Decoding Reed-Solomon Skew-Differential Codes

José Gómez-Torrecillas, Gabriel Navarro, José Patricio Sánchez-Hernández

***Abstract*—A large class of MDS linear codes is constructed. These codes are endowed with an efficient decoding algorithm. Both the definition of the codes and the design of their decoding algorithm only require from Linear Algebra methods, making them fully accesible for everyone. Thus, the first part of the paper develops a direct presentation of the codes by means of parity-check matrices, and the decoding algorithm rests upon matrix and linear maps manipulations. The somewhat more sophisticated mathematical context (non-commutative rings) needed for the proof of the correctness of the decoding algorithm is postponed to the second part. A final section locates the Reed-Solomon skew-differential codes introduced here within the general context of codes defined by means of skew polynomial rings.**

INTRODUCTION

The treatment of cyclic linear codes as ideals of a quotient of a polynomial ring inspired the extension of cyclic-like conditions to the realm of skew-polynomial (non-commutative) rings both from the perspective of block codes [4]–[6], [11], [26] and convolutional codes [12], [14], [15], [25], [29], [30]. One of the nicest features of some (commutative) cyclic codes is the possibility of designing efficient algebraic decoding algorithms taking advantage of their rich algebraic structure [19], [28], [32]. These classic approaches have been adapted or, in some cases, inspire, decoding procedures for some families of cyclic-like codes based on non-commutative polynomial arithmetics [6], [16], [17], [24]. Dealing with these codes, presented in the language of left ideals and modules, requires a training in non-commutative rings which could limit their difusion and potential practical use among coding theorists and engineers.

In this paper we simplify and extend to a considerably broader class of codes the algebraic decoding algorithms designed in [17] and [18] for skew RS and convolutional differential RS codes, respectively. These codes were presented as left ideals of certain non-commutative polynomial rings, and their decoding algorithms make use of advanced algebraic tools like evaluation of non-commutative polynomials. In contrast, both the construction of the codes in this note, and the description and implementation of their decoding algorithm only require basic Linear Algebra over a field. The introduction of the (minimal) algebraic machinery needed to prove the correctness of the decoding algorithm is postponed till Section III. In this way, the material of Section I is ready to use even without knowing what a non-commutative ring is.

We work over a general field K, since our interests include block linear codes ($K = \mathbb{F}_q$, a finite field) and convolutional codes ($K = \mathbb{F}_q(t)$, the rational function field in one variable t over $\mathbb{F}_q$).

We call our codes Reed-Solomon (RS) skew-differential codes because they are defined from a skew derivation of the field K by means of parity check matrices (Definition 2), and they become MDS with respect to the usual Hamming metric. As linear subspaces, they are dual to some special type of linearized Reed-Solomon codes in the sense of [26], and also they may be seen as a particular case of (σ, δ)–codes from [6], with a carefully chosen left ideal generator. This choice allows the design of the algebraic decoding algorithm developed in this paper.

Each code $C_{(\varphi_u,\alpha,d)}$ depends on the three parameters reflected in the notation. Concretely, φ_u is a transformation of K, defined from an element $u \in K$ and the skew derivation, that becomes linear with respect to a suitable subfield K^{φ_u} of K, α is a cyclic vector of φ_u, and d is the designed minimum Hamming distance of the code. After the description of the code $C_{(\varphi_u,\alpha,d)}$, Section I proceeds to the design of its algebraic decoding algorithm (see Algorithm 1). It runs at follows: from a word corrupted by up to $\tau = \lfloor \frac{d-1}{2} \rfloor$ errors, a matrix is computed recursively from the syndromes. The left kernel of this matrix contains a nonzero vector ρ (Proposition 3). With this vector at hand, a second matrix L is recursively computed. This matrix gives an easy procedure to get the positions of the errors (Theorem 4). Once these positions are known, the values of the errors are the solution of a linear system of equations.

Section II deals with the examples. We analyze when the finiteness condition (see Proposition 1) that grants the construction of the code $C_{(\varphi_u,\alpha,d)}$ holds (Proposition 7 and Corollary 8). Its turns out that in the cases of interest (block and convolutional skew-differential codes), the code $C_{(\varphi_u,\alpha,d)}$ is always built (Subsections II-A and II-B). Concrete examples of application of Algorithm 1 are shown in both cases. The computations are done with the help of the SageMath symbolic computation system [31].

Section III is devoted to prove the mathematical results that ground Algorithm 1. Our algebraic setup requires from the ring of additive endomorphisms of K generated by K and the map φ_u, and the application of Jacobson-Bourbaki's Correspondence to identify which maps φ_u are suitable for constructing skew-differential codes (Proposition 11). Jacobson-

José Gómez-Torrecillas is with IMAG and Department of Algebra of University of Granada.

Gabriel Navarro is with CITIC and Department of Computer Science and Artificial Intelligence of University of Granada.

José Patricio Sánchez-Hernández is with Department of Algebra of University of Granada.

Research supported by grants A-FQM-470-UGR18 from Junta de Andalucía and FEDER and PID2019-110525GB-100 from AEI and FEDER The third author was supported by The National Council of Science and Technology (CONACYT) with a scholarship for a Postdoctoral Stay in the University of Granada.

Manuscript received May XXXX, 20XX; revised XXXXX XX, 20XX.

This paper was presented in part at *Quadratic Forms, Rings and Codes (Universit de Lens, July, 8th, 2021).*

Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Bourbaki's Correspondence is a more general version of Galois Correspondence that applies to additive maps (like φ_u) which are not necessarily field automorphisms (see [33] or [34]). We also use the non-commutative Wronskian and Vandermonde matrices investigated in [9]. Indeed, it turns out that $\mathcal{R}$ is isomorphic to the factor ring of a skew polynomial ring by the minimal polynomial of φ_u (see Proposition 14). This isomorphism eases the conceptual manipulation of the elements of $\mathcal{R}$ in Proposition 17 and Theorem 18. The section is closed with the proof of the correctness of Algorithm 1.

Section IV gives a precise description of an RS skew-differential code $C_{(\varphi_u,\alpha,d)}$ as a left ideal of $\mathcal{R}$ (Corollary 25). From the practical point of view, this serves to see in detail how the codes investigated in [17] and [18] are obtained as particular cases of $C_{(\varphi_u,\alpha,d)}$ (see Examples 28 and 29). Besides, a generator of $C_{(\varphi_u,\alpha,d)}$ is explicitly given. On the theoretical side, Corollary 25 identifies RS skew-differential codes as members of the very general family of module (σ,δ)–codes defined in [6]. In fact, we precisely characterize the module (σ,δ)–codes with "word ambient" ring $\mathcal{R}$ (Proposition 23 and Proposition 24), and describe which of these codes are RS skew-differential codes (Corollary 25), thus enjoying an efficient algebraic decoding algorithm. We also discuss (Remark 30) how the dual of an RS skew-differential code becomes a linearized skew Reed-Solomon code in the sense [26].

The Appendix contains some remarks on computational aspects both of the construction of the codes and the decoding algorithm.

I. Definition of the codes and specification of their algebraic decoding algorithm

Let K be a field. For any additive map[1] $\phi : K \to K$, set

$$K^{\phi} = \{b \in K : \phi(ab) = \phi(a)b \text{ for all } a \in K\}.$$

A straightforward argument shows that K^{ϕ} is a subfield of K and, obviously, ϕ becomes a K^{ϕ}–linear map. A tempting idea is to use good enough field extensions K/K^{ϕ} to design K–linear error corrector codes with efficient algebraic decoding algorithms. In this note, we consider additive maps on K stemming from skew derivations.

A *skew derivation* on K is a pair (σ,δ), where σ is a field automorphism of K, and $\delta : K \to K$ is an additive map subject to the condition

$$\delta(ab) = \sigma(a)\delta(b) + \delta(a)b, \tag{1}$$

for all $a, b \in K$.

Given $u \in K$, let $\varphi_u : K \to K$ be defined by

$$\varphi_u(a) = \sigma(a)u + \delta(a), \tag{2}$$

for all $a \in K$.

Proposition 1. *Assume that the dimension of K as a K^{φ_u}–vector space is $m < \infty$. The minimal polynomial of the K^{φ_u}–linear map φ_u has degree m and, henceforth, it has at least a cyclic vector. Moreover $\alpha \in K$ is such a cyclic vector if and only if the matrix*

$$A = \begin{pmatrix} \alpha & \varphi_u(\alpha) & \cdots & \varphi_u^{m-1}(\alpha) \\ \varphi_u(\alpha) & \varphi_u^2(\alpha) & \cdots & \varphi_u^{m}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{m-1}(\alpha) & \varphi_u^{m}(\alpha) & \cdots & \varphi_u^{2m-2}(\alpha) \end{pmatrix}$$

is invertible.

Proposition 1 establishes an adequate context to define our codes from the matrix A. It is worth to mention that the computation of a cyclic vector α can be randomized and does not require the computation of K^{φ_u} (see Remark 9 in Section II).

Definition 2. Given $2 \le d \le m$, define the K–linear code $C_{(\varphi_u,\alpha,d)} \subseteq K^m$ of dimension $m-d+1$ as the left kernel of the matrix

$$H = \begin{pmatrix} \alpha & \varphi_u(\alpha) & \cdots & \varphi_u^{d-2}(\alpha) \\ \varphi_u(\alpha) & \varphi_u^2(\alpha) & \cdots & \varphi_u^{d-1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{m-1}(\alpha) & \varphi_u^{m}(\alpha) & \cdots & \varphi_u^{m+d-3}(\alpha) \end{pmatrix},$$

that is, $C_{(\varphi_u,\alpha,d)} = \{w \in K^m : wH = 0\}$.

It can be proved (see Theorem 13 below) that the minimum Hamming distance of this code is d, so it is an MDS code. We will call these codes *Reed Solomon (RS) skew-differential codes*. It is worth to mention that when K is either a finite field or a rational function field over a finite field, its dimension as a K^{φ_u}–vector space is finite for any choice of the skew derivation (σ,δ), of the element u and of the cyclic vector α (see Section II). Indeed, m is always equal to the order m of the automorphism σ, when the latter is not the identity map.

Next, let us describe the decoding algorithm for $C_{(\varphi_u,\alpha,d)}$, that corrects up to $\tau = \lfloor \frac{d-1}{2} \rfloor$ errors. Suppose that we receive a word

$$y = (y_0, \ldots, y_{m-1}) \in K^m$$

with $y = c + e \in K^m$, where c is a codeword, and

$$e = (e_0, \ldots, e_{m-1})$$

is an error vector, which is assumed to be nonzero in the discussion below. Suppose that the nonzero components $e_{k_1}, \ldots, e_{k_v} \in K$ of e occur at the positions $0 \le k_1 < \cdots < k_v \le m-1$. We assume that $v \le \tau$.

We start by computing, for $i = 0, \ldots, d-2$, the syndromes

$$S_{i,0} = \sum_{j=0}^{m-1} y_j \varphi_u^{i+j}(\alpha), \tag{3}$$

which are the components of the vector yH.

For every pair i, k of nonnegative integers such that $i + k \le 2\tau - 1$ we may compute $S_{i,k} \in K$ recursively from (3) according to the rule

$$S_{i,k+1} = \sigma^{-1}(\delta(S_{i,k}) - S_{i+1,k}). \tag{4}$$

[1] That is, it satisfies that $\phi(a+b) = \phi(a) + \phi(b)$ for all $a, b \in K$.

We may thus compute the columns of the matrix

$$S = \begin{pmatrix} S_{0,0} & S_{0,1} & \cdots & S_{0,\tau-1} \\ S_{1,0} & S_{1,1} & \cdots & S_{1,\tau-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{\tau,0} & S_{\tau,1} & \cdots & S_{\tau,\tau-1} \end{pmatrix}.$$

Next, for $1 \leq r \leq \tau$, let S_r denote the matrix formed by the r first columns of S and compute

$$\theta = \max\{r : rank\, S_r = r\}.$$

Proposition 3. *The left kernel of the matrix*

$$B = \begin{pmatrix} S_{0,0} & S_{0,1} & \cdots & S_{0,\theta-1} \\ S_{1,0} & S_{1,1} & \cdots & S_{1,\theta-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{\theta,0} & S_{\theta,1} & \cdots & S_{\theta,\theta-1} \end{pmatrix}$$

is a one dimensional vector subspace of $K^{\theta+1}$ spanned by a vector $\rho = (\rho_0, \ldots, \rho_\theta)$ with $\rho_\theta \neq 0$.

The next step is the localization of the positions $k_1, \ldots, k_v \in \{0, \ldots, m-1\}$ at which the error values $e_{k_1}, \ldots, e_{k_v}$ appear. This will be done with the help of a locator matrix built as follows.

For $j = 0, \ldots, m-1$ and $i = 0, \ldots m-\theta-1$, set

$$l_{0,j} = \begin{cases} \rho_j & \text{if } j = 0, \ldots, \theta \\ 0 & \text{if } j = \theta+1, \ldots, m-1 \end{cases}, \qquad l_{i,-1} = 0. \quad (5)$$

We may then construct a matrix

$$L = \begin{pmatrix} l_{0,0} & l_{0,1} & \cdots & l_{0,m-1} \\ l_{1,0} & l_{1,1} & \cdots & l_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m-\theta-1,0} & l_{m-\theta-1,1} & \cdots & l_{m-\theta-1,m-1} \end{pmatrix} \quad (6)$$

by defining its entries recursively as

$$l_{i+1,j} = \sigma(l_{i,j-1}) + \delta(l_{i,j}). \quad (7)$$

For $i = 0, \ldots, m-1$ let ϵ_i denote the vector of K^m whose i–th component equal to 1, and every other component is 0. By $Row(LA)$ we denote the row space of the matrix LA.

Theorem 4. *The error positions $k_1, \ldots, k_v$ are, precisely, those*

$$k \in \{0, \ldots, m-1\}$$

such that $\epsilon_k \notin Row(LA)$.

The error values $e_{k_1}, \ldots, e_{k_v} \in K$ are the unique solution of the linear system

$$S_{i,0} = \sum_{j=1}^{v} e_{k_j} \varphi_u^{i+k_j}(\alpha), \qquad (0 \leq i \leq v-1).$$

We are now ready to specify our decoding algorithm.

Algorithm 1. Decoding algorithm for an RS skew-differential code $C_{(\varphi_u,\alpha,d)}$.

- The input is a received word $y = (y_0, \ldots, y_{m-1}) \in K^m$ with no more than $\tau = \lfloor \frac{d-1}{2} \rfloor$ errors.
- The output is an error vector $e = (e_0, \ldots, e_{m-1}) \in K^m$ such that $y - e \in C_{(\varphi_u,\alpha,d)}$.
- The algorithm runs according to the following steps:
  1. Compute $S_{i,0}$ according to (3) for $i = 0, \ldots, d-2$. If $S_{i,0} = 0$ for every $i = 0, \ldots, d-2$, then $e = 0$.
  2. Compute recursively S_r for $r \geq 2$ by means of (4) until $rank\, S_r < r$. Set $\theta = r-1$.
  3. Compute a nonzero $\rho = (\rho_0, \ldots, \rho_\theta)$ in the kernel of the matrix B formed by the first $\theta+1$ rows of S_θ.
  4. Compute the matrix L according to (5) and (7).
  5. The error positions set $T = \{k_1, \ldots, k_v\}$ is determined by
  $$T = \{k \in \{0, \ldots, m-1\} : \epsilon_k \notin Row(LA)\}.$$
  6. The error values $e_{k_1}, \ldots, e_{k_v}$ are the solutions of the linear system
  $$\sum_{j=0}^{m-1} y_j \varphi_u^{i+j}(\alpha) = \sum_{j=1}^{v} e_{k_j} \varphi_u^{i+k_j}(\alpha), \quad (0 \leq i \leq v-1).$$
  7. Set $e_i = 0$ for $i \notin T$. The error is $e = (e_0, \ldots, e_{m-1})$.

Remark 5. The location of the error positions from the matrix LA in the fifth step of Algorithm 1 can be done by several methods. For instance, one may compute the reduced row echelon form of LA, as we do in the examples exhibited in Section II.

II. Examples

Before giving concrete examples, we discuss under which conditions the dimension of K as a K^{φ_u}–vector space is finite. Of course, we also would like to avoid the extreme case where $K = K^{\varphi_u}$, so we also consider this situation. We first record separately an alternative description of K^{φ_u}, which will used several times. Keep the notation introduced in Section I.

Lemma 6. *The subfield K^{φ_u} of K admits the following description:*

$$K^{\varphi_u} = \{b \in K \,|\, \sigma(b)u + \delta(b) = ub\}.$$

Proof. It follows from the identities

$$\varphi_u(ab) = \sigma(a)\sigma(b)u + \sigma(a)\delta(b) + \delta(a)b$$

and

$$\varphi_u(a)b = \sigma(a)ub + \delta(a)b,$$

valid for any $a, b \in K$. □

Proposition 7. *1) The equality $K = K^{\varphi_u}$ holds if and only if $\delta(a) = u(a - \sigma(a))$ for every $a \in K$.*
2) If $K \neq K^{\varphi_u}$ then

$$K^{\varphi_u} = \begin{cases} K^\delta & \text{if } \sigma = id_K \\ K^\sigma & \text{if } \sigma \neq id_K. \end{cases}$$

Proof. Statement (1) follows immediately from Lemma 1.

As for statement (2) concerns, let us first observe that, since $\delta(ab) = \delta(ba)$ for every $a, b \in K$,

$$\delta(b)(\sigma(a) - a) = \delta(a)(\sigma(b) - b), \quad (8)$$

by virtue of (1). Now, since $K \neq K^{\varphi_u}$, we pick $a \in K \setminus K^{\varphi_u}$. If $b \in K^{\varphi_u}$, then

$$\delta(a)(\sigma(b)-b) = \delta(b)(\sigma(a)-a) = (b-\sigma(b))u(\sigma(a)-a),$$

which is only possible if $\sigma(b)-b=0$, as $\delta(a) \neq u(\sigma(a)-a)$. Therefore, $b \in K^\sigma$ and, henceforth, $\delta(b) = u(b-\sigma(b)) = 0$. We thus get that $K^{\varphi_u} \subseteq K^\sigma \cap K^\delta$. The converse inclusion is easily checked, so that we obtain

$$K^{\varphi_u} = K^\sigma \cap K^\delta. \tag{9}$$

If $\sigma = id_K$, then (9) obviously implies $K^{\varphi_u} = K^\delta$. If $\sigma \neq id_K$, we may already pick $a \in K \setminus K^\sigma$. Then, for $b \in K^\sigma$ we get form (8), that

$$\delta(b)(\sigma(a)-a) = \delta(a)(\sigma(b)-b) = 0.$$

Hence, $\delta(b) = 0$ and $K^\sigma \subseteq K^\delta$, which implies, in view of (9), that $K^{\varphi_u} = K^\sigma$. □

Given an automorphism $\sigma \neq id_K$ of the field K, and $v \in K$, we may define

$$\delta_{\sigma,v}(a) = v(\sigma(a)-a), \tag{10}$$

for all $a \in K$, thus obtaining a map $\delta_{\sigma,v} : K \to K$ which is a σ–derivation. Indeed, it is already known (see, for instance, [21, Proposition 1.1.20]), that every σ–derivation of the commutative field K can be expressed in this form (this fact is easily derived from (8)). With this notation, we derive the following consequence of Proposition 7.

Corollary 8. *Assume that $\sigma \neq id_K$ is an automorphism of K of finite order m, and that $\delta = \delta_{\sigma,v}$. If $u \neq -v$, then the dimension of K over K^{φ_u} is m.*

Remark 9. In practice, the computation of one of the cyclic vectors α predicted by Proposition 1 can be implemented by a randomized search in K until the matrix A becomes invertible. This avoids in most cases the computation of the subfield K^{φ_u}, since the parameter m is either the order of the automorphism σ or, in the pure differential case of interest (namely $K = \mathbb{F}(t)$), the characteristic of the finite field $\mathbb{F}$.

Next, we discuss how our construction applies to block and convolutional codes. We also illustrate the execution of our decoding algorithm with some concrete examples.

A. Block codes

Let us assume here that $K = \mathbb{F}$ is the finite field with p^r elements for some prime p, so our codes become linear block codes over the alphabet $\mathbb{F}$. Every automorphism of $\mathbb{F}$ is a power of the Frobenius automorphism τ and, consequently, has finite order. Additionally, any derivation on $\mathbb{F}$ is inner, this is to mean, it is given by (10), so Corollary 8 provides us plenty of non trivial examples. The steps of the design method of a RS skew-differential block code may be then enumerated as follows:

1) Choose a natural $0 < h < r$, and set $\sigma = \tau^h$ and $m = \frac{r}{(r,h)}$, the order of σ.
2) Choose v and u in $\mathbb{F}$, with $u+v \neq 0$, in order to set the σ-derivation $\delta : \mathbb{F} \to \mathbb{F}$ as $\delta(c) = v(\sigma(c)-c)$ and the additive map φ_u as $\varphi_u(c) = \sigma(c)u + \delta(c)$ for any $c \in \mathbb{F}$.
3) By a random search, find a cyclic vector α (see Remark 9).
4) Finally, choose a designed distance $2 \leq d \leq m$, and set the parity check matrix H as in Definition 2.

The degrees of freedom of this process suggest how wide this class of block codes is. Furthermore, RS skew-differential block codes are not cyclic, see Section IV. Nevertheless, Algorithm 1 provides a decoding method as efficient as the classical Peterson-Gorenstein-Zierler algorithm.

Let us now describe a concrete example. Consider $\mathbb{F} = \mathbb{F}_2(a)$ the field with $256 = 2^8$ elements, where $a^8 + a^4 + a^3 + a^2 + 1 = 0$. For brevity, except for 0 and 1, we write the elements of $\mathbb{F}$ as powers of a. Let σ be the Frobenius automorphism of $\mathbb{F}$, that is, $\sigma(c) = c^2$ for any $c \in \mathbb{F}$, which has order $m = 8$. Then, Corollary 8 says that our code is of length 8. We set $v = a$, yielding the σ-derivation given by $\delta(c) = ac^2 + ac$ for every $c \in \mathbb{F}$, and $u = a^2$, so $\varphi_u(c) = a^{26}c^2 + ac$ for every $c \in \mathbb{F}$.

We now choose $\alpha = a^9$. The matrix A from Proposition 1 takes now the form

$$A = \begin{pmatrix} a^9 & a^{146} & a^{103} & a^{244} & a^{214} & a^{89} & a & a^{200} \\ a^{146} & a^{103} & a^{244} & a^{214} & a^{89} & a & a^{200} & a^{237} \\ a^{103} & a^{244} & a^{214} & a^{89} & a & a^{200} & a^{237} & a^{95} \\ a^{244} & a^{214} & a^{89} & a & a^{200} & a^{237} & a^{95} & a^{105} \\ a^{214} & a^{89} & a & a^{200} & a^{237} & a^{95} & a^{105} & a^{175} \\ a^{89} & a & a^{200} & a^{237} & a^{95} & a^{105} & a^{175} & a^{184} \\ a & a^{200} & a^{237} & a^{95} & a^{105} & a^{175} & a^{184} & a^{21} \\ a^{200} & a^{237} & a^{95} & a^{105} & a^{175} & a^{184} & a^{21} & a^{159} \end{pmatrix}.$$

The determinant of A equals a^{47}, so that α is a cyclic vector. Finally, we set a designed distance $d = 5$. Let then $C = C_{(\varphi_u, a^9, 5)} \subseteq \mathbb{F}^8$ be the $[8, 4, 5]_{256}$-linear code defined as the left kernel of the following matrix H. From H, by standard methods, we have also computed a generating matrix G. Explicitly,

$$H = \begin{pmatrix} a^9 & a^{146} & a^{103} & a^{244} \\ a^{146} & a^{103} & a^{244} & a^{214} \\ a^{103} & a^{244} & a^{214} & a^{89} \\ a^{244} & a^{214} & a^{89} & a \\ a^{214} & a^{89} & a & a^{200} \\ a^{89} & a & a^{200} & a^{237} \\ a & a^{200} & a^{237} & a^{95} \\ a^{200} & a^{237} & a^{95} & a^{105} \end{pmatrix}$$

and

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & a^{105} & a^{69} & a^{221} & a^{41} \\ 0 & 1 & 0 & 0 & a^{109} & a^{25} & a^{232} & a^{166} \\ 0 & 0 & 1 & 0 & a^{145} & a^{54} & a^{104} & a^{36} \\ 0 & 0 & 0 & 1 & a^{251} & a^{141} & a^{42} & a^{60} \end{pmatrix}.$$

The reader may refer to Section IV and Remark 26 for the explicit calculation of a non-commutative generator polynomial of C so that the encoding can be performed in a similar way as for cyclic codes.

Let us exemplify the encoding-decoding process. We recall that the error-correcting capacity of C is $\tau = 2$. Suppose we want to transmit the message

$$M = \left(a^{61}, a^{102}, a^{182}, a^{250}\right),$$

so that we encode it to a codeword

$$c = MG = \left(a^{61}, a^{102}, a^{182}, a^{250}, a^{33}, a^{126}, a^{121}, a^{226}\right) \in C.$$

During the transmission, c is corrupted by adding the error vector

$$e = \left(0, a^2, 0, a^2, 0, 0, 0, 0\right),$$

yielding then the received word

$$y = c + e = \left(a^{61}, a^{6}, a^{182}, a^{107}, a^{33}, a^{126}, a^{121}, a^{226}\right).$$

Now, we run Algorithm 1. We first calculate the syndromes

$$yH = \left(a^{32}, a^{96}, a^{250}, a^{236}\right),$$

so it is detected some error. The syndrome matrix is then

$$S = \begin{pmatrix} a^{32} & a^{3} \\ a^{96} & a^{67} \\ a^{250} & a^{221} \end{pmatrix}.$$

The first column of S is a multiple of its second column, so that S has rank 1 and, henceforth, $\theta = 1$. Therefore, the matrix B in Algorithm 1 takes the form

$$B = \begin{pmatrix} a^{32} \\ a^{96} \end{pmatrix}.$$

and a basis of its left kernel is provided by the vector

$$\rho = \left(a, a^{192}\right).$$

The matrix L defined in (6) becomes

$$L = \begin{pmatrix} a & a^{192} & 0 & 0 & 0 & 0 & 0 & 0 \\ a^{27} & a^{125} & a^{129} & 0 & 0 & 0 & 0 & 0 \\ a^{132} & a^{44} & a^{148} & a^{3} & 0 & 0 & 0 & 0 \\ a^{193} & a^{105} & a^{215} & a^{102} & a^{6} & 0 & 0 & 0 \\ a^{222} & a^{134} & a^{212} & a^{108} & a^{134} & a^{12} & 0 & 0 \\ a^{205} & a^{117} & a^{209} & a^{216} & a^{212} & a^{25} & a^{24} & 0 \\ a^{158} & a^{70} & a^{195} & a^{206} & a^{88} & a^{245} & a^{222} & a^{48} \end{pmatrix},$$

and LA results

$$LA = \begin{pmatrix} a^{246} & a^{98} & a^{77} & a^{98} & a^{245} & a^{164} & a^{146} & a^{23} \\ a^{137} & a^{27} & a^{44} & a^{27} & a^{24} & a^{129} & a^{103} & a^{22} \\ a^{203} & a^{169} & a^{175} & a^{169} & a^{222} & a^{76} & a^{244} & a^{124} \\ a^{26} & a^{40} & a^{184} & a^{40} & a^{160} & a^{124} & a^{214} & a^{58} \\ a^{10} & a^{203} & a^{21} & a^{203} & a^{155} & a^{58} & a^{89} & a^{116} \\ a^{43} & a^{26} & a^{159} & a^{26} & a^{25} & a^{116} & a & a^{169} \\ a^{61} & a^{10} & a^{198} & a^{10} & a^{28} & a^{169} & a^{200} & a^{40} \end{pmatrix}.$$

The identification of the positions $k \in \{0, 1 \ldots, 7\}$ such that $\epsilon_k \notin \mathrm{Row}(LA)$ can be easily done if we compute the row reduced echelon form of LA,

$$LA_{rref} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

It is clear that ϵ_1 and ϵ_3 do not belong to $\mathrm{Row}(LA)$. Therefore, there are errors at positions 1 and 3. We finally need to solve a linear system in order to recover the error values. Indeed, the error values are the solution of the system

$$\begin{pmatrix} a^{146} & a^{103} \\ a^{244} & a^{214} \end{pmatrix} \begin{pmatrix} e_1 \\ e_3 \end{pmatrix} = \begin{pmatrix} a^{32} & a^{96} \end{pmatrix}.$$

The solution is, as expected, $e_1 = a^2$ and $e_3 = a^2$.

B. Convolutional codes

Another case of interest is $K = \mathbb{F}(t)$, the field of rational functions over a finite field $\mathbb{F}$. Linear codes over $\mathbb{F}(t)$ are examples convolutional codes, see [10] for details. It is well-known that the group $\mathrm{Aut}_{\mathbb{F}}\mathbb{F}(t)$ of all $\mathbb{F}$–linear automorphisms of the field $\mathbb{F}(t)$ can be identified with the projective general linear group $\mathrm{PGL}(2,\mathbb{F})$ via the map $\Phi : \mathrm{PGL}(2,\mathbb{F}) \to \mathrm{Aut}_{\mathbb{F}}\mathbb{F}(t)$, which maps any matrix $M = \left(\begin{smallmatrix} \sigma_1 & \sigma_2 \\ \sigma_3 & \sigma_4 \end{smallmatrix}\right) \in \mathrm{PGL}(2,\mathbb{F})$ to the automorphism $\Phi(M)$ determined by the rule $t \mapsto \frac{\sigma_1 t + \sigma_2}{\sigma_3 t + \sigma_4}$. Every automorphism of K has then finite order and, on the other hand, the field of constants of any derivation of $\mathbb{F}(t)$ has finite index. Thus, Proposition 7 says that virtually all choices of σ, δ and u lead to non trivial RS skew-differential convolutional codes to which the decoding algorithm 1 may be applied.

Remark 10. Algorithm 1 deals with the Hamming metric, which is not the usual distance considered in convolutional codes. However, the use of Hamming distances in the convolutional setting might be of interest in the technology of distributed storage (see [18, Sect. 2] and [1]).

Let us now detail a specific example. Let $\mathbb{F} = \mathbb{F}_2(a)$, where $a^2 + a + 1 = 0$, the field with four elements and set $K = \mathbb{F}(t)$ the field of rational functions with coefficients in $\mathbb{F}$. We shall follow likewise the construction method in Subection II-A.

As commented above, an automorphism of K is determined by four elements $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ in $\mathbb{F}$ verifying $\sigma_1\sigma_4 - \sigma_2\sigma_3 \neq 0$. Set $\sigma_1 = 0$, $\sigma_2 = 1$, $\sigma_3 = 1$ and $\sigma_4 = a$ yielding the automorphism $\sigma : K \to K$ determined by $\sigma(t) = 1/(t + a)$, which has order $m = 5$. For simplicity, we fix $v = 1$, so that $\delta(c) = \sigma(c) - c$ for any $c \in K$, and $u = 0$, and then $\varphi_u = \delta$. Now, consider $\alpha = t$. Since the matrix

$$A = \left(\begin{array}{ccc} t & \frac{t^2+at+1}{t+a} & \frac{t^2+at+1}{t+1} \\ \frac{t^2+at+1}{t+a} & \frac{t^2+at+1}{t+1} & \frac{t^4+at^3+t^2}{t^3+1} \\ \frac{t^2+at+1}{t+1} & \frac{t^4+at^3+t^2}{t^3+1} & \frac{t^2+at+1}{t} \\ \frac{t^4+at^3+t^2}{t^3+1} & \frac{t^2+at+1}{t} & \frac{t^2+at+1}{a^2t^2+t} \\ \frac{t^2+at+1}{t} & \frac{t^2+at+1}{a^2t^2+t} & \frac{a^2t^4+t^3+at^2+at+1}{a^2t^3+at^2+t} \end{array}\right.$$

$$\left.\begin{array}{cc} \frac{t^4+at^3+t^2}{t^3+1} & \frac{t^2+at+1}{t} \\ \frac{t^2+at+1}{t} & \frac{t^2+at+1}{a^2t^2+t} \\ \frac{t^2+at+1}{a^2t^2+t} & \frac{a^2t^4+t^3+at^2+at+1}{a^2t^3+at^2+t} \\ \frac{a^2t^4+t^3+at^2+at+1}{a^2t^3+at^2+t} & \frac{t^2+at+1}{at^2+t} \\ \frac{t^2+at+1}{at^2+t} & \frac{t^2+at+1}{t+a+1} \end{array}\right)$$

is non-singular, we get from Proposition 1 that α is a cyclic vector for δ. Finally, the designed distance is selected to be $d =$

3. So the skew-differential convolutional code $C = C_{(\delta,t,3)}$ can correct a single error, and a parity check matrix takes the form

$$H = \begin{pmatrix} t & \frac{t^2+at+1}{t+a} \\ \frac{t^2+at+1}{t+a} & \frac{t^2+at+1}{t+1} \\ \frac{t^2+at+1}{t+1} & \frac{t^4+a^2t^3+t^2}{t^3+1} \\ \frac{t^4+a^2t^3+t^2}{t^3+1} & \frac{t^2+at+1}{t} \\ \frac{t^2+at+1}{t} & \frac{t^2+at+1}{a^2t^2+t} \end{pmatrix}$$

Let us briefly exemplify our decoding algorithm. Suppose that we receive the word

$$y = \left(0,\, 1,\, a^2,\, \frac{t^2+t}{a^2t^2+t+1},\, 0\right),$$

whose matrix of syndromes is as follows:

$$S = \begin{pmatrix} \frac{t^3+at^2+t}{t^4+at^2+at+1} \\ \frac{t^3+at^2+t}{a^2t^5+t^4+t^3+a^2t^2+t+1} \end{pmatrix}.$$

Henceforth, the system detects errors during the transmission. Clearly $\theta = 1$ and $B = S$, and the vector ρ becomes $\rho = (1, at+1)$. The matrix L takes the form

$$L = \begin{pmatrix} 1 & a^2t+1 & 0 & 0 & 0 \\ 0 & \frac{a^2t^2+1}{t+a} & \frac{t+1}{t+a} & 0 & 0 \\ 0 & \frac{t^2+at+1}{at+a} & 1 & \frac{t+a+1}{at+a} & 0 \\ 0 & \frac{t^4+at^3+t^2}{at^3+a} & \frac{t^2+at+1}{at^2+at+a} & \frac{1}{a^2t^2+t+a} & \frac{a^2t}{at+1} \end{pmatrix}.$$

We then compute LA and its row reduced echelon form obtaining that

$$LA_{\mathrm{rref}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

It is clear that ϵ_4 does not belong to $\mathrm{Row}LA$, so we find an error at position 4. When computing the error value we find that

$$e_4 = \frac{t^2}{t^4+at^2+at+1}.$$

Therefore, the correction gives the codeword

$$c = \left(0,\, 1,\, a^2,\, \frac{t^2+t}{a^2t^2+t+1},\, \frac{t^2}{t^4+at^2+at+1}\right),$$

and the original message would be $M = \big(0,\, 1,\, a^2\big)$.

III. Mathematical set up and proofs

The aim of this section is to prove the mathematical results that ground Algorithm 1. So, let (σ, δ) be a skew-derivation on a field K, as defined in Section I. Recall that, for each $u \in K$, we define

$$\varphi_u(a) = \sigma(a)u + \delta(a), \tag{11}$$

for all $a \in K$, thus obtaining a map $\varphi_u : K \to K$. This additive map becomes right K^{φ_u}–linear, where

$$K^{\varphi_u} = \{b \in K : \varphi_u(ab) = \varphi_u(a)b \text{ for all } a \in K\}$$

is the φ_u–invariant subfield of K.

Let $\mathrm{End}(K)$ denote the ring of endomorphisms of K as an additive group. Let $\mathcal{R}$ be the subring of $\mathrm{End}(K)$ generated by K and φ_u. Here, K is seen as a subring of $\mathrm{End}(K)$ by considering each element a of K as the additive endomorphism given by multiplication by a.

Proposition 11. *If the dimension of K as a K^{φ_u}–vector space is $m < \infty$, then the minimal polynomial of φ_u as a K^{φ_u}–linear map has degree m. Consequently, φ_u has at least a cyclic vector $\alpha \in K$. Moreover,*

$$\mathcal{R} = K \oplus K\varphi_u \oplus \cdots \oplus K\varphi_u^{m-1}. \tag{12}$$

Proof. It easily follows from (1) that, in $\mathrm{End}(K)$,

$$\varphi_u a = \sigma(a)\varphi_u + \delta(a), \tag{13}$$

for all $a \in K$. This implies that $\mathcal{R} = K + K\varphi_u + K\varphi_u^2 + \cdots$.

Now, since $\dim_{K^{\varphi_u}} K = m$, the minimal polynomial of φ_u as a K^{φ_u}–linear map has degree $n \le m$. This in particular implies that $\mathcal{R} = K + K\varphi_u + \cdots + K\varphi_u^{n-1}$. On the other hand, by Jacobson-Bourbaki's correspondence [33, Theorem 4.1], $m = \dim_K \mathcal{R}$. We thus derive that $n = m$ and (12). □

In the rest of the paper, we assume that $\dim_{K^{\varphi_u}} K = m < \infty$. According to Proposition 11, the minimal equation of φ_u over K^{φ_u} has degree m, that is, is of the form

$$0 = \varphi_u^m + \mu_{m-1}\varphi_u^{m-1} + \cdots + \mu_1\varphi_u + \mu_0 \tag{14}$$

with $\mu_i \in K^{\varphi_u}$ for $i = 0, \ldots, m-1$.

Let $\alpha \in K$. For any subset $\{t_1, \ldots, t_n\} \subseteq \{0, \ldots, m-1\}$, define, following [9], the matrix

$$W(\varphi_u^{t_1}(\alpha), \ldots, \varphi_u^{t_n}(\alpha)) = \begin{pmatrix} \varphi_u^{t_1}(\alpha) & \varphi_u^{t_2}(\alpha) & \cdots & \varphi_u^{t_n}(\alpha) \\ \varphi_u^{t_1+1}(\alpha) & \varphi_u^{t_2+1}(\alpha) & \cdots & \varphi_u^{t_n+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{t_1+n-1}(\alpha) & \varphi_u^{t_2+n-1}(\alpha) & \cdots & \varphi_u^{t_n+n-1}(\alpha) \end{pmatrix}.$$

Lemma 12. *Given $\alpha \in K$, the following conditions are equivalent.*

1) *α is a cyclic vector for the K^{φ_u}–linear map φ_u.*
2) *$W(\alpha, \varphi_u(\alpha), \ldots, \varphi_u^{m-1}(\alpha))$ is an invertible matrix.*
3) *$W(\varphi_u^{t_1}(\alpha), \ldots, \varphi_u^{t_n}(\alpha))$ is an invertible matrix for every subset $\{t_1, \ldots, t_n\} \subseteq \{0, \ldots, m-1\}$.*

Proof. For every nonzero $c \in K$, consider the conjugate of u by c:

$${}^c u = \sigma(c)uc^{-1} + \delta(c)c^{-1}.$$

By Lemma 6,

$$K^{\varphi_u} = \{c \in K \setminus \{0\} \mid {}^c u = u\} \cup \{0\};$$

the latter being the $(\sigma - \delta)$–centralizer of u in the terminology of [9]. Since α is a cyclic vector for φ_u precisely when $\{\alpha, \varphi_u(\alpha), \ldots, \varphi_u^{m-1}(\alpha)\}$ is a K^{φ_u}–basis of K, we may apply [9, Theorem 5.3] to deduce that the three conditions are equivalent. □

Proof of Proposition 1. It is a consequence of Proposition 11 and Lemma 12.

Fix a cyclic vector $\alpha \in K$ of φ_u. Let $A = W(\alpha, \varphi_u(\alpha), \ldots, \varphi_u^{m-1}(\alpha))$ which, by Lemma 12, is an invertible matrix with coefficients in K.

Theorem 13. *For $2 \leq d \leq m$, let $C_{(\varphi_u,\alpha,d)} \subseteq K^m$ be the left kernel of the matrix*

$$H = \begin{pmatrix} \alpha & \varphi_u(\alpha) & \cdots & \varphi_u^{d-2}(\alpha) \\ \varphi_u(\alpha) & \varphi_u^2(\alpha) & \cdots & \varphi_u^{d-1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{m-1}(\alpha) & \varphi_u^m(\alpha) & \cdots & \varphi_u^{m+d-3}(\alpha) \end{pmatrix}. \quad (15)$$

Then $C_{(\varphi_u,\alpha,d)}$ is a K–linear code of dimension $m-d+1$ and minimum Hamming distance d.

Proof. Since H consists of the first $d-1$ columns of the invertible matrix A, we get that the dimension of the left K–vector subspace $C_{(\varphi_u,\alpha,d)}$ is $m-d+1$. Every submatrix M of order $d-1$ of H is of the form

$$M = \begin{pmatrix} \varphi_u^{k_1}(\alpha) & \varphi_u^{k_1+1}(\alpha) & \cdots & \varphi_u^{k_1+d-2}(\alpha) \\ \varphi_u^{k_2}(\alpha) & \varphi_u^{k_2+1}(\alpha) & \cdots & \varphi_u^{k_2+d-2}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{k_{d-1}}(\alpha) & \varphi_u^{k_{d-1}+1}(\alpha) & \cdots & \varphi_u^{k_{d-1}+d-2}(\alpha) \end{pmatrix},$$

where $\{k_1, \ldots, k_{d-1}\} \subseteq \{0, \ldots, m-1\}$. We see that

$$M = W(\varphi_u^{k_1}(\alpha), \ldots, \varphi_u^{k_{d-1}}(\alpha))^t,$$

which is, by Lemma 12, invertible. Hence, the Hamming distance of $C_{(\varphi_u,\alpha,d)}$ is d. □

The proof of Lemma 12 is based on a result from [9] in the realm of the theory of skew polynomials. Indeed, for some purposes, it is useful to understand the ring $\mathcal{R}$ as a factor ring of a ring of skew polynomials. Let us derive such a description.

The skew derivation (σ, δ) leads to the construction of a non commutative polynomial ring $R = K[x; \sigma, \delta]$, often called a skew polynomial ring (see, e.g., [21]). The elements of R are polynomials in an indeterminate x with coefficients from K written on the left (that is, the monomials $1, x, x^2, \ldots$ form a basis of R as a left vector space over K). The multiplication of R is subject to the following rule:

$$xa = \sigma(a)x + \delta(a), \quad (16)$$

for all $a \in K$.

Proposition 14. *The map $\pi : R \to \mathcal{R}$ that sends $\sum_i f_i x^i$ onto $\sum_i f_i \varphi_u^i$ is a surjective ring homomorphism whose kernel is $R\mu = \mu R$, where*

$$\mu = x^m + \sum_{i=0}^{m-1} \mu_i x^i$$

is a polynomial in R built from the coefficients of the minimal equation of φ_u, see (14)*.*

Hence, there is a K–linear isomorphism of rings $R/R\mu \cong \mathcal{R}$.

Proof. Observe that π is clearly left K–linear and, from Proposition 11, surjective. It is multiplicative since φ_u satisfies (13). Its kernel is an ideal I of R which, as a left ideal, is generated by the monic polynomial $h \in R$ in I of least degree, due to the left Euclidean division algorithm enjoyed by R (see, e.g. [21]). Also, the degree of h is the dimension of $R/I \cong \mathcal{R}$ as a left K–vector space. By Proposition 11, this dimension equals m. We see that μ fits these requirements, so that $h = \mu$, and $I = R\mu$. Finally, since I is an ideal, we get that $I = \mu R$ as well. □

We may thus identify $\mathcal{R}$ with $R/R\mu$, and, therefore, its elements with polynomials in R with degree smaller than m (this identification makes correspond φ_u with x). This view makes some concepts more natural, like the degree of an element of $\mathcal{R}$.

The coordinate isomorphism of left K–vector spaces

$$\mathfrak{v} : \mathcal{R} \to K^m, \qquad (\sum_{i=0}^{m-1} f_i x^i \mapsto (f_0, f_1, \ldots, f_{m-1}))$$

allows the transfer of elements and vector subspaces between both K–vector spaces.

We are ready to consider our decoding algorithm. Let $c \in C_{(\varphi_u,\alpha,d)}$ be a codeword that is transmitted through a noisy channel, and let

$$y = (y_0, y_1, \ldots, y_{m-1}) \in K^m$$

be the received word. We may decompose $y = c + e$, where

$$e = (e_0, e_1, \ldots, e_{m-1}) \in K^m$$

is the error vector. By $k_1, \ldots, k_v \in \{0, 1, \ldots, m-1\}$ we denote the positions where the nonzero error values $e_{k_1}, \ldots, e_{k_v} \in K$ occur. We prove first that the latter can be computed from y once the positions are known.

Proposition 15. *If $0 \leq i \leq d-2$, then*

$$\sum_{j=0}^{m-1} y_j \varphi_u^{i+j}(\alpha) = \sum_{j=1}^{v} e_{k_j} \varphi_u^{i+k_j}(\alpha). \quad (17)$$

Therefore, if $v \leq d-1$, then $(e_{k_1}, \ldots, e_{k_v})$ is the unique solution of the linear system of equations

$$\sum_{j=0}^{m-1} y_j \varphi_u^{i+j}(\alpha) = \sum_{j=1}^{v} e_{k_j} \varphi_u^{i+k_j}(\alpha), \qquad (0 \leq i \leq v-1). \quad (18)$$

Proof. The equations (17) hold because $C_{(\varphi_u,\alpha,d)}$ is the left kernel of the matrix H defined in (15). The linear system (18) has a unique solution since the matrix

$$\begin{pmatrix} \varphi_u^{k_1}(\alpha) & \varphi_u^{k_1+1}(\alpha) & \cdots & \varphi_u^{k_1+v-1}(\alpha) \\ \varphi_u^{k_2}(\alpha) & \varphi_u^{k_2+1}(\alpha) & \cdots & \varphi_u^{k_2+v-1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{k_v}(\alpha) & \varphi_u^{k_v+1}(\alpha) & \cdots & \varphi_u^{k_v+v-1}(\alpha) \end{pmatrix} = W(\varphi_u^{k_1}(\alpha), \ldots, \varphi_u^{k_v}(\alpha))^t$$

is invertible by Lemma 12.

□

Our aim is then to design an algorithm for computing the positions $k_1, \ldots, k_v$ where the errors $e_{k_1}, \ldots, e_{k_v}$ appear. We assume in our exposition that $e \neq 0$.

For every pair (i,k) of non-negative integers, set

$$S_{i,k} = \sum_{j=1}^{v} \varphi_u^{i+k_j}(\alpha)\psi^k(e_{k_j}), \tag{19}$$

where

$$\psi(a) = \sigma^{-1}(\delta(a) - ua) \tag{20}$$

for all $a \in K$.

Lemma 16. *For all pairs (i,k) of non-negative integers, we have*

$$\sigma(S_{i,k+1}) = \delta(S_{i,k}) - S_{i+1,k} \tag{21}$$

Moreover,

$$S_{i,0} = \sum_{j=0}^{m-1} y_j \varphi_u^{i+j}(\alpha), \tag{22}$$

for every $i = 0, \ldots, d-2$, and the values $S_{i,k}$ can be computed recursively by means of (21) *from the received word y whenever $i + k \leq d-2$.*

Proof. Observe that

$$\sigma(a\psi(b)) = \delta(ab) - \varphi_u(a)b, \tag{23}$$

for all $a, b \in K$. Indeed,

$$\begin{array}{rcl} \sigma(a\psi(b)) & \overset{(20)}{=} & \sigma(a)(\delta(b) - ub) \\ & \overset{(1)}{=} & \delta(ab) - \delta(a)b - \sigma(a)ub \\ & \overset{(11)}{=} & \delta(ab) - \varphi_u(a)b. \end{array}$$

For every pair (i,k),

$$\begin{array}{rcl} \sigma(S_{i,k+1}) & \overset{(19)}{=} & \sum_{j=1}^{v} \sigma(\varphi_u^{i+k_j}(\alpha)\psi^{k+1}(e_{k_j})) \\ & \overset{(23)}{=} & \sum_{j=1}^{v} \delta(\varphi_u^{i+k_j}(\alpha)\psi^k(e_{k_j})) \\ & & -\sum_{j=1}^{v} \varphi_u^{i+k_j+1}(\alpha)\psi^k(e_{k_j}) \\ & \overset{(19)}{=} & \delta(S_{i,k}) - S_{i+1,k}. \end{array}$$

Finally, since K is commutative, (22) follows from (17). □

Set $T = \{k_1, \ldots, k_v\}$, and let A_T be the submatrix of $A = W(\alpha, \varphi_u(\alpha), \ldots, \varphi_u^{m-1}(\alpha))$ formed by the columns at positions $k_1, \ldots, k_v$, that is

$$A_T = \begin{pmatrix} \varphi_u^{k_1}(\alpha) & \varphi_u^{k_2}(\alpha) & \cdots & \varphi_u^{k_v}(\alpha) \\ \varphi_u^{k_1+1}(\alpha) & \varphi_u^{k_2+1}(\alpha) & \cdots & \varphi_u^{k_v+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{k_1+m-1}(\alpha) & \varphi_u^{k_2+m-1}(\alpha) & \cdots & \varphi_u^{k_v+m-1}(\alpha) \end{pmatrix}.$$

Proposition 17. *Define, for every $1 \leq r$, the matrix*

$$E_r = \begin{pmatrix} e_{k_1} & \psi(e_{k_1}) & \cdots & \psi^{r-1}(e_{k_1}) \\ e_{k_2} & \psi(e_{k_2}) & \cdots & \psi^{r-1}(e_{k_2}) \\ \vdots & \vdots & \ddots & \vdots \\ e_{k_v} & \psi(e_{k_v}) & \cdots & \psi^{r-1}(e_{k_v}) \end{pmatrix}.$$

and set

$$\theta = \max\{r : rank\, E_r = r\}.$$

1) *If $V \subseteq K^m$ is the left kernel of the matrix $A_T E_\theta$, then $\mathfrak{v}^{-1}(V) = \mathcal{R}\rho$ for some $\rho \in \mathcal{R}$ of degree θ.*
2) *If B is the matrix formed by the first $\theta+1$ rows of $A_T E_\theta$, then we may choose $\rho = \rho_0 + \rho_1 x + \cdots + \rho_\theta x^\theta$, for any nonzero vector $(\rho_0, \rho_1, \ldots, \rho_\theta)$ in the left kernel of B.*

Proof. (1) We will prove that the K–vector subspace $I = \mathfrak{v}^{-1}(V)$ of $\mathcal{R}$ is a left ideal. To do this, we need just to check that $xI \subseteq I$. Given $\sum_{i=0}^{m-1} a_i x^i \in \mathcal{R}$ we get from (16), since $\mu = 0$ in $\mathcal{R}$, that

$$x(\sum_{i=0}^{m-1} a_i x^i) = \sum_{i=0}^{m-1} (\sigma(a_{i-1}) + \delta(a_i) - \sigma(a_{m-1})\mu_i)x^i, \tag{24}$$

where we set $a_{-1} = 0$.

Suppose that $(a_0, \ldots, a_{m-2}, a_{m-1})A_T E_\theta = 0$. The maximality of θ ensures that the last column of $E_{\theta+1}$ is a right linear combination of the former θ columns. Hence,

$$(a_0, \ldots, a_{m-2}, a_{m-1})A_T E_{\theta+1} = 0.$$

Observe that

$$A_T E_{\theta+1} = \begin{pmatrix} S_{0,0} & S_{0,1} & \cdots & S_{0,\theta} \\ S_{1,0} & S_{1,1} & \cdots & S_{1,\theta} \\ \vdots & \vdots & \ddots & \vdots \\ S_{m-1,0} & S_{m-1,1} & \cdots & S_{m-1,\theta} \end{pmatrix}.$$

Therefore,

$$\sum_{i=0}^{m-1} a_i S_{i,k} = 0, \qquad \text{for all } 0 \leq k \leq \theta. \tag{25}$$

For $0 \leq k \leq \theta - 1$ we have

$$\begin{array}{rcl} \sum_{i=0}^{m-1}(\sigma(a_{i-1}) + \delta(a_i))S_{i,k} & \overset{(1)}{=} & \sum_{i=0}^{m-1}\{\sigma(a_{i-1})S_{i,k} \\ & & +\delta(a_i S_{i,k}) - \sigma(a_i)\delta(S_{i,k})\} \\ & \overset{(25)}{=} & \sum_{i=0}^{m-1} \sigma(a_{i-1})S_{i,k} \\ & & -\sum_{i=0}^{m-1} \sigma(a_i)\delta(S_{i,k}) \\ & \overset{(21)}{=} & \sum_{i=0}^{m-1} \sigma(a_{i-1})S_{i,k} \\ & & -\sum_{i=0}^{m-1} \sigma(a_i)\sigma(S_{i,k+1}) \\ & & -\sum_{i=0}^{m-1} \sigma(a_i)S_{i+1,k} \\ & = & \sum_{i=0}^{m-1} \sigma(a_{i-1})S_{i,k} \\ & & -\sigma(\sum_{i=0}^{m-1} a_i S_{i,k+1}) \\ & & -\sum_{i=0}^{m-1} \sigma(a_i)S_{i+1,k} \\ & \overset{(25)}{=} & \sum_{i=0}^{m-1} \sigma(a_{i-1})S_{i,k} \\ & & -\sum_{i=0}^{m-1} \sigma(a_i)S_{i+1,k} \\ & = & -\sigma(a_{m-1})S_{m,k}. \end{array}$$

Since, by (14), $\varphi_u^m + \sum_{i=0}^{m-1} \mu_i \varphi_u^i = 0$, we get

$$\begin{array}{rcl} S_{m,k} & = & \sum_{j=1}^{v} \varphi_u^{m+k_j}(\alpha)\psi^k(e_{k_j}) \\ & = & \sum_{j=1}^{v}[-\sum_{i=0}^{m-1} \mu_i \varphi_u^{k_j+i}(\alpha)]\psi^k(e_{k_j}) \\ & = & -\sum_{i=0}^{m-1} \mu_i \sum_{j=1}^{v} \varphi_u^{k_j+i}(\alpha)\psi^k(e_{k_j}) \\ & = & -\sum_{i=0}^{m-1} \mu_i S_{i,k}. \end{array}$$

Then $\sum_{i=0}^{m-1}(\sigma(a_{i-1})+\delta(a_i))S_{i,k} = \sum_{i=0}^{m-1}\sigma(a_{m-1})\mu_i S_{i,k}$ and, therefore,

$$(b_0, b_1, \ldots, b_{m-1})A_T E_\theta = 0,$$

where $b_i = \sigma(a_{i-1})+\delta(a_i)-\sigma(a_{m-1})\mu_i$ for $i = 0, \ldots, m-1$.

We thus deduce from (24) that $x(\sum_{i=0}^{m-1} a_i x^i) \in I$ whenever $\sum_{i=0}^{m-1} a_i x^i \in I$. Hence, I is a left ideal of $\mathcal{R}$ and $I = \mathcal{R}\rho$ for some nonzero polynomial ρ. As for its degree concerns, we have

$$\deg \rho = \dim_K \frac{\mathcal{R}}{\mathcal{R}\rho} = \dim_K \frac{K^m}{V} = \theta,$$

since $A_T E_\theta$ is full rank.

(2) We know from (1) that, if $\rho = \rho_0 + \cdots + \rho_\theta x^\theta$, then the vector $(\rho_0, \ldots, \rho_\theta, 0, \ldots, 0) \in K^m$ belongs to the left kernel of $A_T E_\theta$. Now, the statement should be clear. □

Next, we will state the result that will allow the location of the error positions. We need to construct a matrix from the polynomial ρ given in Proposition 17.

For $j = 0, \ldots, m-1$ and $i = 0, \ldots m-\theta-1$, set

$$l_{0,j} = \begin{cases} \rho_j & \text{if } j = 0, \ldots, \theta \\ 0 & \text{if } j = \theta+1, \ldots, m-1 \end{cases}, \qquad l_{i,-1} = 0.$$

We may then construct a matrix

$$L = \begin{pmatrix} l_{0,0} & l_{0,1} & \cdots & l_{0,m-1} \\ l_{1,0} & l_{1,1} & \cdots & l_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ l_{m-\theta-1,0} & l_{m-\theta-1,1} & \cdots & l_{m-\theta-1,m-1} \end{pmatrix} \quad (26)$$

by defining its entries recursively as

$$l_{i+1,j} = \sigma(l_{i,j-1}) + \delta(l_{i,j}).$$

For $i = 0, \ldots, m-1$, let ϵ_i denote the vector of K^m whose i–th component is equal to 1, and every other component is 0. By $Row(LA)$ we denote the row space of the matrix LA.

Theorem 18. *If $T = \{k_1, \ldots, k_v\}$ is the set of error positions, then*

$$T = \{k \in \{0, \ldots, m-1\} : \epsilon_k \notin Row(LA)\}.$$

Proof. Let us first prove that the rows of L form a K–basis of $V = \ker(\cdot A_T E_\theta)$, the left kernel of $A_T E_\theta$. According to Proposition 17, $\mathfrak{v}(\mathcal{R}\rho) = V$, where $\rho = \sum_{i=0}^{\theta} \rho_i x^i$.

Observe that $\rho, x\rho, \ldots, x^{m-1-\theta}\rho$ have different degrees $\theta, \ldots, m-1$, so they are K–linearly independent in $\mathcal{R}$. Since the dimension of $\mathcal{R}\rho$ is $m-\theta$, we get that they form a basis and, hence, the rows of

$$M_\rho = \begin{pmatrix} \mathfrak{v}(\rho) \\ \mathfrak{v}(x\rho) \\ \vdots \\ \mathfrak{v}(x^{m-1-\theta}\rho) \end{pmatrix}$$

give a basis of $\mathfrak{v}(\mathcal{R}\rho)$. Note that the first row of M_ρ is $\mathfrak{v}(\rho)$. Indeed, a straightforward computation based on (1) leads to admit that the j-th row of L is, precisely, $\mathfrak{v}(x^j\rho)$, for $j = 0, \ldots, m-1-\theta$. Thus, $L = M_\rho$.

Let I be denote the identity matrix of size $m \times m$, and denote by I_T the submatrix of I formed by the columns at positions $k_1, \ldots, k_v$. Note that $A_T = AI_T$. This implies that $Row(LA) = \ker(\cdot I_T E_\theta)$. Indeed, we have proved that $Row(L) = \ker(\cdot A_T E_\theta)$, so that

$$\begin{aligned} Row(LA) &= \{x \,|\, xA^{-1} \in Row(L)\} \\ &= \{x \,|\, xA^{-1} \in \ker(\cdot A_T E_\theta)\} \\ &= \ker(\cdot I_T E_\theta). \end{aligned}$$

Let $i \in \{0, \ldots, m-1\}$. If $i \in T$, then $\epsilon_i I_T E_\theta$ is the i-th row of E_θ, while if $i \notin T$, then $\epsilon_i I_T E_\theta = 0$. Since every row of E_θ is non zero, we get that $\epsilon_i \in Row(LA)$ if and only if $i \notin T$. □

In our decoding algorithm, we need to compute θ from the received word y. To this end, set

$$\tau = \lfloor \frac{d-1}{2} \rfloor,$$

the integer part of $(d-1)/2$.

Lemma 19. *For every $r \geq 1$, define the matrix*

$$S_r = \begin{pmatrix} S_{0,0} & S_{0,1} & \cdots & S_{0,r-1} \\ S_{1,0} & S_{1,1} & \cdots & S_{1,r-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{\tau,0} & S_{\tau,1} & \cdots & S_{\tau,r-1} \end{pmatrix}.$$

If $v \leq \tau$, then $\theta = \max\{r : rank\, S_r = r\}$.

Proof. Observe that $S_r = ME_r$, where

$$M = \begin{pmatrix} \varphi_u^{k_1}(\alpha) & \varphi_u^{k_2}(\alpha) & \cdots & \varphi_u^{k_v}(\alpha) \\ \varphi_u^{k_1+1}(\alpha) & \varphi_u^{k_2+1}(\alpha) & \cdots & \varphi_u^{k_v+1}(\alpha) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{k_1+\tau}(\alpha) & \varphi_u^{k_2+\tau}(\alpha) & \cdots & \varphi_u^{k_v+\tau}(\alpha) \end{pmatrix}.$$

Since $v \leq \tau$, the rank of M is v due to Lemma 12. We thus get that $rk\, S_r = rk\, E_r$ for all $r \geq 1$, which gives the desired determination of θ. □

Next, we derive the proofs of Proposition 3 and Theorem 4.

Proof of Proposition 3. Lemma 19 gives that, whenever $v \leq \tau$,

$$\max\{r : rank\, S_r = r\} = \theta = \max\{r : rank\, E_r = r\}.$$

By Lemma 16, the matrix B consists of the first $\theta+1$ rows of $A_T E_\theta$, as in Proposition 17.(2). Now, B has rank θ, so that its left kernel is of dimension 1 as a K–vector space. Proposition 2 guarantees that $(\rho_0, \rho_1, \ldots, \rho_\theta)$ belongs to this kernel and $\rho_\theta \neq 0$.

Proof of Theorem 4. By Lemma 19, since we assume that $v \leq \tau$, we get that

$$\max\{r : rank\, S_r = r\} = \theta = \max\{r : rank\, E_r = r\}.$$

Thus, Theorem 18 is of application to obtain the first part of Theorem 4. The second statement is given by Proposition 15.

We finally state and prove the main result in this paper.

Theorem 20. *Assume K to be a commutative field and $v \leq \tau = \lfloor (d-1)/2 \rfloor$. Then Algorithm 1 correctly computes the error vector.*

Proof. The output of Line 1 is $e = 0$ since $C_{(\varphi_u,\alpha,d)}$ is the kernel of the matrix H in Definition 2. Line 2 runs whenever $e \neq 0$. In such a case, since we are assuming the the number of errors is $v \leq \tau$, it follows from Proposition 15 that $S_{i,0} \neq 0$ for at least one $0 \leq i \leq \tau$, as the linear system (18) has a unique solution. This is to mean that $S_1 \neq 0$ and, henceforth, always under the condition $v \leq \tau$, the number θ computed in Line 2 equals $\max\{r : rank\, S_r = r\}$. Proposition 3 guarantees the existence of a nonzero vector ρ to be computed in Line 3, which serves as the initial datum to the calculation of the matrix L in Line 4. Finally, Theorem 4 assures that the error positions and values computed in Lines 5 and 6 lead to a correct output in Line 7. □

IV. Skew-differential codes as (σ,δ)–codes.

In Section III, the ring $\mathcal{R}$ was proved (Proposition 14) to be isomorphic to a factor ring of the skew-polynomial ring $R = K[x;\sigma,\delta]$. Indeed, as we will see later, the codes $C_{(\varphi_u,\alpha,d)}$ are left ideals of $\mathcal{R}$ and, henceforth, they constitute a class of (σ,δ)–codes in the sense of [6], which enjoys an efficient algebraic decoding algorithm (see Algorithm 1). In this section, our aim is to describe precisely how the codes $C_{(\varphi_u,\alpha,d)}$ look like from the perspective of the ring R, although this view, we think, is less practical, for our purposes, than our choice in the previous sections, which are independent from the forthcoming material.

Given $u \in K$ we may consider the principal left ideal $R(x-u)$ of R generated by $x-u \in R$. Since K is a subring of R in the obvious way, the factor left R–module $R/R(x-u)$ is a left K–vector space of dimension 1. An explicit isomorphism

$$R/R(x-u) \cong K \tag{27}$$

sends the equivalence class of $g(x) \in R$ onto its *right evaluation* $g[u] \in K$, defined as the remainder of the left Euclidean division

$$g(x) = q(x)(x-u) + g[u], \tag{28}$$

where $q(x) \in R$ is a suitable polynomial.

The left R–module structure of $R/R(x-u)$ is transferred to K via the isomorphism (27), and it leads to a ring homomorphism

$$\lambda : R \longrightarrow \mathrm{End}(K), \tag{29}$$

where $\mathrm{End}(K)$ is still denote the ring of all additive endomorphisms of K. Recall that λ sends $f \in R$ onto the map defined by left multiplication by f according to the left R–module structure of K. A straightforward computation shows that λ sends $f = \sum_i f_i x^i$ onto $\sum_i f_i \varphi_u^i$, so that it acts as the map π from Proposition 14. Henceforth, the kernel of λ equals $R\mu = \mu R$, and λ induces the isomorphism $R/R\mu \cong \mathcal{R}$ from Proposition 14. We are assuming, as in Section III, that $\mu \neq 0$ and its degree is m. Recall that a K–basis of $\mathcal{R}$ is $\{1, x, \ldots, x^{m-1}\}$ where we are identifying each element of $\mathcal{R}$ with its unique representative in R of degree smaller than m. This natural basis of $\mathcal{R}$ leads to the corresponding coordinate isomorphism

$$\mathfrak{v} : \mathcal{R} \to K^m.$$

Definition 21. A K–linear code $C \subseteq K^m$ is said to be a (σ,δ,u)–*code* if $\mathfrak{v}^{-1}(C)$ is a left ideal of $\mathcal{R}$. These codes will be referred to as *skew-differential codes*.

Remark 22. In [6], a *module* (σ,δ)–*code* is defined as submodule of a left module of the form R/Rf, for some nonzero skew polynomial $f \in R$. Indeed, their definition is given for K a finite field but, obviously, it makes sense for a general field. From this perspective, the (σ,δ,u)–codes are instances of module (σ,δ)–codes, when one sets $f = \mu$, the minimal polynomial of φ_u over K^{φ_u} (and, hence, $R/Rf = \mathcal{R}$).

Every (σ,δ,u)–code admits a nice presentation in terms of linear skew polynomials. Recall that, for any $c \in K^*$, we have the conjugate ${}^c u = \sigma(c)uc^{-1} + \delta(c)c^{-1}$.

Proposition 23. *Every (σ,δ,u)–code is of the form*

$$C = \mathfrak{v}(\mathcal{R}g),$$

where

$$g = [x - {}^{c_1}u, \ldots, x - {}^{c_k}u]_\ell, \tag{30}$$

the least common left multiple in R of $x - {}^{c_1}u, \ldots, x - {}^{c_k}u$, for some $c_1, \ldots, c_k \in K^*$.

Proof. Observe that the left $\mathcal{R}$–module $R/R(x-u)$ is simple because it is of dimension 1 as a left K–vector space. By Jacoboson-Bourbaki's Theorem (see [33, Theorem 4.1]), λ gives a ring isomorphism $\mathcal{R} \cong \mathrm{End}(K_{K^{\varphi_u}})$, so $\mathcal{R}$ is isomorphic to a full matrix ring with coefficients in K^{φ_u}. We know thus that every simple left $\mathcal{R}$–module is isomorphic to $R/R(x-u)$. This entails (see, e.g., [13, pp. 40-41]) that every maximal left ideal of $\mathcal{R}$ is of the form $\mathcal{R}(x-{}^c u)$ for a suitable non-zero $c \in K$. Since every left ideal of $\mathcal{R}$ is the intersection of finitely many maximal left ideals, we get the description (30). □

Our next aim is to discuss when the representation (30) is irredundant, which will also lead to the computation of a parity-check matrix of the code C. We will use that the non-commutative evaluation defined in (28) obeys some rules, which are to be recalled.

Following [22] define by recursion, for $a \in K$:

$$N_0(a) = 1,$$

$$N_{n+1}(a) = \sigma(N_n(a))a + \delta(N_n(a)).$$

If $g(x) = \sum_i g_i x^i \in K[x;\sigma,\delta]$ then, by [22, Lemma 2.4],

$$g[a] = \sum_i g_i N_i(a). \tag{31}$$

Following [9], define the *Vandermonde matrix*

$$V_n(c_1,\ldots,c_k) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ c_1 & c_2 & \cdots & c_k \\ N_2(c_1) & N_2(c_2) & \cdots & N_2(c_k) \\ \vdots & \vdots & \ddots & \vdots \\ N_{n-1}(c_1) & N_{n-1}(c_2) & \cdots & N_{n-1}(c_k) \end{pmatrix}$$

and the *Wronskian matrix*

$$W_n^u(c_1,\dots,c_k) = \begin{pmatrix} c_1 & c_2 & \cdots & c_k \\ \varphi_u(c_1) & \varphi_u(c_2) & \cdots & \varphi_u(c_k) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_u^{n-1}(c_1) & \varphi_u^{n-1}(c_2) & \cdots & \varphi_u^{n-1}(c_k) \end{pmatrix}$$

for each $n \geq 1$. Then, by [9, Proposition 4.4],

$$V_n({}^{c_1}u,\dots,{}^{c_k}u)diag(c_1,..,c_k) = W_n^u(c_1,\dots,c_k), \quad (32)$$

where $diag(c_1,\dots,c_k)$ denotes the diagonal matrix built from the list $c_1,\dots,c_k$.

Proposition 24. *Let $\{c_1,\dots,c_k\} \subseteq K^*$ be a linearly independent set over K^{φ_u}, with $k \leq m-1$, and set*

$$g = [x - {}^{c_1}u,\dots,x - {}^{c_k}u]_\ell.$$

Then $deg(g) = k$, g is a right divisor of μ, and $\mathfrak{v}(\mathcal{R}g)$ is the left kernel of the Wronskian matrix

$$W_m^u(c_1,\dots,c_k).$$

In other words, $C = \mathfrak{v}(\mathcal{R}g)$ is a K–linear skew-differential code of dimension $m-k$ with parity-check matrix $W_m^u(c_1,\dots,c_k)$.

Proof. By [9, Theorem 5.3], $deg(g) = k$. On the other hand, $f = \sum_{k=0}^{m-1} f_k x^k \in \mathcal{R}g$ if and only if $x - {}^{c_j}u$ right divides f for all $j = 1,\dots,m$. This is equivalent, by (28) and (31), to the condition

$$(f_0,\dots,f_{m-1})V_m({}^{c_1}u,...,{}^{c_k}u) = 0.$$

By (32), this is equivalent to the condition

$$(f_0,\dots,f_{m-1})W_m^u(c_1,\dots,c_k) = 0$$

as required. □

We are now ready to locate our RS skew-differential codes within the class of all (σ,δ,u)–codes.

Corollary 25. *The code $C_{(\varphi_u,\alpha,d)}$ is a (σ,δ,u)–code given by $C_{(\varphi_u,\alpha,d)} = \mathfrak{v}(\mathcal{R}g)$, where*

$$g = [x - {}^{\alpha}u, x - {}^{\varphi_u(\alpha)}u,\dots,x - {}^{\varphi_u^{d-2}(\alpha)}u]_\ell.$$

Remark 26. The code $C_{(\varphi_u,\alpha,d)}$ was defined by means of its parity-check matrix H (see Definition 2). Of course, in order to specify the encoding of messages, one may use the standard method for linear codes of constructing a generator matrix from H, as done in Subsections II-A and II-B.

An alternative is to use the arithmetic of the ring $\mathcal{R}$. Indeed, one may compute the skew polynomial g from Corollary 25 and use it as an encoder similarly to the commutative cyclic case. As for the computation of g concerns, one may use the non-commutative extended Euclidean algorithm (see, e.g. [7, Ch. I, Theorem 4.33]). For instance, in the example described in Subsection II-A, the set of conjugates becomes

$$\{{}^{\alpha}u, {}^{\varphi_u(\alpha)}u, {}^{\varphi_u^2(\alpha)}u, {}^{\varphi_u^3(\alpha)}u\} = \{a^{137}, a^{212}, a^{141}, a^{225}\},$$

so that a generator polynomial of this code is

$$g = \left[x - a^{137}, x - a^{212}, x - a^{141}, x - a^{225}\right]_\ell = x^4 + a^{187}x^3 + a^{99}x^2 + a^{98}x + a^{218}.$$

Remark 27. Module (σ,δ)–codes over a finite field $\mathbb{F}$ generated be a polynomial of the form $[x-\alpha_1,\dots,x-\alpha_n]_\ell \in \mathbb{F}[x;\sigma,\delta]$ have been proved to be MDS in [6, Theorem 5] whenever $\alpha_1,\dots,\alpha_n \in \mathbb{F}$ are suitable powers of an element in the algebraic closure of $\mathbb{F}$ subject to additional conditions called "Hamming 1" and "Hamming 2". These codes are different, in the case $K = \mathbb{F}$, from that of Corollary 25, which are known to be MDS by Theorem 13. Also, a decoding algorithm, different from Algorithm 1 was designed in [6], under the condition "Hamming 1" which, in particular, requires $\delta = 0$.

Next, we analyze how skew RS codes from [17] and RS differential convolutional codes [18] are particular examples of RS skew-differential codes. In this way, Algorithm 1 both extends to a considerable broader class of codes and, also, simplifies the decoding algorithms designed in [17] and [18].

Example 28. Let σ be an automorphism of K of finite order m, and choose a cyclic vector α of σ as a vector space over K^σ. Set $\beta = \alpha^{-1}\sigma(\alpha)$ and

$$g = [x-\beta, x-\sigma(\beta),\dots,x-\sigma^{d-2}(\beta)]_\ell,$$

the left least common multiple being computed in $K[x;\sigma]$. Skew RS codes from [17] are defined as $\mathfrak{v}(\mathcal{R}g)$, where

$$\mathcal{R} = K[x;\sigma]/\langle x^m - 1\rangle.$$

Since $1^\alpha = \beta$, and $\varphi_1 = \sigma$ we see, after Corollary 25, that

$$\mathfrak{v}(\mathcal{R}g) = C_{(\sigma,\alpha,d)}.$$

That is, for $u = 1$ and $\delta = 0$, we obtain the skew RS codes from [17]. Thus, we may apply the decoding algorithm presented here, which is simpler than that of [17].

Example 29. Let δ be any $\mathbb{F}$–linear derivation of the the field $\mathbb{F}(t)$ of rational functions in the variable t with coefficients in a finite field $\mathbb{F}$. If p is the characteristic of $\mathbb{F}$, then the degree of $\mathbb{F}(t)$ over the field of constants $\mathbb{F}(t)^\delta$ is p, and the minimal polynomial of δ becomes $\mu = x^p - \gamma x$, where $\gamma = \delta^p(t)/\delta(t)$ (see [18] for details). For any $c \in \mathbb{F}(t)$, the logarithmic derivative is defined as $L(c) = c^{-1}\delta(c)$. Choose a cyclic vector α for the $\mathbb{F}(t)^\delta$–linear map δ and set

$$g = [x-L(\alpha), x-L(\delta(\alpha)),\dots,x-L(\delta^{d-2}(\alpha))]_\ell \in \mathbb{F}(t)[x;\delta].$$

In [18], the differential convolutional RS codes are defined as $\mathfrak{v}(\mathcal{R}g)$, where, this time,

$$\mathcal{R} = \mathbb{F}(t)[x;\delta]/\langle x^p - \gamma x\rangle.$$

Since $0^\alpha = L(\alpha)$ and $\varphi_0 = \delta$, we deduce from Corollary 25 that

$$\mathfrak{v}(\mathcal{R}g) = C_{(\delta,\alpha,d)}.$$

In other words, we obtain the differential convolutional RS codes from [18] by setting $\sigma = id_{\mathbb{F}(t)}$ and $u = 0$, to which we also may apply the decoding algorithm presented in this paper. Again, it results simpler than that from [18].

Our last remark in this section deals with the relationship, kindly pointed out by one of the referees, between the codes to which our decoding algorithm applies to, and those introduced in [26].

Remark 30. Reed-Solomon skew-differential codes are also related to the class of linearized Reed-Solomon codes defined in [26, Definition 31] whenever the base ring is a field. Concretely, the dual of a Reed-Solomon skew-differential code is, as a vector subspace of K^m, a linearized Reed-Solomon code. Observe that the operator φ_u is exactly the operator described in [26, Definition 20], so that, the matrix H described in Definition 2 is the transpose of the generator matrix given in [26, Definition 31] by setting $\ell = 1$, $n_1 = m$, $a^{(1)} = u$ and $\beta_i^{(1)} = \varphi_u^{i-1}(\alpha)$ for $i = 1, \ldots, m$. Observe also that the condition of $\{\beta_1^{(1)}, \ldots, \beta_m^{(1)}\}$ being linearly independent over the centralizer [22] is equivalent to the matrix A in Proposition 1 being invertible.

A decoding algorithm for linearized Reed-Solomon codes is described in [3]. It works for the skew metric as defined in [26], which is in general different from the Hamming metric. Another decoding algorithm for these codes, for a sum-rank metric, is presented in [8] in the case $\delta = 0$. Linearized Goppa codes are then introduced and interpreted as duals, with respect to a suitable bilinear form, to linearized RS codes. One should not expect that some of them become RS skew-differential codes in our sense, as the duality stated in this context is not the usual.

Fast decoding algorithms for linearized RS codes with coefficients in a finite field appeared recently in [27] and [2].

APPENDIX

A. Computation of the cyclic vector.

Let us briefly discuss why a randomized calculus, as proposed in Remark 9, of the cyclic vector α for the K^{φ_u}–linear map $\varphi_u : K \to K$ is a reasonable method. First, recall that $\alpha \in K$ is such a cyclic vector whenever $\{\alpha, \varphi_u(\alpha), \ldots, \varphi_u^{m-1}(\alpha)\}$ is a K^{φ_u}–basis of K. Equivalently, α is a generator of K as a module over the commutative polynomial ring $K^{\varphi_u}[X]$, where the action of the indeterminate X on K is given by $Xb = \varphi_u(b)$ for all $b \in K$.

We know that K is already a cyclic $K^{\varphi_u}[X]$–module by Proposition 11 and, what is more, $K \cong K^{\varphi_u}[X]/\langle \mu \rangle$ as modules over $K^{\varphi_u}[X]$, where μ is the minimal polynomial of φ_u as a K^{φ_u}–linear map. Henceforth, cyclic vectors for φ_u are in bijective correspondence with polynomials in $K^{\varphi_u}[X]$ of degree up to $m-1$ which are coprime with μ. We see, thus, that, if K is not finite, then almost every element in K becomes a cyclic vector for a fixed φ_u.

For finite fields, there is an explicit formula expressing the number of polynomials with degree smaller than that of a given polynomial and coprime with it [23, Lemma 3.69]. Setting in our case $K^{\varphi_u} = \mathbb{F}_q$, when K is finite, and $n_1, \ldots, n_r$ the degrees of the distinct irreducible factors appearing in the canonical factorization of the minimal polynomial $\mu \in \mathbb{F}_q[X]$, we obtain that the probability for a given $\alpha \in K$ to be a cyclic vector is

$$(1 - q^{-n_1}) \cdots (1 - q^{-n_r}).$$

Of course, there also exists the possibility of computing a cyclic vector by means of the classical algorithm based upon the calculus, by elementary row and column transformations, of a diagonal matrix equivalent to the characteristic matrix of φ_u with respect to a given basis (see, e.g. [20, pp. 195–198]). This deterministic method, in contrast with our preferred randomized method, requires the computation of K^{φ_u} and of a basis of K as a vector space over this subfield in order to get the matrix with coefficients in K^{φ_u} representing φ_u.

B. Complexity calculation.

We state here some guidelines about the time complexity of Algorithm 1. In general, since we work over an arbitrary field, we calculate it with respect to the number of operations (additions, multiplications, applications of σ and δ) on the base field. Step 1 is simply obtained by the product yH, where H is the parity-check matrix in Definition 2, so it belongs to $\mathcal{O}(md)$. Step 2 requires to compute the matrix S_τ in the worst case, which can be done in $\mathcal{O}(\tau^2)$, and the calculation of the rank of S_r for $1 \leq r \leq \tau$. The traditional approach to compute the rank is by Gaussian elimination, which can be done in $\mathcal{O}(\tau^\omega)$, where ω is the matrix multiplication exponent. Since matrices are relatively small, we may consider the classical algorithm and set $\omega = 3$. So that Step 2 can be performed in $\mathcal{O}(\tau^3)$. Step 3 can be done by the execution of an algorithm that outputs the row reduced echelon form, whose execution time is in $\mathcal{O}(\tau^3)$ by using the standard algorithm. Matrix L in Step 4 can be computed in $\mathcal{O}(m(m-\tau))$ operations. Assuming that A is pre-calculated, the product LA is in $\mathcal{O}(m^2(m-\tau))$ with the standard matrix multiplication algorithm. An efficient way of dealing with Step 5 consists of computing the row reduced echelon form LA_{rref} of LA, and check which unitary vectors are not rows of LA_{rref}, so the runtime is in $\mathcal{O}(m^2(m-\tau))$. Finally, in Step 6 we need to solve a linear system whose coefficient matrix has order τ, so this step can be done in $\mathcal{O}(\tau^3)$ operations. The complexity is then dominated by the matrix product and the row reduced echelon form computations, and Algorithm 1 can be executed in $\mathcal{O}(m^3 + \tau^3)$ operations on the base field. That is to say, since τ is $\lfloor \frac{m-k+1}{2} \rfloor$, where k is the dimension of the code, Algorithm 1 is in $\mathcal{O}(m^3)$. Obviously, this bound can be improved if we use faster algorithms for matrix multiplication and row reduced echelon form computation. Nevertheless, a detailed study of this issue is out the scope of the paper.

REFERENCES

[1] S.B. Balaji, M. Nikhil Krishnan, Myna Vajha, Vinayak Ramkumar, Birenjith Sasidharan, P. Vijay Kumar, *Erasure coding for distributed storage: an overview.* Sci. China Inf. Sci. 61, 100301 (2018).

[2] H. Bartz, T. Jerkovits, S. Puchinger, and J. Rosenkilde, *Fast decoding of codes in the rank, subspace, and sum-rank metric*, IEEE Trans. Inform. Theory **67** (2021), 5026–5050.

[3] D. Boucher, *An algorithm for decoding skew Reed-Solomon codes with respect to the skew metric,* Designs, Des. Codes Cryptogr. **88** (2020), 1991–2005.

[4] D. Boucher, W. Geiselmann, and F. Ulmer. *Skew-cyclic codes*. Appl. Algebr. Eng. Comm **18** (2007), 379–389.

[5] D. Boucher, F. Ulmer. *Coding with skew polynomial rings.* J. Symb. Comp. **44** (2009), 1644–1656.

[6] D. Boucher, F. Ulmer. *Linear codes using skew polynomials with automorphisms and derivations.* Des. Codes Cryptogr. **70** (2014) 405–431.

[7] J. L. Bueso, J. Gómez-Torrecillas, and A. Verschoren. *Algorithmic methods in Non-Commutative Algebras. Applications to Quantum Groups*. Springer, Dordretch, 2003.

[8] X. Caruso, Residues of skew rational functions and linearized Goppa codes, preprint. https://arxiv.org/abs/1908.08430.

[9] J. Delenclos and A. Leroy. *Noncommutative symmetric functions and W-polynomials.* Journal of Algebra and Its Applications, **6** (2007), 815–837.

[10] G. D. Forney. *Convolutional codes I: Algebraic structure.* IEEE Trans. Inform. Theory, **16** (1970) 720–738.

[11] E. M. Gabidulin, Theory of codes with maximum rank distance, Problems of Information Transmission 21 (1) (1985), 1–12.

[12] H. Gluesing-Luerssen and W. Schmale. *On cyclic convolutional codes.* Acta Appl. Math. **82** (2004), 183–237.

[13] J. Gómez-Torrecillas. *Basic Module Theory over Non-commutative Rings with Computational Aspects of Operator Algebras.* In: Algebraic and Algorithmic Aspects of Differential and Integral Operators, M. Barkatou, T. Cluzeau, G. Regensburger, M. Rosenkranz, eds. LNCS 8372, pages 23-82, Springer, 2014.

[14] J. Gómez-Torrecillas, F. J. Lobillo and G. Navarro. *A new perspective of cyclicity in convolutional codes.* IEEE Trans. Inform. Theory **62** (2016), 2702–2706.

[15] J. Gómez-Torrecillas, F. J. Lobillo, and G. Navarro. *Ideal codes over separable ring extensions.* IEEE Trans. Inform. Theory, **63** (2017), 2796–2813.

[16] J. Gómez-Torrecillas, F. J. Lobillo, and G. Navarro. *A Sugiyama-like decoding algorithm for convolutional codes.* IEEE Trans. Inform. Theory, **63** (2017) 6216–6226.

[17] J. Gómez-Torrecillas, F. J. Lobillo, and G. Navarro. *Peterson-Gorenstein-Zierler algorithm for skew RS codes.* Linear and Multilinear Algebra, **66** (2018) 469–487.

[18] J. Gómez-Torrecillas, F. J. Lobillo, G. Navarro, and P. Sánchez-Hernández. *Peterson-Gorenstein-Zierler Algorithm for differential convolutional codes.* Appl. Algebr. Eng. Comm., **32** (2021) 321344

[19] D. Gorenstein, and N. Zierler. *A class of error-correcting codes in p^m symbols.* J. Soc. Ind. Appl. Math. **9** (1961), 207–214.

[20] N. Jacobson. *Basic Algebra 1*. 2nd Ed. W. H. Freeman and Co., 1985.

[21] N. Jacobson. *Finite-dimensional division algebras over fields.* Springer-Verlag, 1996. Corrected 2nd print 2010.

[22] T. Y. Lam, and A. Leroy. *Vandermonde and wronskian matrices over division rings.* J. Algebra, **119** (1988), 308–336.

[23] R. Lidl, and H. Niederreiter. *Finite Fields.* 2nd Ed. Cambrigde University Press, 1997.

[24] S. Liu, F. Manganiello, and F. R. Kschischang. *Construction and decoding of generalized skew-evaluation codes.* In: 2015 IEEE 14th CanadianWorkshop on Information Theory (CWIT), St. John's, NL, pages 9–13, 2015.

[25] S. R. López-Permouth, and S. Szabo. *Convolutional codes with additional algebraic structure.* J. Pure Appl. Algebra **217** (2013), 958–972.

[26] U. Martínez-Peñas. *Skew and linearized Reed-Solomon codes and maximum sum rank distance codes over any division ring.* J. Algebra **504** (2018) 587-612.

[27] U. Martínez-Peñas and F. R. Kschischang, *Reliable and secure multishot network coding using linearized reed-solomon codes,* IEEE Transactions on Information Theory 65 (8) (2019), 4785–4803.

[28] W. W. Peterson. *Encoding and error-correction procedures for the Bose-Chaudhuri codes. IRE.* Trans. Inform. Theory **6** (1960), 459–470.

[29] P. Piret. *Structure and constructions of cyclic convolutional codes.* IEEE Trans. Inform. Theory, **22** (1976), 147–155.

[30] C. Roos. *On the structure of convolutional and cyclic convolutional codes.* IEEE Trans. Inform. Theory, **25** (1979), 673–686.

[31] SageMath, the Sage Mathematics Software System (Version 8.2), The Sage Developers, 2019, http://www.sagemath.org.

[32] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. A. Namekawa. *Method for solving key equation for decoding Goppa codes.* Inform. Control, **27** (1975), 87–99.

[33] M. Sweedler. *The predual theorem to the Jacobson-Bourbaki theorem.* Transactions of the American Mathematical Society, **213** (1975) 391–406.

[34] D. J. Winter. *The structure of fields.* Springer-Verlag New York, 1974.

José Gómez-Torrecillas received the Ph. D. degree (1992) in Mathematics from the University of Granada, where he is a Professor in the Department of Algebra since 2000. He has authored about one hundred research papers on pure and applied aspects of Algebra. He is coauthor of the book *Algorithms in Non-Commutative Algebra: Applications to Quantum Groups* (Springer, 2003). He has been member of the steering committee of ISSAC from 2011 to 2014 and head of more than 10 research projects. Currently, he is head of the research group *Algebra and Information Theory* at the University of Granada. He serves as the editor for Algebra for the *Transactions of A. Razmadze Mathematical Institute* and as a member of the Editorial Board of *Journal of Algebra and its Applications*. His current research interests include Algebraic Coding Theory, algebraic and categorical aspects of Quantum Groupoids, and algorithms in Non-Commutative Algebra.

Gabriel Navarro is an Associated Professor with the Department of Computer Sciences and Artificial Intelligence at the University of Granada. He is also member of the Research Centre for Information and Communications Technologies of the University of Granada (CITIC-UGR). He received its Ph.D. degree in mathematics (2006) from the University of Granada. His current research interests are coding theory, computational algebra and fuzzy and rough sets.

José Patricio Sánchez-Hernández received the Ph.D. degree (2016) in mathematics from the National Autonomous University of Mexico (UNAM). From 2018 to 2020, he was in a Postdoctoral Stay in the research group Algebra and Information Theory at the University of Granada. His current research interests are Coding Theory, Theory of Modules and Lattice Theory.