Iterative decoding of skew constacyclic codes

Epiphane Kayodé Nouetowa, Ivan Pogildiakov

Iterative decoding of skew constacyclic codes

E.K. Nouetowa and I. Pogildiakov

Univ Rennes, CNRS, IRMAR - UMR 6625, Rennes Cedex, France
kayode-epiphane.nouetowa@univ-rennes.fr, ivan.pogildiakov@gmail.com

Abstract. The aim of this note is to design an iterative decoding algorithm for skew constacyclic codes defined over finite fields, which is inspired from [2] and [3]. We analyse the algorithm in the single error case, and use computer simulations in the general one.

Acknowledgments.

This work was conducted within the the France 2030 framework porgramme, Centre Henri Lebesgue ANR−11−LABX−0020 − 01.

1 Introduction

Skew cyclic codes are a subclass of linear codes containing the cyclic codes. These codes and their decoding algorithms have been the subject of several works [4,8,10]. Recently, M. Bossert proposed an iterative decoding algorithm for binary cyclic codes [2] using the minimum weight codewords of "dual" codes. Later, M. Bossert et al. extended that work to non-binary cyclic codes [3]. The aim of this note is to adapt these algorithms to skew constacyclic codes using Euclidean duals.

The text is organized as follows. In Section 2, we recall the definition of skew constacyclic codes and a characterization of their Euclidean duals. In Section 3, we give our decoding strategy and show the link with the strategy applied in [3]. In Section 4, we initialize an analysis of the resulting iterative decoding algorithm and provide a condition under which the algorithm always fails.

2 Some generalities on skew constacyclic codes

Recall that a **linear code** $\mathcal{C}$ over a finite field $\mathbb{F}_q$ of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_q^n$. The **Euclidean dual** $\mathcal{C}^\perp$ of $\mathcal{C}$ is defined as

$$\mathcal{C}^\perp = \{v \in \mathbb{F}_q^n \mid \langle v, c\rangle = 0 \text{ for all } c \in \mathcal{C}\},$$

where $\langle x, y\rangle = \sum_{i=0}^{n-1} x_i y_i$ is the Euclidean scalar product of $x = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_q^n$ and $y = (y_0, \ldots, y_{n-1}) \in \mathbb{F}_q^n$. The **minimum distance** d of $\mathcal{C}$ is the smallest

of the (Hamming) weights of the non-zero codewords. Furthermore, a linear code $\mathcal{C}$ is called cyclic if for all $c = (c_0, \ldots, c_{n-1})$ in $\mathcal{C}$ the vector $(c_{n-1}, c_0, \ldots, c_{n-2})$ also belongs to $\mathcal{C}$.

Let θ be an automorphism of $\mathbb{F}_q$, and let ε be a non-zero element of $\mathbb{F}_q$. Consider the map

$$\phi_\varepsilon : \begin{cases} \mathbb{F}_q^n & \longrightarrow \mathbb{F}_q^n, \\ (a_0, ..., a_{n-1}) & \longmapsto (\varepsilon\theta(a_{n-1}), \theta(a_0), \ldots, \theta(a_{n-2})). \end{cases}$$

Skew constacyclic codes are defined as follows.

Definition 1 (Definition 1 of [6]). *A (θ, ε)-**constacyclic code** $\mathcal{C}$ is a linear code over $\mathbb{F}_q$ such that for any $c = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n$,*

$$c \in \mathcal{C} \Rightarrow \phi_\varepsilon(c) \in \mathcal{C}.$$

*If $\varepsilon = 1$, then $\mathcal{C}$ is called θ-**cyclic**. If $\varepsilon = -1$, then $\mathcal{C}$ is called θ-**negacyclic**.*

Note that if θ is the identity, then a θ-cyclic code (resp. θ-negacyclic) is a cyclic (resp. negacyclic) code.

The skew polynomial ring $(R, +, \cdot)$, or Ore ring [11], is the set of polynomials $\mathbb{F}_q[x; \theta]$ over $\mathbb{F}_q$ equipped with the usual component-wise addition '+' and where the multiplication '$\cdot$' is defined by the rule

$$x \cdot a = \theta(a)x \text{ for all } a \in \mathbb{F}_q.$$

Clearly, the ring R is non-commutative if θ is different from the identity. It is well known that R is a left and right Euclidean ring. Moreover, the center of R is the commutative polynomial ring $\mathbb{F}_q^\theta[x^{|\theta|}]$, where $\mathbb{F}_q^\theta$ stands for the subfield of $\mathbb{F}_q$ fixed by θ, and $|\theta|$ is the order of θ.

In this text we use the conventional representation of the elements $c = (c_0, \ldots, c_{n-1})$ of $\mathbb{F}_q^n$ as skew polynomials $c(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}$ of degree less than n. Under this correspondence, a (θ, ε)-constacyclic code $\mathcal{C}$ can be viewed as a left R-submodule $Rg(x)/R(x^n - \varepsilon)$ of $R/R(x^n - \varepsilon)$ [6], where $g(x)$ is a monic skew right divisor of $x^n - \varepsilon$ called **skew generator polynomial** of the code $\mathcal{C}$. The dimension of $\mathcal{C}$ is $k = n - \deg(g(x))$. One has

$$\mathcal{C} = \{m(x)g(x) \mid m(x) \in R, \deg(m(x)) < k\}.$$

We write $\mathcal{C} = (g)_{n,\theta}^{\varepsilon}$. To characterize the Euclidean dual of $\mathcal{C}$ we recall below the definition of the skew reciprocal polynomial of a skew polynomial in R.

Definition 2 (Definition 3 of [6]). *Let $h = \sum_{i=0}^{k} h_i x^i \in R$ be a skew polynomial of degree k. The* **skew reciprocal polynomial** *$h^*(x)$ of $h(x)$ is the skew polynomial*

$$h^*(x) = \sum_{i=0}^{k} x^{k-i} \cdot h_i = \sum_{i=0}^{k} \theta^i(h_{k-i})x^i.$$

The **left monic skew reciprocal polynomial** $h^\natural(x)$ *of* $h(x)$ *is*

$$h^\natural(x) = \frac{1}{\theta^{k-m}(h_m)} h^*(x),$$

where $m = \min\{i \mid h_i \neq 0\}$.

The following proposition gives the Euclidean dual of a (θ, ε)-constacyclic code. Notice that we extend the automorphism θ of $\mathbb{F}_q$ to the automorphism $\theta\colon R \to R$ of R by linearity, i.e. by the rule $\sum_{i=0}^{k} a_i x^i \mapsto \sum_{i=0}^{k} \theta(a_i) x^i$.

Proposition 1 (Theorem 1 of [6]). *Let* $\mathcal{C} = (g)^{\varepsilon}_{n,\theta}$ *be a* (θ, ε)*-constacyclic code over* $\mathbb{F}_q$*. The Euclidean dual* $\mathcal{C}^\perp$ *of* $\mathcal{C}$ *is a* $(\theta, 1/\varepsilon)$*-constacyclic code over* $\mathbb{F}_q$ *defined as* $\mathcal{C}^\perp = (h^\natural)^{1/\varepsilon}_{n,\theta}$*, where* h *is the monic skew polynomial such that*

$$x^n - \varepsilon = \theta^n(h(x))g(x).$$

The skew polynomial $h(x)$ *is called* **skew check polynomial** *of the code* $\mathcal{C}$*.*

Example 1 *Let* $\mathbb{F}_4 = \mathbb{F}_2(a)$ *with* $a^2 + a + 1 = 0$*,* $\theta\colon u \mapsto u^2 \in Aut(\mathbb{F}_4)$*, and* $R = \mathbb{F}_4[x; \theta]$*. Let* $n = 12$*. Consider* $g(x) = x^8 + a^2x^6 + x^5 + x^4 + x^3 + x + a \in R$*. One has* $x^{12} - 1 = h(x)g(x)$*, where* $h(x) = x^4 + a^2x^2 + x + a^2$*. The skew polynomial* $g(x)$ *generates a* θ*-cyclic code* $\mathcal{C}$ *of length* 12*, dimension* 4 *and minimum distance* 7 *over* $\mathbb{F}_4$*.*

We have $h^*(x) = a^2x^4 + x^3 + a^2x^2 + 1$ *and* $h^\natural(x) = x^4 + ax^3 + x^2 + a$*. Therefore, the dual* $\mathcal{C}^\perp$ *of* $\mathcal{C}$ *is the* θ*-cyclic code over* $\mathbb{F}_4$ *with skew generator polynomial* $h^\natural(x)$ *having length* 12*, dimension* 8*, and minimum distance* 4*.*

The following technical lemma finds its use in next section.

Lemma 1 (Lemma 1 of [6] and Lemma 7 of [5]). *Let* $f(x)$ *and* $g(x)$ *be skew polynomials in* R*, and let* $k = \deg(f)$*. One has*

1. $(f(x)g(x))^* = \theta^k(g(x)^*)f(x)^*$.
2. *If the constant coefficient of* f *is nonzero, then* $(f(x)^*)^* = \theta^k(f(x))$.
3. *If* $f(x)g(x)$ *is central, then* $f(x)g(x) = g(x)f(x)$.

3 Decoding strategy

Let $\mathcal{C}$ be a (θ, ε)-constacyclic code over $\mathbb{F}_q$ of length n with skew generator polynomial $g(x)$ and skew check polynomial $h(x)$. In this section, we assume that ε belongs to the field $\mathbb{F}_q^\theta$ fixed by θ and that the order $|\theta|$ of θ divides the length n of the code $\mathcal{C}$. Therefore, $x^n - \varepsilon$ is a central polynomial, $R/(x^n - \varepsilon)$ is a left principal ideal ring and $\mathcal{C}$ is a left ideal $(g(x))/(x^n - \varepsilon)$. Furthermore, according to Point 3 of Lemma 1, we have

$$x^n - \varepsilon = h(x)g(x) = g(x)h(x). \tag{1}$$

Lemma 2. *Given $u \in \mathcal{C}^\perp$ and $c \in \mathcal{C}$, the following holds*:

$$c(x)\theta^{-\ell}(u(x)^*) \equiv 0 \pmod{x^n - \varepsilon},$$

where $\ell = \deg(u(x))$.

Proof. Let $c \in \mathcal{C} = (g)^{\varepsilon}_{n,\theta}$ and $u \in \mathcal{C}^\perp$. Consider $m(x)$ and $v(x)$ in R such that $c(x) = m(x)g(x)$ and $u(x) = v(x)h(x)^*$. According to Lemma 1, we have

$$\begin{aligned} u(x)^* &= \theta^{\deg(v(x))}((h(x)^*)^*)v(x)^* && \text{(Point 1. of Lemma 1)} \\ &= \theta^{\deg(v(x))+\deg(h(x))}\left(h(x)\right)v(x)^* && \text{(Point 2. of Lemma 1)} \\ &= \theta^{\ell}\left(h(x)\right)v(x)^* && \text{(because } \deg(h^*(x)) = \deg(h(x))\text{)}. \end{aligned}$$

Therefore, we have $\theta^{-\ell}\left(u(x)^*\right) = h(x)\theta^{-\ell}\left(v(x)^*\right)$ and

$$\begin{aligned} c(x)\theta^{-\ell}(u(x)^*) &= m(x)g(x)h(x)\theta^{-\ell}(v(x)^*) \\ &= m(x)(x^n - \varepsilon)\theta^{-\ell}(v(x)^*) && \text{(according to (1))} \\ &= m(x)\theta^{-\ell}(v(x)^*)(x^n - \varepsilon) && \text{(because } x^n - \varepsilon \text{ is central)} \\ &\equiv 0 \pmod{x^n - \varepsilon}. \end{aligned}$$

■

Remark 1. The polynomials $u(x)$ and $\theta^{-\ell}(u(x)^*)$ have the same Hamming weight.

Definition 3. *Two non-zero words c_1 and $c_2 \in \mathbb{F}_q^n$ are called (θ, ε)-cyclically equivalent if there exist b in $\mathbb{F}_q$ and i in $\mathbb{N}$ such that $c_2 = b\phi_\varepsilon^i(c_1)$ which means that $c_2(x) = bx^i c_1(x) \bmod (x^n - \varepsilon)$. In this case we write $c_1 \sim_{\theta,\varepsilon} c_2$.*

One can show that $\sim_{\theta,\varepsilon}$ is an equivalence relation. Each class of (θ, ε)-cyclically equivalent words in $\mathbb{F}_q^n$ contains a monic representative (considered as a polynomial).

Now we pick the following two sets of words playing an important role in the decoding strategy. Let w be a positive integer. We define

- $\mathcal{B}_w = \{\text{all monic representatives in } \mathcal{C}^\perp/\sim_{\theta,\frac{1}{\varepsilon}} \text{ of Hamming weight } w\}$,
- $\overline{\mathcal{B}}_w = \{\theta^{-\ell}(u(x)^*) \mid u \in \mathcal{B}_w \text{ and } \ell = \deg(u(x))\}$.

Remark 2. Clearly, the set $\mathcal{B}_w$ is not unique. But, however, one can show that the choice of one or the other has no influence on the construction of the frequency matrix that we detail in what follow.

The elements of $\mathcal{B}_w$ are monic skew polynomials. Therefore, the skew reciprocal polynomial of each element of $\mathcal{B}_w$ has constant coefficient one. According to Remark 1, each element of $\overline{\mathcal{B}}_w$ has Hamming weight w. Thus, the elements of $\overline{\mathcal{B}}_w$ are of the form

$$1 + \lambda_{\beta_1}x^{\beta_1} + \cdots + \lambda_{\beta_{w-1}}x^{\beta_{w-1}} \in \mathbb{F}_q[x;\theta],$$

where $\beta_1, \dots \beta_w$ are distinct elements of $\{1, \dots, n-1\}$, and $\lambda_{\beta_1}, \dots, \lambda_{\beta_{w-1}}$ are non-zero elements of $\mathbb{F}_q$.

Unless otherwise stated, in the following we denote by $y = c + e$ a received word, where $c \in \mathcal{C}$ is a codeword, $e \in \mathbb{F}_q^n$ is an error of Hamming weight τ at most the half distance bound.

Remark 3. If $u \in \mathcal{C}^\perp$, i.e. $u(x) = v(x)h(x)^*$ with $\deg(v(x)) < n - \deg(h(x))$, then according to Lemma 1(see also proof of Lemma 2) $\theta^{-\ell}(u(x)^*) = h(x)\theta^{-\ell}(v(x)^*)$. If θ is the identity and $\varepsilon = 1$, then, for u in $\mathcal{C}^\perp$, $\theta^{-\ell}(u^*)$ is in fact a codeword of the cyclic code of length n and of generator polynomial $h(x)$. This code is called "dual" code in [2,3], and we would like to emphasis that this "dual" code is not the Euclidean dual $\mathcal{C}^\perp$, unless $h(x)$ is self-reciprocal.

Proposition 2. *Consider $f(x) \in \overline{\mathcal{B}}_w$, we have*

$$y(x)f(x) \equiv e(x)f(x) \pmod{x^n - \varepsilon}.$$

Proof. Lemma 2 implies $c(x)f(x) \equiv 0 \pmod{x^n - \varepsilon}$. Therefore, as $y(x) = c(x) + e(x)$, we have $y(x)f(x) = c(x)f(x) + e(x)f(x) \equiv e(x)f(x) \pmod{x^n - \varepsilon}$. ∎

Let $d^\perp$ be the minimum distance of $\mathcal{C}^\perp$. We want to concentrate our attention on the sets $\mathcal{B}_{d^\perp}$ and $\overline{\mathcal{B}}_{d^\perp}$.

Example 2 (Example 1, continued) *We have $d^\perp = 4$. Using* Magma *[1], one finds:* $\overline{\mathcal{B}}_{d^\perp} = \{1 + x^3 + x^6 + x^9, 1 + a^2x + x^2 + ax^4, 1 + a^2x^2 + a^2x^4 + x^9, 1 + x + a^2x^3 + ax^8, 1 + x^4 + x^6 + x^{10}, 1 + a^2x^3 + ax^4 + x^5, 1 + x + x^6 + x^7, 1 + ax + x^4 + ax^9\}$.

Let $f(x) = 1 + \lambda_{\beta_1}x^{\beta_1} + \cdots + \lambda_{\beta_{d^\perp-1}}x^{\beta_{d^\perp-1}}$ be an element of $\overline{\mathcal{B}}_{d^\perp}$. Consider the following skew polynomial in R:

$$\begin{aligned}\omega_f^0(x) &= y(x)f(x) \bmod (x^n - \varepsilon)\\ &= e(x)f(x) \bmod (x^n - \varepsilon)\\ &= e(x) + e(x)\lambda_{\beta_1}x^{\beta_1} + \cdots + e(x)\lambda_{\beta_{d^\perp-1}}x^{\beta_{d^\perp-1}} \bmod (x^n - \varepsilon).\end{aligned}$$

Note that $\omega_f^0(x)$ is the sum of the error $e(x)$ and its shifts by the skew monomials $\lambda_{\beta_i}x^{\lambda_{\beta_i}}$, $i \in \{1, \dots, d^\perp - 1\}$. Therefore, the degree of each monomial in $\omega_f^0(x)$ is an error position or the sum of an error position and a non-zero β_i. Given $i \in \{1, \dots, d^\perp - 1\}$, we define $\omega_f^i(x)$ as follows:

$$\omega_f^i(x) = \omega_f^0(x)\varepsilon^{-1}\theta^{-\beta_i}\left(\lambda_{\beta_i}^{-1}\right)x^{n-\beta_i} \bmod (x^n - \varepsilon), \tag{2}$$

where $\varepsilon^{-1}\theta^{-\beta_i}\left(\lambda_{\beta_i}^{-1}\right)x^{n-\beta_i}$ is the inverse of the skew monomial $\lambda_{\beta_i}x^{\beta_i}$ modulo $x^n - \varepsilon$. Notice that $\omega_f^i(x)$ is the sum of the error $e(x)$ and its shifts.

We need the following list of vectors:

$$\mathcal{L} = \left[\omega_f^i \mid f(x) \in \overline{\mathcal{B}}_{d^\perp}, i \in \{0, \dots, d^\perp - 1\}\right]. \tag{3}$$

Remark that there are exactly $|\overline{\mathcal{B}}| \times d^{\perp}$ elements in $\mathcal{L}$.

Now, we order the elements of the ground finite field $\mathbb{F}_q$ as $\{\sigma_0, \ldots, \sigma_{q-1}\}$, where $\sigma_0 = 0$. Let us build a $q \times n$ frequency matrix as follows:

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{\sigma_0,0} & \mathcal{T}_{\sigma_0,1} & \ldots & \mathcal{T}_{\sigma_0,n-1} \\ \mathcal{T}_{\sigma_1,0} & \mathcal{T}_{\sigma_1,1} & \ldots & \mathcal{T}_{\sigma_1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{T}_{\sigma_{q-1},0} & \mathcal{T}_{\sigma_{q-1},1} & \ldots & \mathcal{T}_{\sigma_{q-1},n-1} \end{bmatrix},$$

where

$$\mathcal{T}_{\sigma_\mu,j} = \left|\{\omega_f^i \in \mathcal{L} \mid (\omega_f^i)_j = \sigma_\mu\}\right|. \tag{4}$$

Note that the rows of $\mathcal{T}$ are enumerated by the elements of $\mathbb{F}_q$. Given $\mu \in \{0, \ldots, q-1\}$ and $j \in \{0, \ldots, n-1\}$, the entry $\mathcal{T}_{\sigma_\mu,j}$ is the number of skew polynomials $\omega_f^i(x)$ in $\mathcal{L}$ having coefficient $\sigma_\mu \in \mathbb{F}_q$ at the position j. Withing our approach, we use $\mathcal{T}$ to take a decision on the error position and the occurrence at this position in the error vector.

The idea of the decoding strategy is based on the fact that the number of ω_f^i in $\mathcal{L}$ having a coefficient equal to $\sigma_0 = 0$ at the error positions is expected to be low (see Section 4 for more analysis). We define:

$$\nu = \min\{\mathcal{T}_{\sigma_0,j} \mid j \in \{0, \ldots, n-1\}\}, \quad \mathcal{P}_\nu = \{j \in \{0, \ldots, n-1\} \mid \mathcal{T}_{\sigma_0,j} = \nu\}. \tag{5}$$

The elements of $\mathcal{P}_\nu$ are taken as the possible error positions. The sum of the elements of each column of $\mathcal{T}$ is the same, i.e. $\sum_{\mu=0}^{q-1} \mathcal{T}_{\sigma_\mu,j} = |\mathcal{L}| = |\overline{\mathcal{B}}| \times d^{\perp}$. Given a position $j \in \mathcal{P}_\nu$, we determine the largest element

$$\mathcal{M}_j = \max\{\mathcal{T}_{\sigma_\mu,j} \mid \mu \in \{1, \ldots, q-1\}\} \tag{6}$$

of the column j. Therefore, we can delete from $y(x)$ the error magnitudes at the identified erroneous positions by replacing $y(x)$ with $y(x) - \sum_{j \in \mathcal{P}_\nu} \sigma_{\mu_j} x^j$, where $\mu_j \in \{1, \ldots, q-1\}$ are such that $\mathcal{T}_{\sigma_{\mu_j},j} = \mathcal{M}_j$. One checks if $y(x)$ is a code-word. Otherwise, one starts again by the calculation of another $\mathcal{T}$ with the new $y(x)$.

The decoding algorithm is summarized in Algorithm 1.

Let us finish this section by demonstrating the work of Algorithm 1 on a concrete example.

Example 3 (Example 1, continued) *Suppose that*

$$y(x) = x^{11} + x^{10} + ax^9 + ax^8 + a^2x^7 + a^2x^5 + a^2x^3 + ax^2 + 1.$$

One can verify that $y(x)$ is not a codeword of the code $\mathcal{C}$ by dividing $y(x)$ on the right by $g(x)$. Therefore, we are going to apply the decoding strategy to $y(x)$ hopping to recover a corrupted codeword.

Consider the following order in the base finite field $\mathbb{F}_4$:

$$\sigma_0 = 0, \sigma_1 = 1, \sigma_2 = a, \sigma_3 = a^2.$$

Algorithm 1 Decoding algorithm for $\mathcal{C} = (g)^{\varepsilon}_{n,\theta}$

Require: $\overline{\mathcal{B}}_{d^\perp}, y(x) = c(x) + e(x)$, where $c \in \mathcal{C}$; i_{max}
Ensure: $c(x)$ or Failure
1: $i = 0$;
2: **while** $y(x) \notin \mathcal{C}$ and $i \leq i_{max}$ **do**
3: Construct $\mathcal{L}$ defined by (3)
4: Construct $\mathcal{T}$ defined by (4)
5: Determine ν and $\mathcal{P}_\nu$ as in (5)
6: Refine $y(x) = y(x) - \sum_{j \in \mathcal{P}_\nu} \sigma_{\mu_j} x^j$
7: $i \leftarrow i + 1$;
8: **end while**
9: **if** $y(x) \in \mathcal{C}$ **then**
10: **return** $y(x)$
11: **else**
12: **return** Failure
13: **end if**

We put $y_0(x) = y(x)$. At the i-th iteration the decoding algorithm constructs a matrix $\mathcal{T} = \left(\mathcal{T}_{\sigma_\mu, j}\right)_{\mu,j}$ from the polynomial $y_{i-1}(x)$ and the list $\overline{\mathcal{B}}_{d^\perp}$.

Iteration 1*. One finds*

$$\mathcal{T} = \begin{bmatrix} 11 & 11 & 3 & 13 & 11 & 8 & 14 & \mathbf{2} & 10 & 14 & 12 & 3 \\ 7 & 5 & 17 & 9 & 9 & 6 & 6 & 4 & 8 & 6 & 6 & 5 \\ 7 & 11 & 7 & 5 & 7 & 8 & 6 & \boxed{18} & 6 & 6 & 8 & 7 \\ 7 & 5 & 5 & 5 & 5 & 10 & 6 & 8 & 8 & 6 & 6 & 17 \end{bmatrix},$$

$$\nu = 2, \quad \mathcal{P}_\nu = 7, \quad \mathcal{M}_7 = 18, \quad \sigma_{\mu_7} = a.$$

The algorithm takes the following desicion: most probably there is an error in $y_0(x)$ at position 7 and most probably the corresponding entry in the error vector at position 7 is $\sigma_2 = a$. Therefore, we put

$$y_1(x) = y_0(x) - ax^7 = x^{11} + x^{10} + ax^9 + ax^8 + x^7 + a^2x^5 + a^2x^3 + ax^2 + 1.$$

One can verify that $y_1(x)$ does not belong to $\mathcal{C}$, and therefore we proceed with the algorithm.

Iteration 2*. One computes*

$$\mathcal{T} = \begin{bmatrix} 16 & 19 & \mathbf{2} & 18 & 16 & 14 & 19 & 18 & 14 & 19 & 19 & \mathbf{2} \\ 6 & 5 & \boxed{22} & 8 & 6 & 2 & 5 & 8 & 10 & 5 & 5 & 2 \\ 4 & 3 & 6 & 2 & 4 & 6 & 3 & 2 & 6 & 3 & 3 & 6 \\ 6 & 5 & 2 & 4 & 6 & 10 & 5 & 4 & 2 & 5 & 5 & \boxed{22} \end{bmatrix},$$

$$\nu = 2, \quad \mathcal{P}_\nu = \{2, 11\}, \quad \mathcal{M}_2 = \mathcal{M}_{11} = 22, \quad \sigma_{\mu_2} = 1 \text{ and } \sigma_{\mu_{11}} = a^2.$$

Note that the minimum value of the first row of $\mathcal{T}$ occurs several times in the row. Therefore, the algorithm decides that most probably there are errors in $y_1(x)$ at positions 2 and 11, and most probably the corresponding entries in the error vector are $\sigma_1 = 1$ and $\sigma_3 = a^2$. Hence we put

$$y_2(x) = y_1(x) - (a^2x^{11} + x^2) = ax^{11} + x^{10} + ax^9 + ax^8 + x^7 + a^2x^5 + a^2x^3 + a^2x^2 + 1.$$

One verifies that $y_2(x)$ is an element of the code $\mathcal{C}$. Thus, the decoding is successfully done and we have just managed to find out that $y(x) = c(x) + e(x)$, where

$$c(x) = ax^{11} + x^{10} + ax^9 + ax^8 + x^7 + a^2x^5 + a^2x^3 + a^2x^2 + 1 \in \mathcal{C},$$
$$e(x) = a^2x^{11} + ax^7 + x^2.$$

Let us also notice that there were only three errors, which coincides with the error capacity of the code $\mathcal{C}$, and the recovered message $c(x)$ is unique.

4 Plausibility analysis of Algorithm 1

A plausibility analysis of an iterative decoding algorithm for binary cyclic codes was given in [2]. It was improved and completed for non-binary cyclic codes in [9]. This analysis (left column of Page 655) can be adapted to our situation.

In what follows we present a conjecture (Conjecture 1) on the failure of the algorithm. This conjecture is motivated by Example 4 and proved when the error weight is one (Lemma 3).

Let us first introduce a new set. Given $f(x) = 1 + \lambda_{\beta_1}x^{\beta_1} + \ldots + \lambda_{\beta_{d^\perp-1}}x^{\beta_{d^\perp-1}}$ in R, the support S_f^0 of $f(x)$ is

$$S_f^0 := \{0, \beta_1, \ldots, \beta_{d^\perp-1}\},$$

and for $0 < i < d^\perp$ we denote by S_f^i the support of $f(x)\varepsilon^{-1}\theta^{-\beta_i}\left(\lambda_{\beta_i}^{-1}\right)x^{n-\beta_i} \bmod (x^n - \varepsilon)$:

$$S_f^i := \{(v - \beta_i) \bmod n,\ v \in S_f^0\}.$$

Consider the intersection $\mathcal{I}$ of the supports S_f^i:

$$\mathcal{I} = \bigcap_{i=0, f\in\overline{\mathcal{B}}_{d^\perp}}^{d^\perp-1} S_f^i.$$

We have the following conjecture.

Conjecture 1. If $\mathcal{I} \neq \{0\}$, then the set $\mathcal{P}_\nu$ constructed at the first stage of the iterative decoding Algorithm 1 contains non-erroneous positions, and, therefore, the algorithm returns Failure.

This means that the decoding can not be done with the dual codewords of weight $d^\perp$. The following lemma gives a proof of Conjecture 1 when the error weight is equal to 1.

Lemma 3. *Conjecture 1 is true if the error weight is equal to* 1.

PROOF. Assume that $e(x) = e_{\gamma_1} x^{\gamma_1}$. Each $\omega_f^i \in \mathcal{L}$ is of the form

$$\omega_f^i(x) = e_{\gamma_1} x^{\gamma_1} + e_{\gamma_1} \theta^{\gamma_1}(\lambda_{\beta_1}) x^{\gamma_1+\beta_1} + \cdots + e_{\gamma_1} \theta^{\gamma_1}(\lambda_{\beta_{d^\perp-1}}) x^{\gamma_1+\beta_{d^\perp-1}} \bmod x^n - \varepsilon$$

for all $f(x) \in \overline{\mathcal{B}}_{d^\perp}$ and $i \in \{0, \ldots, d^\perp - 1\}$. Therefore, at the position γ_1 of the vector ω_f^i, we have the error magnitude e_{γ_1}. According to (5), $\nu = \mathcal{T}_{\sigma_0,\gamma_1}$ is zero. Algorithm 1 succeeds in this case if and only if $\mathcal{P}_\nu = \{\gamma_1\}$. This is equivalent to $\mathcal{I} = \{0\}$. Namely, one has the following equivalences:

$$\begin{aligned}
\mathcal{P}_\nu \neq \{\gamma_1\} &\Leftrightarrow \exists \beta \in \{1, \ldots, n-1\},\ (\beta + \gamma_1) \bmod n \in \mathcal{P}_\nu \\
&\Leftrightarrow \exists \beta \in \{1, \ldots, n-1\}, \forall i, f,\ (\beta + \gamma_1) \bmod n \in Supp(\omega_f^i(x)) \\
&\Leftrightarrow \exists \beta \in \{1, \ldots, n-1\}, \forall i, f,\ \beta \in S_f^i \\
&\Leftrightarrow \{1, \ldots, n-1\} \cap \mathcal{I} \neq \emptyset \\
&\Leftrightarrow \mathcal{I} \neq \{0\}.
\end{aligned}$$

■

The following example illustrates Conjecture 1 in the case when Algorithm 1 fails for sample of errors of weight bigger than 1.

Example 4 *Consider the* $[54, 19, 21]_9$ *θ-cyclic code* $\mathcal{C} = (g)_{54,\theta}$ *defined over* $\mathbb{F}_9 = \mathbb{F}_3(a)$, *where* $a^2 = a+1$, *and* $g(x) = a^2x^{35} + a^7x^{34} + x^{33} + a^7x^{32} + a^3x^{31} + 2x^{30} + a^2x^{28} + a^7x^{27} + a^7x^{26} + a^2x^{24} + ax^{23} + a^5x^{22} + a^3x^{21} + x^{20} + a^2x^{19} + a^2x^{18} + ax^{17} + a^5x^{15} + ax^{12} + a^3x^{11} + x^{10} + a^7x^9 + a^3x^7 + x^6 + a^2x^5 + a^6x^4 + a^2x^2 + ax + 1$.

The skew check polynomial of $\mathcal{C}$ *is* $h(x) = a^2x^{19} + ax^{18} + a^5x^{17} + a^5x^{15} + a^3x^{14} + a^7x^{13} + ax^{12} + a^7x^{11} + 2x^{10} + a^3x^9 + x^8 + ax^7 + ax^6 + a^2x^5 + a^2x^4 + a^5x^3 + a^7x^2 + ax + 2$.

The dual code of $\mathcal{C}$ *is a* $[54, 35, 6]_9$ *θ-cyclic code with skew generator polynomial* $h^*(x)$. *We have* $x^{54} - 1 = (x^2 - 1)^{27} = g(x)h(x) = h(x)g(x)$ *and one can check that* $h^*(x)$ *divides* $(x^2 - 1)^{18}(x + 1) = (x^{36} + x^{18} + 1)(x + 1)$.

The set $\overline{\mathcal{B}}_6$ *is obtained by considering all the multiples of* $(x^{36}+x^{18}+1)(x+1)$ *of weight* 6*:*

$$\begin{aligned}
\overline{\mathcal{B}}_6 = \{(x^{36} + x^{18} + 1)u \mid u \in \{&x + 1, 2x^2 + 1, x^3 + 1, 2x^4 + 1, \\
&2x^6 + 1, x^7 + 1, 2x^8 + 1, x^9 + 1\}\}.
\end{aligned}$$

The set $\mathcal{I}$ *is therefore equal to* $\{0, 18, 36\}$. *We considered a few of hundreds of thousands of samples, and we found no error vectors of weights up to* 14, *which Algorithm* 1 *can correct.*

For this reason, we opted to work with $\overline{\mathcal{B}}_{13}$ *(cf. the table below).*

In order to overcome the fact that $\mathcal{I}$ may be distinct from $\{0\}$, one chooses to replace $\overline{\mathcal{B}}_{d^\perp}$ with $\overline{\mathcal{B}}_w$, $w \geq d^\perp$, in the entry of Algorithm 1. We have implemented the algorithm in C and present some preliminary computational results in the table bellow.

Table 1. Results of computer simulations of Algorithm 1.

Codes	$[54, 27, 18]_9$		$[54, 19, 21]_9$		$[62, 26, 19]_4$	
Duals	$[54, 27, 18]_9$		$[54, 35, 6]_9$		$[62, 36, 13]_4$	
$d^{\perp}, w$	$18, 18$		$6, 13$		$13, 13$	
Success rate	$\tau \leq 7$	1	$\tau \leq 12$	1	$\tau \leq 7$	1
	$\tau = 8$	$0,9923$	$\tau = 13$	$0,9999$	$\tau = 8$	$0,9878$
	$\tau = 9$	$0,7532$	$\tau = 14$	$0,9918$	$\tau = 9$	$0,8346$

5 Conclusion

In this text we provide a generalization of the iterative decoding of [2] to the class of skew constacyclic codes that are ideal codes. We have initiated a preliminary analysis of our algorithm, and we aim at providing a more accurate analysis of its success rate. We implemented an improved version of Algorithm 1 both in *Magma* and in *C*, and conducted multiple experiments on several skew constacyclic codes over small finite fields. In the sequel, we would like also to compare our algorithm to other decoding algorithms (designed for skew BCH codes, for example).

References

1. W. Bosma, J. Cannon, C. Playoust: The Magma algebra system. I. The user language. Journal of Symbolic Computation, **24**, 3-4, 235–265, 1997 .
2. M. Bossert: On decoding using codewords of the dual code. arXiv preprint, 2001.02956, Jan. 2020.
3. J. Xing, M. Bossert, S. Bitzer, L. Chen: Iterative decoding of non-binary cyclic codes using minimum-weight dual codewords. in Proc. IEEE Int. Symp. Inform. Theory (ISIT), CA, U.S.A, pp. 333-337, June 2020.
4. D. Boucher, W. Geiselmann, F. Ulmer: Skew-cyclic codes. Appl. Algebra Engin. Commun. Comp., **18**, 379–389, 2007.
5. D. Boucher, F. Ulmer: Coding with skew polynomial rings. J. Symb. Comp., **44**, 1644–1656, 2009.
6. D. Boucher, F. Ulmer: A note on the dual codes of module skew codes. Liqun Chen. Cryptography and coding: 13th IMA international conference, IMACC 2011, Oxford, UK, December 12-15, 2011.
7. D. Boucher, F. Ulmer: Self-dual skew codes and factorization of skew polynomials. J. Symb. Comp., **60**, 47–61, 2014.
8. L. Chaussade, P. Loidreau, F. Ulmer: Skew codes of prescribed distance or rank. Designs, Codes and Cryptography, 50 (3), pp.267-284, 2009.
9. L. Chen, J. Xing, J. Yuan: Plausibility analysis of Shift-Sum decoding for cyclic codes 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, pp. 652-657, 2021.
10. J. Gómez-Torrecillas, F. J. Lobillo, G. Navarro: Peterson-Gorenstein-Zierler algorithm for skew RS codes. Linear and Multilinear Algebra, 66, 3, 469–487, 2018.
11. O. Ore: Theory of Non-commutative Polynomials, The annals of Mathematics, 2nd Ser, 34(3), 480-508, 1933.