_k^q}{y-y_k} \\
&= \frac{1}{x-x_k}\cdot\frac{y^q-y-y_k^q-y_k}{y-y_k}
\end{aligned}$$

Thus,

$$h_e = \sum_k e_k \frac{1}{x-x_k}\cdot\frac{y^q-y-y_k^q-y_k}{y-y_k}$$

This formula is essentially what is used for error evaluation in [1, §3]. It is easily verified that h_k has a pole of order 1 at P_k and at Q and no other poles, and that $\operatorname{res}_P h_k dx = -res_Q h_k dx = 1$. Now it is clear that $f \in R$ is an error locator if and only if fh_e has no poles except at Q. This leads to the key equation.

Definition 4.3. We say that $f, \varphi \in R$ satisfy the *key equation* if

$$fh_e = \varphi$$

Proposition 4.4. *The following are equivalent:*

1. *f satisfies the key equation.*
2. *$f(P_k) = 0$ at all error positions $(k : e_k \neq 0)$.*
3. *$S_e(fz_u) = 0$ for all $u \in \Lambda$.*

The extension of Kötter's algorithm computes successive approximations to the key equation.

Definition 4.5. For $f \in R$ we define $\operatorname{span} f$ to be the smallest nongap c such that $S_e(z_c f) \neq 0$ (or $\operatorname{span} f = \infty$ if no such c exists). We say that $f, \varphi \in R$ satisfy the *nth key equation* if φ is a linear combination of the ζ_v with $v \in \Lambda^c$ and $v \le n$, and

$$\nu_Q(fh_e - \varphi) \ge n - 2g + 2$$

The *discrepancy of f, φ at $n+1$* is the unique $\alpha \in k$ such that

$$\nu_Q(fh_e - \varphi - \alpha\zeta_{n+1}) > n - 2g + 2 \tag{4}$$

Note that the definition allows for $f = 0$. For example, if $v \in \Lambda^c$ then $0, \zeta_v$ satisfy the $(v-1)$th key equation and the discrepancy is -1 at v.

5 Kötter's Algorithm with Computation of Error Evaluator Polynomials

We now present a matrix formulation for solving the key equation using Kötter's algorithm. Proofs of the results in this section are given for all one-point codes in [7].

Certain subtleties of implementation are not included in the algorithm. They are amply discussed in the references, so we limit ourselves to a few brief comments. Let the encoding code be $C_{m^\star}$. The algorithm has iterations for all integers $m = 0$ to M where M is generally larger than $m^\star$. In practice majority voting is included in step 1 when $m > m^\star$ [2]. If majority voting is unsuccessful, or if the size of the δ-set [6] gets to large, decoding is aborted. The last iteration, M, is determined by the weight and the δ-set of the error vectors to be corrected[5]. For high-rate codes, error vectors of weight much larger than the minimum distance can be corrected provided the distribution of errors is "random."

Data Structure For $i = 0, \ldots, q-1$,

$$B_i^{(m)} = \begin{bmatrix} f_i & \varphi_i \\ g_i & \psi_i \end{bmatrix}^{(m)}$$

Initialization

$$B_0^{(-1)} = \begin{bmatrix} 1 & 0 \\ 0 & -(y^{q-1}+1) \end{bmatrix}$$

and for $i \neq 0$,

$$B_i^{(-1)} = \begin{bmatrix} y^i & 0 \\ 0 & -y^{q-1-i} \end{bmatrix}$$

Algorithm For $m = 0$ to M.

1. Compute the discrepancies,
   Let $s_i = \deg f_i$, let
   $$\alpha_i = S_e(\tilde{f}_i)$$
   where $\tilde{f}_i$ is the R-part of $f z_{m-s_i}$.
2. For each $i = 0, \ldots, q-1$, let j be the least positive residue of $m - i$ modulo q
   Set $r_i = 1 + (m - s_i - s_j)/q$
   If $r_i \leq 0$ or $\alpha_i = 0$, set
   $$A_i = \begin{bmatrix} 1 & -\alpha_i x^{-r_i} \\ 0 & 1 \end{bmatrix} \tag{5}$$

If $r_i > 0$, set

$$A_i = \begin{bmatrix} x^{r_i} & -\alpha_i \\ \alpha_i^{-1} & 0 \end{bmatrix} \tag{6}$$

Then

$$\begin{bmatrix} f_i^{(m)} & \varphi_i^{(m)} \\ g_j^{(m)} & \psi_j^{(m)} \end{bmatrix} = A_i \begin{bmatrix} f_i^{(m-1)} & \varphi_i^{(m-1)} \\ g_j^{(m-1)} & \psi_j^{(m-1)} \end{bmatrix}$$

We now summarize the properties of the data computed.

Theorem 5.1. *Upon completion of the mth iteration of the algorithm:*

1. $o(f_i^{(m)}) \cong i \mod q$, *and* $f_i^{(m)}$ *has leading coefficient 1;*
2. $f_i^{(m)}$ *and* $\varphi_i^{(m)}$ *satisfy the* $m - o(f_i^{(m)})$ *key equation;*
3. $o(g_i^{(m)}) \leq m - o(f_i^{(m)}) + q$;
4. $g_i^{(m)}$ *and* $\psi_i^{(m)}$ *satisfy the* $o(f_i^{(m)}) - q - 1$ *key equation with discrepancy* 1 *at* $o(f_i^{(m)}) - q$; *and*
5. $o(f_i^{(m)}) = i(q+1)$ *if and only if* $g_i^{(m)} = 0$, *in which case* $\psi_i^{(m)} = -(y^{q+1-i})$ *if* $i \neq 0$ *or* $\psi_i^{(m)} = -(y^{q+1}+1)$ *if* $i = 0$.

Corollary 5.2. *For* M *large enough,* $f_i^{(M)}$, $\varphi_i^{(M)}$ *satisfy the key equation and* $f_i^{(M)}$ *is the function in* R *of smallest pole order congruent to* i *modulo* q *to do so. The* $f_i^{(M)}$ *generate* I_e, *the ideal of functions vanishing at the error positions.*

6 Error Evaluation Using Error Evaluators

We now consider two methods for error evaluation. We will drop the superscripts on the functions computed in the algorithm, so $f_i = f_i^{(M)}$.

The $\{f_i\}$ form a generating set for I_e, so for any P_k there is some f_i which has a zero of multiplicity 1 at P_k. Therefore we can use the following proposition to evaluate errors.

Proposition 6.1. *If* f *is a locator with multiplicity one at an error position* P_k *then*

$$e_k = \frac{\varphi(P_k)}{f'(P_k)} \tag{7}$$

where $f' = df/dx$.

PROOF: We will show that for any P_k at which f vanishes,

$$\varphi(P_k) = e_k f'(P_k) \tag{8}$$

If f has a zero of multiplicity one we may divide by $f'(P_k)$ and get (7). Recall that $h_e = \sum_l e_l h_l$ with h_l having poles only at P_l and Q and those of order 1. Since f vanishes at P_k, $fh_l(P_k) = 0$ for $l \neq k$. Thus $\varphi(P_k) = e_k(fh_k)(P_k)$. Now $x - x_k$ is a uniformizing parameter at P_k so we may expand f_i and h_k with respect to $x - x_k$. We get

$$\begin{aligned} h_k &= (x - x_k)^{-1} + \cdots \\ f_i &= a_1(x - x_k) + a_2(x - x_k)^2 + \cdots \end{aligned}$$

Now $(fh_k)(P_k) = a_1$ which equals $df/dx(P_k)$ as was to be proved. □

The downside of using this method is that one may have to check several of the f_i before finding one which has a zero of multiplicity 1 at a given P_k.

Another method is used in the current implementation of Kötter's algorithm. Let f be the error locator of lowest pole order. The zeros of f are computed (this may introduce some extraneous locations). The error values are found using (7) by applying a version of L'Hopitals rule with Hasse derivatives if f has a higher order zero at P_k. If P_k is an extraneous position ($e_k = 0$), $\varphi(P_k)$ will be zero with the same multiplicity as f.

In the current implementation, the error evaluators are not computed iteratively, which saves an enormous amount of memory (the size of $2q$ polynomials of length roughly $t+2g-1$). The product fh_e is computed by convolving the syndrome array and f. This, unfortunately requires some delay, since there are approximately $t + 2g$ coefficients.

7 Evaluation Using Update Polynomials

A third alternative for computing error values uses the theory developed here and an observation of Kötter's [3, Th. 2]. In the Berlekamp-Massey algorithm the matrices involved in the update of the locator-evaluator candidates all have determinant 1. Thus the determinant of the data set

$$\begin{bmatrix} f^{(m)} & \varphi^{(m)} \\ g^{(m)} & \psi^{(m)} \end{bmatrix}$$

is invariant (equal to -1). After the final iteration (dropping superscripts) $f\psi - g\varphi = -1$. Evaluating at an error position we get $g(P_k)\varphi(P_k) = 1$. Since $\varphi(P_k) = e_k f(P_k)$,

$$e_k^{-1} = f'(P_k)g(P_k)$$

and there is no need to compute error evaluators.

We now show how this is generalized to Hermitian codes. Proofs of these results are given in [7].

Theorem 7.1. *At each mth iteration of the algorithm,*

$$\sum_{i=0}^{q-1} \det B_i^{(m)} = -1$$

The proof requires the following proposition.

Proposition 7.2. *In the mth iteration of the algorithm, $\alpha_i = \alpha_j$ whenever $i + j \cong m \mod q$.*

Furthermore, for any m,

$$\operatorname{res}_Q \det \begin{bmatrix} f_i^{(m)} & \varphi_i^{(m)} \\ g_j^{(m)} & \psi_j^{(m)} \end{bmatrix} x^a = \begin{cases} 1 & \text{if } i = j \text{ and } a = -1 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$\operatorname{res}_Q \det \begin{bmatrix} f_i^{(m)} & \varphi_i^{(m)} \\ f_j^{(m)} & \varphi_j^{(m)} \end{bmatrix} x^a = 0 \tag{10}$$

$$\operatorname{res}_Q \det \begin{bmatrix} g_i^{(m)} & \psi_i^{(m)} \\ g_j^{(m)} & \psi_j^{(m)} \end{bmatrix} x^a = 0 \tag{11}$$

The theorem gives a formula for computing error values. Dropping the superscript M on $\det B_i^{(M)}$, we have

$$\sum_{i=0}^{q-1} f_i\psi_i - g_i\varphi_i = -1$$

Evaluating at an error position P_k, the left term vanishes so by (7)

$$e_k^{-1} = \sum_{i=1}^{q-1} g_i(P_k)f'(P_k) \tag{12}$$

8 Comparison of Methods

We now compare the Kötter's method with using (12) for computing error values. First, we remark that Proposition 7.2 already has some impact on implementation. It shows that the ith and jth discrepancies are equal if $i + j \cong m \mod q$. Thus the number of discrepancies to be computed is roughly halved ($(q+1)/2$ if q odd, $1 + q/2$ if q, m even, $q/2$ if q even m odd) which leads to a reduction of circuitry.

Using the new method requires finding the exact set of error locations. Since the lowest pole order locator has at most $t + g$ zeros and the others need only be evaluated at these points, this involves at most $n + (t + g)(q - 1)$ polynomial evaluations. If we take $t \approx g \approx q^2/2$ and $n = q^3$, this gives roughly $2n$ polynomial evaluations as compared to n evaluations required to find the zeros of just one locator.

On the other hand, we avoid the computation of the error evaluator φ which requires $t+2g \approx 3q^2/2$ discrepancy computations. These are in some ways more costly than polynomial evaluations since Horner's method may be used for the latter. The computation of error values is more regular with the new method since it avoids evaluating higher order derivatives when f has a higher order zero at P_k.

To summarize, there are advantages to each method and choosing the optimum method will require a closer analysis of hardware issues and may also depend on the number of errors to be corrected.

References

[1] J. P. Hansen, H. E. Jensen, R Kötter, "Determination of error values for algebraic geometry codes and the Forney formula," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 1881-1886, Sept. 1998.

[2] R. Kötter, "A fast parallel implementation of a Berlekamp-Massey algorithm for algebraic-geometric codes," *IEEE Trans. Inform. Theory*, vol. IT-44, pp. 1353-1368, July, 1998.

[3] R. Kötter, "On the determination of error values for codes from a class of maximal curves," in *Proc. 35th Allerton Conf. on Communication, Control and Computing*, Univ. Illinois, 1997.

[4] M. E. O'Sullivan, "The key equation for one-point codes and efficient error evaluation," submitted to *Finite Fields Appl.*

[5] M. E. O'Sullivan, "Decoding of Hermitian codes beyond the minimum distance bound," submitted to *IEEE Trans. Inform. Theory.*

[6] M. E. O'Sullivan, "Decoding of Hermitian codes: the key equation and efficient error evaluation," submitted to *Trans. Inform. Theory.*

[7] M. E. O'Sullivan, "Solving the key equation for one-point codes with Kötter's algorithm," to be submitted to *Trans. Inform. Theory.*

[8] H. Stichtenoth, "A note on Hermitian curves over GF(q^2)," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1345-1348, Sept. 1991.

[9] H. J. Tiersma, "Remarks on Codes from Hermitian curves," *Trans. Inform. Theory*, vol. IT-33, pp. 605-609, July, 1987.