Systematic Encoding via Gröbner Bases for a Class of Algebraic-Geometric Goppa Codes

Chris Heegard, *Fellow, IEEE*, John Little, and Keith Saints

***Abstract*—Any linear code with a nontrivial automorphism has the structure of a module over a polynomial ring. The theory of Gröbner bases for modules gives a compact description and implementation of a systematic encoder. We present examples of algebraic-geometric Goppa codes that can be encoded by these methods, including the one-point Hermitian codes.**

***Index Terms*—Systematic encoding, algebraic-geometric Goppa codes, Gröbner bases.**

I. Introduction

LET X be a smooth, irreducible projective algebraic curve defined over the finite field $\boldsymbol{F}_q$, and let $D = \sum_{i=1}^{n} P_i$ and G be divisors made up of $\boldsymbol{F}_q$-rational points (places of degree 1) of X, with the P_i distinct and $\operatorname{supp}(D) \cap \operatorname{supp}(G) = \emptyset$. Let $L(G)$ be the vector space of rational functions on X with poles and zeroes bounded by G

$$L(G) = \{f \in \boldsymbol{F}_q(X)^* \mid (f) + G \geq 0\} \cup \{0\} .$$

We will concentrate on the Goppa codes of the form

$$C_L(D, G) = \{(f(P_1), \cdots, f(P_n)) \in \boldsymbol{F}_q^n \mid f \in L(G)\} .$$

In this paper we will consider the problem of constructing systematic encoders for these codes. Basic linear algebra gives one immediate approach to this problem. Namely, given any collection of functions $\{f_1, \cdots, f_k\}$ in $L(G)$, whose images in $L(G)/L(G - D)$ form a basis for that space, we get a generator matrix for $C_L(D, G)$ of the form

$$M = \begin{pmatrix} f_1(P_1) & \cdots & f_1(P_n) \\ \vdots & \ddots & \vdots \\ f_k(P_1) & \cdots & f_k(P_n) \end{pmatrix}.$$

By row operations and, if necessary, column interchanges (i.e., relabeling the points of D), we can take M to the form

$$M' = [I_k \mid B] \tag{1}$$

where I_k is a $k \times k$ identity matrix, and B is some $k \times (n-k)$ matrix. Multiplying the row vector $w \in \boldsymbol{F}_q^k$ on the right by M' gives a systematic encoding function $E : \boldsymbol{F}_q^k \to \boldsymbol{F}_q^n$ for $C_L(D, G)$.

By the results of [1], *every* linear code can be obtained via Goppa's construction (using appropriate X, D, G). So it is probably fruitless to ask for an improvement on this matrix-based method in general. However, we may still ask whether the presence of some *extra algebraic structure* on $C_L(D, G)$ might reduce the amount of information needed to describe the systematic encoder. In other words, we ask for a more *compact* description of the encoder than the collection of $k \cdot (n-k)$ entries of the submatrix B in M' above. To be useful, such extra algebraic structure should be present in many good codes (i.e., ones with large minimum distance d relative to k and n). Naturally, we would also hope that the actual encoding operation could be performed efficiently using the compact representation.

Our motivating example in this regard is the well-studied case of cyclic codes (see, e.g. [2, ch. 6]). Let C be an ideal in the ring $\boldsymbol{F}_q[x]/\langle x^n - 1\rangle$. Then, as is well known, C can be generated by (the coset of) a single monic polynomial $g(x) \mid x^n - 1$. Moreover, a systematic encoder can be constructed using just the information contained in $g(x)$ and the polynomial division algorithm. Namely, if $g(x) = x^r + a_{r-1}x^{r-1} + \cdots + a_0$, where $k = \dim(C) = n - r$, we may take as information symbols the coefficients of $x^{n-1}, \cdots, x^r$. Forming any linear combination

$$p(x) = c_{n-1}x^{n-1} + \cdots + c_r x^r$$

and computing the remainder $\overline{p(x)}$ on division by $g(x)$, we find that $c(x) = p(x) - \overline{p(x)}$ is a multiple of $g(x)$, i.e., is a codeword. Furthermore, since $\overline{p(x)}$ is a linear combination of $x^{r-1}, \cdots, x, 1$, $c(x)$ contains the same $x^{n-1}, \cdots, x^r$ terms as $p(x)$. Hence, we have a systematic encoder for C.

To get something similar, we will assume that the code $C_L(D, G)$ has a nontrivial automorphism. In the examples we present, the automorphism of the code we use will be induced by an *$\boldsymbol{F}_q$-rational automorphism* σ of the curve X fixing the divisors G and D, and inducing a nontrivial permutation of the points of D, though this is not necessary for our approach. It is well-known—see, e.g., [3, sec. VII.3.3]—that if we have such a automorphism $\sigma \in \operatorname{Aut}(X)$, the mapping $f \mapsto f \circ \sigma^{-1}$ takes the vector subspace $L(G) \subset \boldsymbol{F}_q(X)$ to itself, and hence induces an automorphism of the code $C_L(D, G)$. In fact, by considering the *orbit decomposition* of the entries of the codewords under the action of the cyclic subgroup H generated by the automorphism of the code, we can see that the codewords of such a code consist of several "blocks" (one

Manuscript received July 12, 1994; revised July 19, 1995. This work was supported in part by the U.S. Army Research Office through the Army Center of Excellence for Symbolic Methods in Algorithmic Mathematics (ACSyAM), Mathematical Sciences Institute of Cornell University under Contract DAAL03-92-G-0126, and in part by NSF under Grant NCR-9207331.

C. Heegard is with the School of Electrical Engineering Cornell University, Ithaca NY 14853 USA.

J. Little is with the Department of Mathematics, College of the Holy Cross, Worcester, MA 01610 USA

K. Saints is with QUALCOMM Inc., San Diego, CA 92121 USA.

IEEE Log Number 9414640.

per orbit) each of which is permuted cyclically. Codes of this form have properties very similar to cyclic codes, and can be systematically encoded in a fashion that parallels the method described above for the cyclic case to a striking degree.

There are two main reasons we believe that this line of thought is interesting. First, it applies directly to many good codes. In order for Goppa's construction to yield good codes from a curve X of genus $g > 0$, X must have many $\boldsymbol{F}_q$-rational points. But curves with many $\boldsymbol{F}_q$-rational points tend to be special in other ways too. In particular, they tend to have many automorphisms.

To give one example, consider the much-studied *Hermitian curve* X_m over the field $\boldsymbol{F}_{m^2}$, given by the affine equation

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

(see [3]–[5]). X_m has a single point Q (rational over $\boldsymbol{F}_{m^2}$) at infinity, and m^3 other $\boldsymbol{F}_{m^2}$-rational points $P_i, i = 1, \cdots, m^3$. As is well known, X_m has the maximum possible number of $\boldsymbol{F}_{m^2}$-rational points for a curve of genus $g = m(m-1)/2$. Write α for a generator of the multiplicative group $\boldsymbol{F}_{m^2}^*$. Then X_m always has an automorphism of the form

$$\sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^{m+1} y. \end{cases} \tag{2}$$

The automorphism σ fixes Q, and permutes the other $\boldsymbol{F}_{m^2}$-rational points. (Of course, X_m has many other automorphisms as well; this particular σ is interesting because it has order $m^2 - 1$, quite large. Hence the number of orbits will be small, and their lengths will be large.) Hence, taking $G = aQ$, and

$$D = \sum_{i=1}^{m^3} P_i$$

we are in the situation described above. The automorphism σ induces an automorphism of each of the codes $C_L(D, aQ)$ on the Hermitian curve X_m. Other curves with automorphisms that can be exploited in the same way include, but are not limited to, the family of curves with many $\boldsymbol{F}_q$-rational points considered by Hansen and Stichtenoth in [7], the Klein quartic [6], the *modular curves* $X_0(\ell)$ considered by Tsfasman, Vlăduţ, and Zink in their construction of long codes exceeding the Gilbert–Varshamov bound [8], any elliptic or hyperelliptic curve, and so forth.

Second, there is an extra algebraic structure on the codes $C_L(D, G)$ in this situation. Writing the entries in each cyclically permuted block as a polynomial in a variable t, we will show that $C = C_L(D, G)$ has the structure of a finitely generated *module* over the ring $\boldsymbol{F}_q[t]$, where multiplication by t acts as a cyclic permutation on each block. The dual code

$$\begin{aligned} C_\Omega(D, G) &= \{(c_1, \cdots, c_n) \mid \sum_{i=1}^{n} c_i f(P_i) = 0, \\ &\qquad \text{for all } f \in L(G)\} \\ &= \{(\mathrm{Res}_{P_1}(\omega), \ldots, \mathrm{Res}_{P_n}(\omega)) \mid \omega \in \Omega(G - D)\} \end{aligned}$$

([3, sec. VII.1]) also has the structure of a $\boldsymbol{F}_q[t]$-module.

To C we can associate a submodule $\overline{C}$ of the free module $\boldsymbol{F}_q[t]^r$, where r is the number of orbits of the points of D under the action of the cyclic group of automorphisms generated by σ. The theory of Gröbner bases for modules over polynomial rings ([9]) now provides both a convenient language for specifying, and a powerful constructive way to implement the desired compact representation of a systematic encoder. We summarize our encoding algorithm as follows. (See Subsection II-C below for a more precise description.) Select and fix some monomial order on terms in the free module $\boldsymbol{F}_q[t]^r$. Then

1) The module $\overline{C}$ has a Gröbner basis $\mathcal{G}$ of size equal to r (the number of orbits), which can be computed using, e.g., Buchberger's algorithm. There are also much more efficient algorithms for computing Gröbner bases in special cases. For Hermitian codes, see our paper [10].
2) As information positions for the code we can take a certain subset of the collection of *nonstandard* monomials for $\overline{C}$ (that is, those monomials appearing as leading terms of some element of $\overline{C}$). The parity checks are then the *standard monomials.*
3) Forming any linear combination w of the nonstandard monomials and applying division by the Gröbner basis $\mathcal{G}$, we reduce to the remainder $\overline{w}^{\mathcal{G}}$ (this contains only standard terms). Then

$$E(w) = w - \overline{w}^{\mathcal{G}} \in \overline{C}$$

gives the systematic encoder.
4) The information needed to describe $\mathcal{G}$ and hence the encoder is a collection of at most $r \cdot (n-k)$ coefficients; this is usually significantly smaller than the number $k \cdot (n-k)$ of coefficients in the right-hand block B of the row-reduced generator matrix in (1).

This yields an encoder which can be compactly represented. Moreover, the computation of the remainder $\overline{w}^{\mathcal{G}}$ takes no more work than computing the parity checks directly from B in (1). We remark that very similar ideas can be used to generate systematic encoders for the m-dimensional extended cyclic codes, which can be viewed as $\boldsymbol{F}_q[t_1, \cdots, t_m]$-modules (see [11] and [12] for the case of m-dimensional cyclic codes).

It is, of course, also true that the codes we consider can be viewed as modules over the group algebra of the full automorphism group of the code. However, when this automorphism group is nonabelian, the group algebra is a noncommutative ring. Gröbner bases for modules over rings of polynomials in noncommuting variables could conceivably be applied to the same effect. In the cases we have considered, however, there seems to be little advantage to this. Even though we are not using *all* of the symmetries of the codewords, restricting to cyclic groups of automorphisms lets us apply some very powerful, yet relatively simple algebraic machinery.

The rest of the paper is organized as follows. In Section II we collect all the theoretical results on automorphisms of curves and the induced automorphisms of Goppa codes that we will need, and we establish the existence of the module structure described above. Finally, a more precise statement of the encoding algorithm is given. Section III is devoted to several detailed examples, illustrating how this encoding method works in practice.

II. Theoretical Results

A. Module Structures on Certain Goppa Codes.

We begin by describing some useful module structures on certain Goppa codes. Most of the cases we will consider come from the following situation.

Let X be an irreducible, smooth projective algebraic curve defined over $\boldsymbol{F}_q$, and let σ be an $\boldsymbol{F}_q$-rational automorphism of X. Then σ permutes the points of X rational over $\boldsymbol{F}_q$ (or equivalently, the places of degree 1 of the function field of X), and hence can be extended to act on divisors on X consisting of $\boldsymbol{F}_q$-rational points. Suppose that:

1) The curve X has an $\boldsymbol{F}_q$-rational automorphism σ that fixes two divisors D, G on X consisting of $\boldsymbol{F}_q$-rational points and with disjoint supports.
2) The automorphism σ induces a nontrivial permutation of the points of D. (This will be true automatically, for instance, if the degree of D is sufficiently large with respect to the genus of X.)

We then have the following standard fact.

(II.A.1) Lemma: The automorphism σ induces a nontrivial automorphism

$$\begin{aligned}\sigma\cdot(f(P_1),\cdots,f(P_n)) &= ((f\circ\sigma^{-1})(P_1),\cdots,(f\circ\sigma^{-1})(P_n)) \\ &= (f(\sigma^{-1}(P_1)),\cdots,f(\sigma^{-1}(P_n)))\end{aligned} \tag{3}$$

of the Goppa code $C_L(D,G)$ constructed from X.

Proof: (see, e.g., [3, sec. VII.3.3]). △

An immediate corollary of (3) above is the following symmetry of the codewords of $C_L(D,G)$.

(II.A.2) Corollary: Let H be the cyclic subgroup of $\mathrm{Aut}(X)$ generated by σ, and let

$$\mathrm{supp}(D) = O_1 \cup O_2 \cup \cdots \cup O_r$$

be the decomposition of the support of D into disjoint orbits under the action of H. Then the entries of codewords corresponding to the points in each O_i are permuted cyclically by σ.

(II.A.3) Examples: Consider the Hermitian curve X_3 over the field $\boldsymbol{F}_9$, given by the affine equation $x^4 = y^3 + y$, as in the introduction. Taking $m = 3$ in the general form (2) above, we obtain an automorphism σ fixing the point Q at infinity, and the divisor

$$D = \sum_{i=1}^{27} P_i$$

formed from the 27 affine $\boldsymbol{F}_9$-rational points. We represent $\boldsymbol{F}_9$ as $\boldsymbol{F}_3[\alpha]/\langle\alpha^2+\alpha-1\rangle$. Then, under the action of $H = \langle\sigma\rangle$, $\mathrm{supp}(D)$ decomposes into five distinct orbits

$$\begin{aligned}&O_1 = O((1,\alpha^7)),\ O_2 = O((1,\alpha^5)),\ O_3 = O((1,\alpha^4)),\\ &O_4 = O((0,\alpha^2)),\ O_5 = \{(0,0)\}.\end{aligned}$$

The orbits O_1, O_2, and O_3 have eight elements each, O_4 has two elements, and O_5 is a singleton. (By the general theory of group actions, the number of elements in an orbit is the number of cosets of the subgroup of H fixing one element of the orbit—hence the number of elements in each orbit is a divisor of the order of the cyclic group H, 8 in this case.) By the Corollary, the entries in any of the codewords of a $C_L(D,aQ)$ code on this curve decompose into five "blocks" according to these orbits, each of which is cyclically permuted by σ.

In this case (and in all others where m is prime), X_m has automorphisms of order greater than m^2-1 as well. Consider

$$\tau : \begin{cases} x \mapsto \alpha^2 x \\ y \mapsto y + \alpha^2. \end{cases}$$

It is easy to check that τ is an automorphism of X_3, of order 12. τ also fixes the divisors D and G. Under the action of $H' = \langle\tau\rangle$, $\mathrm{supp}(D)$ decomposes into *three* orbits

$$O_1' = O((1,\alpha^4)),\ O_2' = O((\alpha,1)),\ O_3' = O((0,0)).$$

The orbits O_1' and O_2' have 12 elements each, while O_3' has three elements. By (II.A.2), we get a second, distinct decomposition of the codewords of $C_L(D,aQ)$ into cyclically permuted blocks.

We next give a more precise description of the $\boldsymbol{F}_q[t]$-module structure on $C_L(D,G)$ that is induced by the action of σ given in (3). The basic idea is very simple: multiplication by t in this module will mean applying the automorphism σ of the code. We note that $C_L(D,G)$ codes can also have automorphisms that are not induced by automorphisms of the curve. The same construction can be applied in those cases as well.

A convenient way to describe this $\boldsymbol{F}_q[t]$-module structure is as follows. We may relabel the points of D as $P_{i,j}$, where $i = 1,\cdots,r$, and for each given i, j runs from 0 to $|O_i|-1$. Pick any one point $P_{i,0} \in O_i$, and enumerate the points in the ith orbit as $P_{i,j} = \sigma^j(P_{i,0})$. We have $P_{i,|O_i|} = P_{i,0}$. Similarly, by convention we will write $P_{i,-1} = \sigma^{-1}(P_{i0}) = P_{i,|O_i|-1}$. Rearranging the components, we may represent the codewords as r-tuples of polynomials in one variable:

$$(h_1(t),\cdots,h_r(t)) \tag{4}$$

where

$$h_i(t) = \sum_{j=0}^{|O_i|-1} f(P_{i,j})t^j \tag{5}$$

and $f \in L(G)$.

The most direct way to incorporate the cyclic permutations of the entries is to view the r-tuples (4) as elements of the $\boldsymbol{F}_q[t]$-module

$$M = \bigoplus_{i=1}^{r} \boldsymbol{F}_q[t]/\langle t^{|O_i|}-1\rangle\ . \tag{6}$$

The collection C of r-tuples obtained from the words of the code $C_L(D,G)$ is closed under sums. Furthermore, multiplication of an r-tuple of polynomials as in (4) by t has the

following effect:

$$\begin{aligned} t \cdot h_i(t) &= \sum_{j=0}^{|O_i|-1} f(P_{i,j}) t^{j+1} \\ &\equiv \sum_{j=0}^{|O_i|-1} f(P_{i,j-1}) t^j \mod \langle t^{|O_i|} - 1 \rangle \qquad (7) \\ &= \sum_{j=0}^{|O_i|-1} f(\sigma^{-1}(P_{i,j})) t^j. \end{aligned}$$

Comparing with (3) above, we see that multiplying an element of C by t is the same as applying the automorphism of the code induced by σ. Hence C is closed under multiplication by t, and we have the following consequence of our observations.

(II.A.4) Proposition: Let C be the $\boldsymbol{F}_q$-vector subspace of the module M in (6) formed by the codewords of the code $C_L(D, G)$. Then C is a $\boldsymbol{F}_q[t]$-submodule of M under the component-wise multiplication given in (7).

(II.A.5) Remark: By the structure theorem for modules over a principal ideal domain (see, e.g., [13, ch. 3]), it follows that C is isomorphic to a direct sum of cyclic $\boldsymbol{F}_q[t]$-modules. However, that algebraic decomposition *does not* coincide with the cyclic block decomposition of the codewords discussed above.

(II.A.6) Alternate Description: Consider the $\boldsymbol{F}_q[t]$–submodule $\overline{C}$ of the free module $\boldsymbol{F}_q[t]^r$ generated by the codewords of $C_L(D, G)$ and the $q_i = (t^{|O_i|} - 1)\boldsymbol{e}_i$, where $i = 1, \cdots, r$, and $\boldsymbol{e}_i$ is the ith standard basis vector in $\boldsymbol{F}_q[t]^r$. In other words, $\overline{C}$ is the inverse image $\pi^{-1}(C)$ under the obvious surjection

$$\boldsymbol{F}_q[t]^r \xrightarrow{\pi} \bigoplus_{i=1}^{r} \boldsymbol{F}_q[t]/\langle t^{|O_i|} - 1 \rangle \ .$$

It is in this sense that $C_L(D, G)$ can be associated with a submodule of a free module over $\boldsymbol{F}_q[t]$, so that the standard theory of Gröbner bases for modules over a polynomial ring may be applied.

(II.A.7) Remark: For completeness, we mention that everything we have said above for the Goppa code $C_L(D, G)$ also carries over to the dual code

$$\begin{aligned} C_\Omega(D, G) &= \{(c_1, \ldots, c_n) \mid \sum_{i=1}^{n} c_i f(P_i) = 0, \\ &\qquad \text{for all } f \in L(G)\} \\ &= \{(\mathrm{Res}_{P_1}(\omega), \ldots, \mathrm{Res}_{P_n}(\omega)) \mid \omega \in \Omega(G - D)\}. \end{aligned}$$

In particular, $C_\Omega(D, G)$ also has the structure of a $\boldsymbol{F}_q[t]$-module. This is most easily seen by using (II.A.2). The automorphism σ of X also induces an automorphism of the dual code, which acts as a cyclic permutation on the entries corresponding to each H-orbit in supp (D).

B. Gröbner Bases for $\boldsymbol{F}_q[t]$-modules.

Applications of symbolic algebraic techniques to problems in coding theory have been considered for instance in [11], [12], [14], and [15]. The theory of Gröbner bases for polynomial ideals is one particularly powerful tool in this area, and expositions have appeared, for example in [16]–[19]. The theory for modules over polynomial rings is completely analogous, but perhaps less familiar. As a general reference, we suggest [9, ch. 3]. For the convenience of the reader, we summarize the portions of that theory that are needed here. Since we will consider only modules over the one-variable polynomial ring $\boldsymbol{F}_q[t]$, the algebra simplifies considerably. Consider the free module $F = \boldsymbol{F}_q[t]^r$. A *monomial* m in F is an element of the form $m = t^i \boldsymbol{e}_j$, where $1 \leq j \leq r$, $\boldsymbol{e}_j$ is the jth standard basis vector in F, and $i \geq 0$. A *monomial ordering* is a total ordering $>$ on the collection of monomials that satisfies $t^i \boldsymbol{e}_j > \boldsymbol{e}_j$ for all j and all $i > 0$, and that is compatible with the module structure in the sense that

$$m_1 > m_2 \Rightarrow t^i m_1 > t^i m_2$$

for all $i \geq 0$.

For modules F over the one-variable polynomial ring, there are just two basic ways to define monomial orders. First, we choose an ordering on the $\boldsymbol{e}_j$ themselves; this is usually done implicitly with the indexing of the standard basis

$$\boldsymbol{e}_1 > \boldsymbol{e}_2 > \cdots > \boldsymbol{e}_r \ .$$

Then we can define one ordering by placing more importance on the index of the basis vector appearing than on the exponent of t.

(II.B.1) Definition: The *position over term* (or POT) ordering on $F = \boldsymbol{F}_q[t]^r$ is defined by

$$t^i \boldsymbol{e}_j >_{POT} t^k \boldsymbol{e}_\ell$$

if $j < \ell$, or $j = \ell$ and $i > k$.

A second ordering is obtained if we place more importance on the exponent.

(II.B.2) Definition: The *term over position* (or TOP) ordering on $F = \boldsymbol{F}_q[t]^r$ is defined by

$$t^i \boldsymbol{e}_j >_{TOP} t^k \boldsymbol{e}_\ell$$

if $i > k$, or $i = k$ and $j < \ell$.

It is also possible to partition the standard basis into subsets and define product TOP orderings, etc., but the possibilities here are much more restricted than for modules over polynomial rings in several variables, because the degree ordering on the powers of t is the only monomial ordering on the rank-1 module $\boldsymbol{F}_q[t]$.

Once we have chosen a monomial ordering $>$, each element f of F will have a unique largest or *leading term*, which we will denote by $LT_>(f)$, or simply $LT(f)$ when no confusion as to the ordering intended is possible. Given a submodule $E \subseteq F$, the set of all leading terms of elements of E generates a submodule of F, which we will denote by $LT(E)$.

(II.B.3) Definitions: A *Gröbner basis* for a submodule $E \subseteq F$ (with respect to an ordering $>$) is a set $\mathcal{G} = \{g_1, \cdots, g_s\} \subset E$ such that $\{LT(g_1), \cdots, LT(g_s)\}$ generates the submodule $LT(E)$. Equivalently, $\mathcal{G}$ is a Gröbner basis for E if the leading term of each element of E is a multiple of the leading term of one of the g_i. The monomials in $LT(E)$ are

called the *nonstandard monomials* for E; the monomials in the complement of $LT(E)$ are the *standard monomials*.

As in the case of ideals, a Gröbner basis for a submodule $E \subseteq F$ is a generating set for E. Gröbner bases exist for every submodule $E \subseteq F$. Moreover, Buchberger's algorithm for ideal Gröbner bases (see, e.g., [18, ch. 2, sec. 7]) extends more or less verbatim to this new situation and gives *one* constructive way to compute Gröbner bases for modules. For modules over the polynomial ring in one variable, the computation amounts to several polynomial GCD calculations by the Euclidean algorithm.

One special feature of modules over $\boldsymbol{F}_q[t]$ (something that is definitely not true for modules over polynomial rings in several variables) is the following uniform bound on the size of a Gröbner basis, the direct analog of the fact that all ideals in the ring itself are principal.

(II.B.4) Proposition: Let E be any submodule of the free module $\boldsymbol{F}_q[t]^r$. Then E has a Gröbner basis $\mathcal{G}$ with the property that for each j, $j = 1, \cdots, r$, there is at most one element of $\mathcal{G}$ whose leading monomial is of the form $t^i \boldsymbol{e}_j$. In particular, E has a Gröbner basis containing at most r elements.

Proof: Consider any Gröbner basis $\mathcal{G} = \{g_1, \cdots, g_s\}$, and assume that among the elements of $\mathcal{G}$, there are two, g_i and g_k, with $LT(g_i) = t^u \boldsymbol{e}_j$ and $LT(g_k) = t^v \boldsymbol{e}_j$ for the same j. Without loss of generality, suppose $u \leq v$. Then $LT(g_k)$ is a multiple of $LT(g_i)$ and the leading terms of $\mathcal{G}' = \mathcal{G} \setminus \{g_k\}$ generate the same submodule of $\boldsymbol{F}_q[t]^r$ as the leading terms of $\mathcal{G}$. It follows from the Definitions (II.B.3) that $\mathcal{G}'$ is also a Gröbner basis for E with fewer elements than $\mathcal{G}$. We may repeat this argument on $\mathcal{G}'$ as long as there are pairs of elements whose leading terms contain the same basis vector; eventually we obtain a Gröbner basis, the leading terms of whose elements contain distinct $\boldsymbol{e}_j$'s. There are at most r remaining elements at this point. △

(II.B.5) Remark: In the applications we consider, including the $q_i = (t^{|O_i|} - 1)\boldsymbol{e}_i$ as generators to simulate the cyclic permutations as in (II.A.6) will imply that the Gröbner bases contain exactly r elements.

Also as in the case of ideals, there is a module *normal form* or *division* algorithm with respect to a Gröbner basis $\mathcal{G}$, which rewrites an arbitrary element $f \in F$ as

$$f = a_1 g_1 + \cdots + a_s g_s + \overline{f}^{\mathcal{G}} \tag{8}$$

where $a_1 g_1 + \cdots + a_s g_s \in E$, and $\overline{f}^{\mathcal{G}}$ (the normal form, or remainder on division) is a linear combination of standard monomials. The normal form is uniquely determined by f, and the choice of ordering. Expression (8) is found starting from f by repeatedly subtracting multiples of the g_i to cancel the leading term of the intermediate dividend, or moving leading terms which cannot be so canceled into the remainder. The formal statement of the general algorithm is exactly the same as for ideals (see, e.g., [18, ch. 2, sec. 1]), so we will not reproduce it here.

If we use the POT ordering (II.B.1), the steps in the normal form algorithm can be organized as several ordinary one-variable polynomial divisions, so we indicate one reasonably efficient way to implement the algorithm in this case. By (II.B.4), any submodule $E \subseteq F = \boldsymbol{F}_q[t]^r$ has a POT Gröbner basis $\mathcal{G} = \{g_1, \cdots, g_s\}$ with $s \leq r$, and where the leading terms of the g_j contain distinct standard basis vectors. We may order the g_j so that their leading terms are listed in decreasing POT order. Given

$$f = \sum f_i \boldsymbol{e}_i$$

to compute the POT normal form, we may proceed as follows. If $LT(g_1)$ contains $\boldsymbol{e}_1$, and

$$g_1 = \sum g_{1i} \boldsymbol{e}_i$$

then we begin by dividing g_{11} into f_1 in $\boldsymbol{F}_q[t]$

$$f_1 = a_1 g_{11} + R_1$$

where R_1 is zero or has smaller degree than g_{11}. Subtracting $a_1 g_1 + R_1 \boldsymbol{e}_1$ from f, and moving $R_1 \boldsymbol{e}_1$ into the remainder, we obtain the intermediate dividend p, and the partial remainder R

$$\begin{aligned} p &= \sum_{i=2}^{r} (f_i - a_1 g_{1i}) \boldsymbol{e}_i \\ R &= R_1 \boldsymbol{e}_1 \end{aligned} \tag{9}$$

Note that p contains no $\boldsymbol{e}_1$ terms, so we may continue and divide g_j, $j \geq 2$ into p. If at any step there are nonzero terms in the intermediate dividend containing a standard basis vector that appears in no $LT(g_j)$, then we move that whole component of the intermediate dividend into the remainder.

The following proposition gives a more precise statement of the resulting POT-normal form algorithm.

(II.B.6) Proposition: The following algorithm computes the POT normal form of $f \in \boldsymbol{F}_q[t]^r$ with a respect to a Gröbner basis $\mathcal{G}$ as in (II.B.4). As before we write

$$f = \sum f_i \boldsymbol{e}_i$$

and

$$g_j = \sum g_{ji} \boldsymbol{e}_i$$

etc., for the component decompositions of elements of $\boldsymbol{F}_q[t]^r$.

Input: f, the ordered POT Gröbner basis $\mathcal{G}$
Output: $a_1, \cdots, a_s$, $R = \overline{f}^{\mathcal{G}}$
Uses: Quot, Rem procedures for one-variable polynomial division

$p := f; R := 0; j := 1$
For $i = 1$ to r do
  If $LT(g_j)$ contains $\boldsymbol{e}_i$ then
    $a_i := \text{Quot}(p_i, g_{ji})$
    $R_i := \text{Rem}(p_i, g_{ji})$
    $p := p - a_i g_j - R_i \boldsymbol{e}_i$
    $R := R + R_i \boldsymbol{e}_i$
    $j := j + 1$
  Else
    $a_i := 0$
    $R := R + p_i \boldsymbol{e}_i$
    $p := p - p_i \boldsymbol{e}_i$.

Proof: The correctness of the algorithm follows by considering the values of p and R after each pass through the For loop. For instance, after the completion of the first pass through the loop (whether $LT(g_1)$ contains $\boldsymbol{e}_1$ or not), by the one-variable polynomial division algorithm and (9), we have

$$f = p + a_1 g_1 + R$$

p contains no $\boldsymbol{e}_1$ terms, and R contains only standard monomials. Then we apply induction on r to conclude the proof. △

By way of contrast, the TOP normal form algorithm would work degree by degree rather than component by component.

A *reduced Gröbner basis* is one for which the leading coefficients of all of the basis elements are 1, and the leading monomials of each of the g_i appear only in that basis element. A reduced Gröbner basis can always be computed from a given Gröbner basis by replacing each basis element by its normal form with respect to the others, and adjusting scalars. It may be shown that once we choose a monomial ordering each $\boldsymbol{F}_q[t]$-module has a unique reduced Gröbner basis. In our examples in Section III, we will give reduced Gröbner bases.

C. The Systematic Encoding Algorithm.

We return to the setting and the standing assumptions of Subsection II-A. By the alternate description (II.A.6) of the $\boldsymbol{F}_q[t]$-module structure on the code $C_L(D, G)$, we may carry out all calculations in the submodule $\overline{C}$ of $\boldsymbol{F}_q[t]^r$ generated by the r-tuples $(h_1(t), \cdots, h_r(t))$ as in (4), and the $q_i = (t^{|O_i|} - 1)\boldsymbol{e}_i$. (Recall, r here is the number of the orbits of $\mathrm{supp}\,(D)$ under the action of the group of automorphisms generated by σ.) As a preprocessing step, a monomial order is specified, and a reduced Gröbner basis $\mathcal{G}$ for $\overline{C}$ is computed. By (II.B.4), $\mathcal{G}$ will contain r elements. (Note that some of the q_i may appear as elements of $\mathcal{G}$.)

(II.C.1) Proposition: Given the Gröbner basis $\mathcal{G}$, the information positions and parity check positions for $C_L(D, G)$ are determined as follows:

a) The information positions are the coefficients of the *nonstandard* monomials appearing in the r-tuples constructed from the codewords $(h_1(t), \cdots, h_r(t))$ as in (4). In other words, the information positions are the coefficients of the nonstandard monomials of the form $t^\ell \boldsymbol{e}_i$, where $\ell \leq |O_i| - 1$; higher powers of t are ignored.

b) The parity check positions are the *standard monomials*.

Proof: This follows immediately from the Alternate Description (II.A.6) of the module structure on $C_L(D, G)$, and the normal form or division algorithm for modules. In particular, the number of nonstandard monomials as in a) will always precisely equal the dimension k of the code. △

Fix some enumeration of the information positions. Perhaps the most natural way to do this is to list the nonstandard monomials appearing in the $(h_1(t), \cdots, h_r(t))$ as in (4) in decreasing order according to the chosen monomial ordering: $m_\ell = t^{i_\ell} \boldsymbol{e}_{j_\ell}$ for $\ell = 1, \cdots, k = \dim(C)$. Do the same for the parity checks. Given $h = (h_1(t), \cdots, h_r(t))$ as in (4), let $VC(h)$ be the vector of coefficients of the terms of h, listed in any convenient order. (We might use the POT order, (II.A.1) for example, to match the cyclic block (orbit) decomposition of the codewords.) We are now ready to give the systematic encoding algorithm for one message word $w \in \boldsymbol{F}_q^k$. This algorithm uses the normal form or division algorithm described in (8) as a subroutine. As noted in (II.B.6), if the POT order is used, the normal form computation can be organized as several ordinary one-variable polynomial divisions.

(II.C.2) Systematic Encoding Algorithm:

Input: the Gröbner basis $\mathcal{G}$, $w \in \boldsymbol{F}_q^k$, $\{m_\ell\}$
Output: $E(w) \in C_L(D, G)$
Uses: Normal form algorithm with respect to given ordering

$$f := \sum_{i=1}^{k} w_i m_i$$

$$\overline{f} := f^{\mathcal{G}} \text{ (see (8) above)}$$

$$E(w) := VC(f - \overline{f}).$$

Since f is a linear combination of only nonstandard monomials, and $\overline{f}$ is a linear combination of only standard monomials, the symbols from w are not changed in the process of computing $E(w)$. By (8), the difference $f - \overline{f}$ is an element of the submodule $\overline{C}$, so it represents a codeword.

As we mentioned in the Introduction, the amount of information that needs to be stored here is generally much smaller than the amount needed for a full $\boldsymbol{F}_q$-basis of the code $C_L(D, G)$. If we use only reduced Gröbner bases, then each Gröbner basis element consists of a nonstandard leading term, and (at most) $n - k$ standard terms, whose coefficients constitute the description of the encoder. There are at most $r \cdot (n - k)$ of these coefficients. In practice, we frequently need even less information, since some parity check positions may come before some information positions in the monomial order, and hence may not appear in all of the basis elements. In addition, some elements of the Gröbner basis may be the generators $q_i = (t^{|O_i|} - 1)\,\boldsymbol{e}_i$, which contain only one nonzero standard term.

When we compare this count with the number of entries of the block B in the row-reduced generator matrix in (1), we see that we have achieved a more compact systematic encoder. Perhaps surprisingly, our method is also comparable in efficiency to the matrix-based method described in the Introduction. The reductions needed to find the normal form $\overline{f}$ do not require significantly more arithmetic than the direct computation of the full set of parity checks via the row-reduced generator matrix (1). To see this note that for $w \in \boldsymbol{F}_q^k$, in computing the matrix product $w[I_k \mid B]$ directly, we must perform $k \cdot (n - k)$ multiplications and $(k - 1) \cdot (n - k)$ sums. On the other hand, reducing

$$f := \sum_{i=1}^{k} w_i m_i$$

to normal form, we need to subtract k multiples of Gröbner basis elements to remove the nonstandard monomials. Each of those multiples will be a constant times a monomial times a

Gröbner basis element, and each Gröbner basis element contains (at most) $n-k$ nonzero standard coefficients (assuming a reduced Gröbner basis). So the total amount of arithmetic is roughly the same.

III. Examples

In this section, we will present a series of examples illustrating how well our method works on several interesting classes of codes. As will be clear, these examples by no means exhaust the situations where the method will be useful.

(III.1) Practical Comment: We mention that by (II.B.4), the greatest reduction in the amount of information needed to specify the systematic encoder will be achieved when we minimize the number of orbits of $\mathrm{supp}\,(D)$ under the action of the subgroup generated by σ. In other words, when several choices of σ are possible, it will generally pay to use the σ of *maximal order*.

(III.2) Examples: One large class of examples has a common pattern. Namely, suppose X is a smooth curve in *special position*, as defined in [15]. In particular, $X \subset \boldsymbol{P}^m$ has just one point Q at infinity (which is rational over $\boldsymbol{F}_q$), and the affine coordinate functions x_i, $i = 1, \cdots, m$ have distinct pole orders at Q. (As proved in [15], given a curve Y and an $\boldsymbol{F}_q$-rational point on Y, there always is a birational isomorphism from Y to a curve in special position, so there is no real restriction here.) Suppose in addition that X has a *monomial automorphism*, that is, an automorphism of the form $\sigma : x_i \mapsto \alpha^{r_i} x_i$, $(i = 1, \cdots, m)$ where α is a generator of $\boldsymbol{F}_q^*$, and the r_i are integers. Then σ fixes any divisor of the form aQ, and permutes the affine $\boldsymbol{F}_q$-rational points of X. Let D be the sum of the affine $\boldsymbol{F}_q$-rational points, each with coefficient 1. By Proposition (II.A.4), any code of the form $C_L(D, aQ)$ has an $\boldsymbol{F}_q[t]$-module structure, and our approach applies.

By the results of [15], these codes can also be obtained by shortening m-dimensional extended cyclic codes. Extended cyclic codes have the structure of $\boldsymbol{F}_q[t_1, \cdots, t_m]$-modules. Interestingly enough, in the case of a monomial automorphism, the $\boldsymbol{F}_q[t]$ module-structure guaranteed by (II.A.4) is compatible with the $\boldsymbol{F}_q[t_1, \cdots, t_m]$-module structure of the extended cyclic code in the sense that the multiplication by t we have defined is the restriction to the shortened code of multiplication by the monomial $t_1^{r_1} t_2^{r_2} \cdots t_m^{r_m}$ in the extended cyclic code. (The r_i are the same exponents appearing in the automorphism σ.)

As a concrete example, we consider the codes $C_L(D, aQ)$ on the Hermitian curve X_3 over $\boldsymbol{F}_9$. As in Example (II.A.3), we represent $\boldsymbol{F}_9$ as $\boldsymbol{F}_3[\alpha]/\langle \alpha^2 + \alpha - 1 \rangle$ and write each nonzero element of the field as a power of α. Under the action of the subgroup of $\mathrm{Aut}\,(X_3)$ generated by the monomial automorphism σ

$$\sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^4 y = -y \end{cases}$$

the affine $\boldsymbol{F}_9$-rational points decompose into five orbits. Using the orbit decomposition, we can rewrite the codewords of $C_L(D, aQ)$ as 5-tuples of polynomials as in (4). There will be a Gröbner basis for $C_L(D, aQ)$ as $\boldsymbol{F}_9[t]$-module with *five* elements in this case.

We will derive a complete compact systematic encoder for $C_L(D, 19Q)$ on this curve. To begin, we note that $g(X_3) = 3$, so that by the Riemann–Roch theorem, the dimension of $L(19Q)$ is $19 + 1 - 3 = 17$. No function in $L(19Q)$ vanishes at all 27 of the points of D, so $n = 27, k = 17$ for this code. (By [3, sec. VII.4.3], the minimum distance is $d = 8$.)

Since x has pole order 3 at Q and y has pole order 4 at Q, a basis for $L(19Q)$ is given by the following collection of monomial functions:

$$\{1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, x^3y, x^2y^2, xy^3, y^4, x^3y^2, x^2y^2, xy^3\}\ .$$

Using the POT ordering (II.B.1) on $\boldsymbol{F}_9[t]^5$, we compute a Gröbner basis for the submodule $\overline{C}$ of $\boldsymbol{F}_9[t]^5$. We may use as generators, for example, the 17 codewords corresponding to the rows of any generator matrix for $C_L(D, 19Q)$, and $(t^8 - 1)\boldsymbol{e}_i$, $i = 1, 2, 3$, $(t^2 - 1)\boldsymbol{e}_4$, and $(t - 1)\boldsymbol{e}_5$. We find the following reduced Gröbner basis:

$$g_1 = (1, \alpha^6, \alpha t^5 + \alpha t^4 + \alpha^6 t^3 + \alpha^2 t^2 + \alpha t + \alpha^2, \alpha^2 t + \alpha, 1)$$
$$g_2 = (0, t + \alpha^5, t^5 + \alpha^5 t^4 + \alpha^7 t^3 + \alpha^7 t + \alpha^7, \alpha^2 t + \alpha^4, 1)$$
$$g_3 = (0, 0, t^6 + \alpha^6 t^5 + \alpha^2 t^4 + \alpha^7 t^3 + \alpha t^2 + \alpha^4 t + \alpha^5, \alpha^3 t + \alpha^3, \alpha^7)$$
$$g_4 = (0, 0, 0, t^2 - 1, 0)$$
$$g_5 = (0, 0, 0, 0, t - 1)$$

From this, using (II.C.1) we see that the information positions for the code (with respect to the POT ordering) can be taken as the coefficients of

- $t^7\boldsymbol{e}_1, \cdots, t\boldsymbol{e}_1, \boldsymbol{e}_1$
- $t^7\boldsymbol{e}_2, \cdots, t\boldsymbol{e}_2$
- $t^7\boldsymbol{e}_3, t^6\boldsymbol{e}_3$.

There are 17 of these, which agrees with the dimension k.

(III.3) Example: We continue with the Hermitian curve X_3. By the heuristic of (III.1), the monomial automorphism described above is not optimal in this case. Indeed, as we have seen in (II.A.3), when $m = 3$, the Hermitian curve X_3 also has a nonmonomial automorphisms of order higher than 8. We will consider next the automorphism

$$\tau : \begin{cases} x \mapsto \alpha^2 x \\ y \mapsto y + \alpha^2. \end{cases}$$

From Example (II.A.3), we recall that the affine $\boldsymbol{F}_9$-rational points of X_3 decompose into *three* orbits under the action of τ—two of length 12 and one of length 3.

Using the same basis for $L(19Q)$ given above and the POT ordering (II.B.1) on $\boldsymbol{F}_9[t]^3$, we compute a Gröbner basis for the submodule $\overline{C}$ of $\boldsymbol{F}_9[t]^3$. As before, we may use as generators the 17 codewords corresponding to the rows of a generator matrix for $C_L(D, 19Q)$ (rearranged according to the orbits of τ) and $(t^{12} - 1)\boldsymbol{e}_i$, $i = 1, 2$, $(t^3 - 1)\boldsymbol{e}_3$.

As we expect by (II.B.4), there are three Gröbner basis elements in all:

$$g_1 = (1, \alpha^3 t^6 + \alpha^7 t^4 + \alpha^7 t^3 + t^2 + \alpha^6 t + \alpha, \alpha^5 t^2 + t + \alpha)$$
$$g_2 = (0, t^7 + \alpha^3 t^6 + \alpha^5 t^5 + \alpha^4 t^4 + \alpha^4 t^3 + \alpha^7 t^2 + \alpha t + 1, \alpha^2 t + \alpha^6)$$
$$g_3 = (0, 0, t^3 - 1) \qquad (10)$$

From this, using (II.C.1) we see that the information positions for the code (with respect to the POT ordering) are the coefficients of

- $t^{11}\boldsymbol{e}_1, \cdots, t\boldsymbol{e}_1, \boldsymbol{e}_1$,
- $t^{11}\boldsymbol{e}_2, \cdots, t^7\boldsymbol{e}_2$.

Note that there are precisely 17 of them as we expect. The remaining ten monomials of the form $t^\ell \boldsymbol{e}_i$, where $\ell \leq |O_i| - 1$, are the parity check positions.

For purposes of comparison, here is a reduced TOP Gröbner basis for the same submodule of $\boldsymbol{F}_9[t]^3$, listed with leading terms in decreasing TOP order (the leading terms are underlined).

$$\begin{aligned} g_1' &= (-t^2 + \alpha^2 t + \alpha^7, \underline{t^4} + \alpha^7 t^3 + \alpha^6 t^2 + \alpha^7 t - 1, \alpha^7 t + \alpha^5) \\ g_2' &= (\underline{t^3} + \alpha^7 t^2 + \alpha^2 t - 1, \alpha^7 t^3 - t^2 + \alpha^3 t, \alpha^7 t^2 + \alpha^7 t + \alpha) \\ g_3' &= (0, 0, \underline{t^3} - 1) \end{aligned} \tag{11}$$

The information and parity check positions are different here in (11) than they are for the POT order basis (10) above. However, note that there are still 17 information positions—the coefficients of

- $t^{11}\boldsymbol{e}_1, \cdots, t^3\boldsymbol{e}_1$,
- $t^{11}\boldsymbol{e}_2, \cdots, t^4\boldsymbol{e}_2$.

In general, using the POT ordering will tend to eliminate more terms from the leftmost components, leaving the parity checks farther to the right, while the TOP ordering would tend to spread the parity checks more evenly among the different components.

To illustrate the operations needed for encoding, we apply the algorithm (II.C.2) to encode the message word

$$w = (t, \alpha t^8 + t^7, 0)$$

using the POT Gröbner basis in (10) and (II.B.6) (we have already converted the string from $\boldsymbol{F}_9^{17}$ to a 3-tuple of polynomials). We begin by subtracting tg_1 from w to obtain

$$w - tg_1 = (0, \alpha t^8 + \alpha^6 t^7 + \alpha^3 t^5 + \alpha^3 t^4 - t^3 + \alpha^2 t^2 + \alpha^5 t, \alpha t^3 - t^2 + \alpha^5 t).$$

Now, we divide the leading entry from g_2 into the second component of the right-hand side, yielding a quotient of $\alpha t + \alpha$, and a remainder of

$$R_2\boldsymbol{e}_2 = (0, \alpha^3 t^6 + \alpha^5 t^5 + \alpha^6 t^4 + \alpha^7 t^3 - t^2 + \alpha^3 t + \alpha^5, 0).$$

Hence

$$w - tg_1 - (\alpha t + \alpha)g_2 - R_2\boldsymbol{e}_2 = (0, 0, \alpha t^3 + \alpha t^2 + \alpha^5 t + \alpha^3)$$

dividing by g_3 (or simply "cycling back" to get exponents in the range $0, 1, 2$ in the last component), we obtain

$$R_3\boldsymbol{e}_3 = (0, 0, \alpha t^2 + \alpha^5 t - 1).$$

The normal form is $\overline{w} = (0, R_2, R_3)$, and the corresponding codeword is

$$w - \overline{w} = (t, \alpha t^8 + t^7 + \alpha^7 t^6 + \alpha t^5 + \alpha^2 t^4 + \alpha^3 t^3 + t^2 + \alpha^7 t + \alpha, \alpha^5 t^2 + \alpha t + 1).$$

(III.4) Examples: All of the $C_L(D, aQ)$ codes on Hermitian curves $x^{m+1} = y^m + y$ over $\boldsymbol{F}_{m^2}$ can be systematically encoded in a very similar fashion. We keep the same notation used above for X_3; in particular D will be the sum of the m^3 affine $\boldsymbol{F}_{m^2}$-rational points of X_m.

Under the action of the automorphism

$$\sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^{m+1} y \end{cases}$$

the point Q at infinity is fixed, and the points of D are permuted. There are precisely m orbits of length $m^2 - 1$: the orbits of the points with both coordinates nonzero. (A convenient set of representatives for these orbits are the affine $\boldsymbol{F}_{m^2}$-rational points on the line $x = 1$.) The points with $x = 0$ and $y \neq 0$ form one further orbit as in (II.A.3). The origin is a singleton orbit. This gives a total of $m + 2$ orbits.

By (II.B.4) there is a Gröbner basis $\mathcal{G}$ for the $\boldsymbol{F}_{m^2}[t]$-module $C_L(D, aQ)$ consisting of $m+2$ elements, and knowing $\mathcal{G}$ gives a compact representation of the systematic encoder. We note that on these curves, for large m, the interesting long Hermitian codes will have $n = m^3$, a will be on the order of m^3, and $k = a + 1 - m(m-1)/2$ will also be on the order of m^3. The number of parity checks is on the order of m^2, so the larger we take m, the greater the gain is in compactness of representation of the encoder: on the order of m^5 coefficients in the row-reduced generator matrix for the code, versus on the order of m^3 coefficients in the Gröbner basis elements. In a related paper [10], we have developed a very efficient specialized algorithm for computing these Gröbner bases. This algorithm uses the special properties of Hermitian curves in a decisive way.

When m is prime, this can be improved slightly. Namely, as in (II.C.1) and (III.3), when m is prime, X_m will have a nonmonomial automorphism of the form

$$\tau : \begin{cases} x \mapsto \alpha^{m-1} x \\ y \mapsto y + \alpha^r \end{cases}$$

where

$$\alpha^{r(m-1)} = -1 \in \boldsymbol{F}_{m^2}.$$

τ has order $m(m+1)$, and the affine $\boldsymbol{F}_{m^2}$-rational points of X_m decompose into m orbits under the action of τ.

Finally, we note that the Hermitian codes we have considered can also be viewed as ideals in the group algebra of a certain nonabelian group Γ of order $m^3(m^2 - 1)$—the full subgroup of $\mathrm{Aut}(X_m)$ fixing the divisors D and aQ. This follows since Γ acts transitively on the points of D. We have not used the full group Γ here, only large cyclic subgroups. However, the construction of a systematic encoder seems more accessible by our approach.

(III.5) Examples: Another interesting class of curves for which our approach applies immediately, and with a monomial automorphism as well, are the curves of the form

$$x^q + x = y^{q_0}(y^q + y) \tag{12}$$

over the field $\boldsymbol{F}_q$, where $q = 2^{2n+1}$ for some $n \geq 1$, and $q_0 = 2^n$, studied by Hansen and Stichtenoth in [7]. We will write Y_n for the projective closure of the curve (12). Y_n has

just one (singular) point $(x, y, z) = (1, 0, 0)$ at infinity. The Y_n also have large automorphism groups and large numbers of points rational over $\boldsymbol{F}_q$. Indeed, it is easy to see from the form of (12) that Y_n passes through *every* point of the affine plane over $\boldsymbol{F}_q$, and achieves the maximum number of $\boldsymbol{F}_q$-rational points for a curve of its genus $g = q_0(q-1)$ allowed by the explicit formulas of Weil (see [7]). We note that Y_n always has an automorphism of the form

$$\sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^r y \end{cases} \tag{13}$$

where α is a generator of $\boldsymbol{F}_q^*$, and r is chosen to satisfy the congruence

$$(q_0 + 1)r \equiv 1 \bmod q - 1$$

(Note that

$$q_0(q-1) + 1 = 2^n(2^{2n+1} - 1) + 1$$

is always divisible by $q_0 + 1 = 2^n + 1$, so such r always exist.)

Taking Q to be the point at infinity, and D to be the sum of the q^2 affine points of Y_n, we see that the points of D are permuted under σ. There are $q-1$ orbits of points where neither coordinate is zero. The nonzero points with $y = 0$ are permuted cyclically giving another orbit. The origin is a singleton orbit, and since r is relatively prime to $q-1$, the remaining points with $x = 0$ form one orbit also. Thus there are $q + 2$ orbits in all. Bases for the vector spaces $L(aQ)$ can be determined from the results of [7]. (They can be expressed as collections of monomials in four variables: x, y and two additional functions f and g, whose pole orders at Q, together with the orders of x and y, generate the semigroup of pole numbers at Q.) By (II.B.4), a Gröbner basis for the $\boldsymbol{F}_q[t]$-module $C_L(D, aQ)$ will contain $q + 2$ elements.

As in the case of the Hermitian curves considered in (III.3), (III.4), the codes $C_L(D, aQ)$ can also be considered as ideals in the group algebra of a nonabelian group. As shown in [7, Proposition 3.2], there is a group S consisting of q^2 automorphisms of Y_n of the form

$$\tau_{\beta,\gamma} : \begin{cases} x \mapsto x + \beta^{q_0} y + \gamma \\ y \mapsto y + \beta \end{cases}$$

($\beta, \gamma \in \boldsymbol{F}_q$) acting transitively on the q^2 $\boldsymbol{F}_q$-rational points of Y_n. So the $C_L(D, aQ)$ codes on Y_n are ideals in the group algebra $\boldsymbol{F}_q[S]$. However, every element of S has order 4 or less, so by (III.1) our choice of automorphism σ in (13) leads to a greater reduction in the amount of information needed to specify encoders than any choice of automorphism in S.

(III.6) Examples: Another very large class of examples for which our approach applies are the hyperelliptic curves (including those of genus 1, usually known as elliptic curves). Codes constructed from elliptic curves in characteristic 2 and 3 have been studied, for example by Driencourt and Michon in [20].

For simplicity of notation, we work first over a field of odd characteristic. Then a hyperelliptic curve is given by an affine equation of the form

$$y^2 = f(x) \tag{14}$$

where $f(x)$ is a polynomial with distinct roots in the algebraic closure of $\boldsymbol{F}_q$. We restrict to the case where $f(x)$ has odd degree $d = 2e + 1$. Then the curve has genus $g = e$, and is in special position. For $e \geq 1$, the single point Q at infinity is a cuspidal singularity. The mapping

$$\sigma : \begin{cases} x \mapsto x \\ y \mapsto -y \end{cases}$$

is an automorphism of the curve (14), called the *hyperelliptic involution*. In characteristic 2, we have the alternate form

$$y^2 + y = f(x)$$

for the equation, and an automorphism

$$\sigma : \begin{cases} x \mapsto x \\ y \mapsto y + 1. \end{cases}$$

Unlike the situation in the previous examples, the full automorphism group of many hyperelliptic curves of genus $g > 1$ equals the cyclic group generated by the hyperelliptic involution.

Q is fixed by σ, and the other $\boldsymbol{F}_q$-rational points are permuted in orbits of length 1 or 2. Hence, provided that there is some $\boldsymbol{F}_q$-rational point not fixed by σ, we may set D to be the divisor of affine points on (14), and apply our approach to the codes $C_L(D, aQ)$ constructed from these curves as well. (But of course, the savings in the compact systematic encoder will not be as drastic as in the previous examples.)

(III.7) Examples: Finally, we want to discuss an example where the curve has several points at infinity, but where the same ideas may be applied. We will consider the celebrated Klein quartic curve K

$$x^3y + y^3z + z^3x = 0$$

over the field $\boldsymbol{F}_8$. K is smooth of degree 4, hence has genus 3. Its full automorphism group is simple of order 168. Codes constructed from K have been discussed, for example, by Hansen (see [6], and the discussion in [21, sec. 5.7.5]).

We write

$$\boldsymbol{F}_8 = \boldsymbol{F}_2[\alpha]/\langle \alpha^3 + \alpha + 1 \rangle.$$

As is well known, K has 24 points rational over $\boldsymbol{F}_8$: the three points

$$Q_0 = (1, 0, 0), \quad Q_1 = (0, 1, 0), \quad Q_2 = (0, 0, 1)$$

and 21 other points P_{ij} obtained by applying the elements of the subgroup F of $\mathrm{Aut}(K)$ generated by

$$\sigma : \begin{cases} x \mapsto \alpha x \\ y \mapsto \alpha^2 y \\ z \mapsto \alpha^4 z \end{cases}$$

and

$$\tau : \begin{cases} x \mapsto y \\ y \mapsto z \\ z \mapsto x \end{cases}$$

to the point $P_{00} = (1, \alpha^2, \alpha^2 + \alpha)$. Precisely

$$P_{ij} = \tau^i(\sigma^j(P_{00})) \, .$$

The points Q_i are permuted among themselves by F, and the other 21 points form a single F-orbit. Hansen considers codes constructed with

$$G_m = m(Q_0 + Q_1 + Q_2)$$

and

$$D = \sum_{ij} P_{ij}.$$

Since the points of D form a single orbit under F; the codes $C_L(D, G_m)$ can be considered as ideals in the group algebra $\boldsymbol{F}_8[F]$. Moreover, since the order of F is relatively prime to the characteristic, $\boldsymbol{F}_8[F]$ is a semi-simple algebra, and the ideals are principal (generated by idempotent elements).

However, F is a nonabelian group of order 21, hence our Gröbner basis approach would not apply directly. (Gröbner bases in a polynomial ring with noncommuting variables satisfying the same relation as σ and τ

$$\sigma\tau = \tau\sigma^4$$

could be applied.) However, we may also simply restrict our attention to the cyclic subgroup generated by σ (according to the heuristic of (III.1); this will be the better choice). Under the action of the cyclic groups $H = \langle\sigma\rangle$ of order 7, the points of D decompose into *three* orbits of size 7: $O_i = \{P_{ij} \mid j = 0, \cdots, 6\}$. By (II.B.4), for the resulting $\boldsymbol{F}_8[t]$-module structure on $C_L(D, G_m)$, there will be three elements in a Gröbner basis. Taking $m = 4$, for example, we get a code with parameters $n = 21$, $k = 10$, $d = 9$. To specify the compact systematic encoder, we need to know the 3×11 coefficients in the Gröbner basis elements. As in the previous cases, this compares very favorably with the 10×11 coefficients in the row-reduced generator matrix.

REFERENCES

[1] R. Pellikaan, B.-Z. Shen, and G. J. M. van Wee, "Which linear codes are algebraic-geometric?" *IEEE Trans. Inform. Theory,* vol. 37, no. 3, pp. 583–602, May 1991.

[2] J. H. van Lint, *Introduction to Coding Theory.* New York: Springer, 1982.

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

[4] ——, "A note on Hermitian codes over GF(q^2)," *IEEE Trans. Inform. Theory,* vol. 34, pp. 1345–1348, 1988.

[5] H. Tiersma, "Remarks on codes from Hermitian curves," *IEEE Trans. Inform. Theory,* vol. IT-33, pp. 605–609, 1987.

[6] J. P. Hansen, "Codes on the Klein quartic, ideals and decoding," *IEEE Trans. Inform. Theory,* vol. IT-33, pp. 923–925, 1987.

[7] J. P. Hansen and H. Stichtenoth, "Group codes on certain algebraic curves with many rational points," *Appl. Algebra in Eng. Commun. Comput.,* vol. 1, pp. 67–77, 1990.

[8] M. A. Tsfasman, S. G. Vlăduţ, and T. Zink, "Modular curves, Shimura curves and Goppa codes better, than the Varshamov-Gilbert bound," *Math. Nachr.,* vol. 109, pp. 21–28, 1982.

[9] W. Adams and P. Loustaunau, *An Introduction to Gröbner Bases.* Providence, RI: Amer. Math. Soc., 1994.

[10] J. Little, K. Saints, and C. Heegard, "On the structure of Hermitian codes," *J. Pure Appl. Algebra,* to appear, May 1995.

[11] S. Sakata, "A Gröbner basis and a minimal polynomial set, of a finite nD array," in *Applied Algebra, Algebraic Algorithms, and Error-correcting Codes: Proc. AAECC-8, Tokyo 1990,* S. Sakata, Ed. Berlin, Germany: Springer, 1991.

[12] A. Poli and L. Huguet, *Error Correcting Codes: Theory and Applications.* Hemel Hempstead, UK: Prentice-Hall, 1992.

[13] N. Jacobson, *Basic Algebra I.* New York: Freeman, 1985.

[14] S. C. Porter, "Decoding codes arising from goppa's construction on algebraic curves," Ph.D. dissertation, Yale University, New Haven, CT, 1988.

[15] K. Saints and C. Heegard, "Algebraic-geometric codes and multidimensional cyclic codes: Theory and algorithms for decoding using Gröbner bases," this issue, pp. 1733–1751.

[16] B. Buchberger, "Gröbner bases: An algorithmic method in polynomial ideal theory," in *Multidimensional Systems Theory: Progress, Directions and Open Problems in Multidimensional Systems,* N. K. Bose, Ed. Dordrecht, The Netherlands: D. Reidel, 1985.

[17] T. Becker and V. Weispfenning, *Gröbner Bases.* New York: Springer, 1992.

[18] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms.* New York: Springer, 1992.

[19] B. Mishra, *Algorithmic Algebra.* New York: Springer, 1993.

[20] Y. Driencourt and J. F. Michon, "Rapport sur les codes géométriques," *Compt. Rend. Acad. Sci. Paris,* vol. 301, pp. 15–17, 1985.

[21] C. Moreno, *Algebraic Curves over Finite Fields.* Cambridge, UK: Cambridge Univ. Press, 1991.