entropy

Article

Skew Convolutional Codes

Vladimir Sidorenko [1,*], Wenhui Li [2], Onur Günlü [3] and Gerhard Kramer [1]

[1] Institute for Communications Engineering, Technical University of Munich, 80333 München, Germany; gerhard.kramer@tum.de

[2] Skolkovo Institute of Science and Technology, 143026 Moscow, Russia; w.li@skoltech.ru

[3] Information Theory and Applications Chair, Technical University of Berlin, 10623 Berlin, Germany; guenlue@tu-berlin.de

* Correspondence: vladimir.sidorenko@tum.de

Received: 30 October 2020; Accepted: 30 November 2020; Published: 2 December 2020

Abstract: A new class of convolutional codes, called skew convolutional codes, that extends the class of classical fixed convolutional codes, is proposed. Skew convolutional codes can be represented as periodic time-varying convolutional codes but have a description as compact as fixed convolutional codes. Designs of generator and parity check matrices, encoders, and code trellises for skew convolutional codes and their duals are shown. For memoryless channels, one can apply Viterbi or BCJR decoding algorithms, or a dualized BCJR algorithm, to decode skew convolutional codes.

Keywords: convolutional codes; skew polynomials; time-varying codes; dual codes; trellises

1. Introduction

Convolutional codes were introduced by Elias in 1955 [1]. With the discovery that convolutional codes can be decoded with Fano sequential decoding [2], Massey threshold decoding [3], and, above all, Viterbi decoding [4], they became quite widespread in practice. Convolutional codes are still widely used in telecommunications, e.g., in Turbo codes [5] and in the WiFi IEEE 802.11 standard [6], in cryptography [7], etc.

The most common are binary convolutional codes; however, communication with higher orders of modulation [8] or streaming of data [9] require *non-binary convolutional codes*. It is known that *periodic time-varying convolutional codes improve the free distance and weight distribution over fixed convolutional codes*; see, e.g., Mooser [10] and Lee [11]. This is the motivation to introduce a new class of periodic time-varying non-binary convolutional codes, i.e., skew convolutional codes. These codes are based on the non-commutative ring of skew polynomials over finite fields and on the skew field of their fractions.

Block codes based on skew polynomials were studied by various authors; see, e.g., publications of Gabidulin [12], Boucher and Ulmer [13,14], Martínez-Peñas [15], Gluesing-Luerssen [16], Abualrub, Ghrayeb, Aydin, and Siap [17].

Convolutional codes are nonblock linear codes over a finite field, but it can be advantageous to treat them as block codes over certain infinite fields. We will use both approaches. Classical convolutional codes are described by usual polynomials. The product of the polynomials corresponds to convolution of vectors of their coefficients and this gives fixed-in-time classical convolutional codes. We replace usual polynomials by skew polynomials to define the new codes. The product of skew polynomials corresponds to skew convolution of their coefficients, which can be obtained by varying elements in the usual convolution. In this way, we obtain time varying convolutional codes.

Our goal is to define and to give a first encounter with skew convolutional codes. In Section 2, we define skew convolutional codes. In Section 3, we obtain generator matrices and encoders for skew codes and show that skew codes are equivalent to time-varying convolutional codes. Some useful properties of skew codes are considered in Section 4. Section 5 introduces dual skew convolutional codes. Trellis decoding of skew codes is considered in Section 6. Section 7 concludes the paper.

2. Skew Convolutional Codes

2.1. Skew Polynomials and Fractions

Consider a field $\mathbb{F}$ and an automorphism θ of the field. Later on, we will use the finite field $\mathbb{F} = \mathbb{F}_{q^m}$, where q is a prime power, with the *Frobenius automorphism*

$$\theta(a) = a^q, \qquad \forall a \in \mathbb{F}. \tag{1}$$

The composition of automorphisms is denoted as $\theta(\theta(a)) = \theta^2(a)$, and, for any integer i, we have $\theta^i(a) = \theta(\theta^{i-1}(a))$. The identity automorphism $\theta(a) = a$ is denoted as $\theta = \text{id}$. For the automorphism (1) for all $a \in \mathbb{F}$, we have $\theta^i(a) = a^{q^i}$ and $\theta^m = \text{id}$ since $a^{q^m} = a$.

Denote by $\mathcal{R} = \mathbb{F}[D;\theta]$ the non- commutative ring [18] of *skew polynomials* in the variable D over $\mathbb{F}$ (with zero derivation) such that

$$\mathcal{R} = \mathbb{F}[D;\theta] = \{a(D) = a_0 + a_1 D + \cdots + a_n D^n \mid a_i \in \mathbb{F} \text{ and } n \in \mathbb{N}\}.$$

The skew polynomials look like usual polynomials $\mathbb{F}[D]$ where the coefficients a_i are placed to the right of the variable D. The addition in $\mathcal{R}$ is as for usual polynomials from $\mathbb{F}[D]$. The multiplication is defined by the basic rule

$$Da = \theta(a)D \tag{2}$$

and is extended to all elements of $\mathcal{R}$ by associativity and distributivity; see Example 1 below. The ring $\mathcal{R}$ has a unique left skew field of fractions $\mathcal{Q}$, from which it inherits its linear algebra properties, see, e.g., [19] for more details.

Example 1. *To demonstrate our results, we use the field $\mathbb{F}_Q = \mathbb{F}_{q^m} = \mathbb{F}_{2^2}$, $q = 2$, $m = 2$, with automorphism $\theta(a) = a^q = a^2$ for all $a \in \mathbb{F}_{2^2}$. The field $\mathbb{F}_{2^2}$ consists of the elements $\{0, 1, \alpha, \alpha^2\}$, where a primitive element α satisfies $\alpha^2 + \alpha + 1 = 0$ and the following relations hold:*

$$\begin{array}{l} \alpha^2 = \alpha + 1, \\ \alpha^3 = 1, \\ \alpha^4 = \alpha, \end{array} \qquad \textit{and } \forall i \in \mathbb{Z} \quad \theta^i = \begin{cases} \theta & \textit{if } i \textit{ is odd,} \\ id & \textit{if } i \textit{ is even.} \end{cases}$$

Let $a(D) = 1 + \alpha D$ and $b(D) = \alpha^2 + D$, $a(D), b(D) \in \mathcal{R}$. Using (2), we compute the product ab as

$$a(D)b(D) = (1 + \alpha D)(\alpha^2 + D) = (\alpha^2 + D) + \alpha\theta(\alpha^2)D + \alpha D^2 = \alpha^2 + \alpha D + \alpha D^2$$

while the product ba is

$$b(D)a(D) = (\alpha^2 + D)(1 + \alpha D) = (\alpha^2 + D) + \alpha^3 D + \theta(\alpha)D^2 = \alpha^2 + \alpha^2 D^2.$$

In this example, we see that $a(D)b(D) \neq b(D)a(D)$.

The left skew field $\mathcal{Q}$ consists of fractions $\frac{b(D)}{a(D)} = a^{-1}(D)b(D) \in \mathcal{Q}$ for all $a(D), b(D) \in \mathcal{R}$, $a(D) \neq 0$. Every fraction can be expanded as the left skew Laurent series in increasing powers of D. In our example, the inverse element $\frac{1}{a(D)} = a(D)^{-1}$ is expanded using long devision as follows:

$$a^{-1}(D) = 1 + \alpha D + D^2 + \alpha D^3 + D^4 + \ldots$$

with $a^{-1}(D)a(D) = a(D)a^{-1}(D) = 1$. We can expand the left fraction $\frac{b(D)}{a(D)} = a^{-1}(D)b(D)$ by long left division or equivalently by left multiplication $b(D)$ by $a^{-1}(D)$ and get

$$\frac{b(D)}{a(D)} = a^{-1}(D)b(D) = \alpha^2 + \alpha D + D^2 + \alpha D^3 + D^4 + \dots .$$

Notice that the right fraction $b(D)a^{-1}(D) = \alpha^2$, since $b(D) = \alpha^2 + D = \alpha^2(1 + \alpha D) = \alpha^2 a(D)$.

2.2. Definition of Skew Convolutional Codes

Much of linear algebra can be generalized from vector spaces over a field to (either left or right) modules over the skew field $\mathcal{Q}$. Indeed, it is shown in [19] that any left $\mathcal{Q}$-module $\mathcal{C}$ is free, i.e., it has a basis, and any two bases of $\mathcal{C}$ have the same cardinality, the dimension of $\mathcal{C}$.

Definition 1 (Skew convolutional code)**.** *Given an automorphism θ of the field $\mathbb{F}$, a skew convolutional $[n,k]$ code $\mathcal{C}$ over the field $\mathbb{F}$ is a left sub-module of dimension k of the free module $\mathcal{Q}^n$.*

The elements of the code $\mathcal{C}$ are called its *codewords*. A codeword $v(D) = \left(v^{(1)}(D), \dots, v^{(n)}(D)\right)$ is an n-tuple over $\mathcal{Q}$, where every component $v^{(i)}(D)$ is a fraction of skew polynomials from $\mathcal{R}$. The code $\mathcal{C}$ is $\mathbb{F} = \mathbb{F}_{q^m}$-linear. Let the weight of every codeword be defined by some selected metric. The *free distance d_f* of a skew convolutional code is defined to be the minimum nonzero weight of any codeword.

For the Hamming metric, which is the most interesting for applications, the weight of a fraction $v^{(i)}(D)$ is the number of nonzero coefficients in its expansion as a left skew Laurent series from $\mathbb{F}((D))$ in increasing powers of D. The weight of a codeword is the sum of weights of its components. Another interesting metric is sum-rank metric, which will be defined later.

2.3. Relations to Fixed Convolutional Codes

Lemma 1. *The class of skew convolutional codes includes the class of fixed (time-invariant) convolutional codes.*

Proof. A time-invariant convolutional $[n,k]$ code $\widetilde{\mathcal{C}}$ over the field $\mathbb{F}$ is defined as a k-dimensional subspace of $\mathbb{F}(D)^n$. Take the identity automorphism $\theta = \text{id}$. Then, the ring $\mathcal{R} = \mathbb{F}[D;\theta]$ becomes a ring $\mathbb{F}[D]$ of usual commutative polynomials. The skew field of fractions $\mathcal{Q}$ becomes the field of rational functions $\mathbb{F}(D)$. In this case, by Definition 1, the skew convolutional code $\mathcal{C}$ coincides with the classical fixed code $\widetilde{\mathcal{C}}$. □

3. Encoding Skew Convolutional Codes

3.1. Polynomial Form of Encoding

A *generator matrix* of a skew convolutional $[n,k]$ code $\mathcal{C}$ is a $k \times n$ matrix $G(D)$ over the skew field $\mathcal{Q}$ whose rows form a basis for the code $\mathcal{C}$. If the matrix $G(D)$ is over the ring $\mathcal{R}$ of skew polynomials, then $G(D)$ is called a *polynomial generator matrix* for $\mathcal{C}$. Every skew code $\mathcal{C}$ has a polynomial generator matrix. Indeed, given a generator matrix $G(D)$ over the skew field of fractions $\mathcal{Q}$, a polynomial generator matrix can be obtained by left multiplying each row of $G(D)$ by the least common left multiple of the denominators in that row. In the paper, we focus on polynomial generator matrices and corresponding encoders.

By Definition 1, every codeword $v(D)$ of a skew code $\mathcal{C}$, which is an n-tuple over the skew field of fractions $\mathcal{Q}$

$$v(D) = \left(v^{(1)}(D),\, v^{(2)}(D), \dots,\, v^{(n)}(D)\right), \quad v^{(j)}(D) \in \mathcal{Q}, \quad 1 \le j \le n \tag{3}$$

can be written as

$$v(D) = u(D)G(D) \tag{4}$$

where $u(D)$ is a k-tuple (k-word) over $\mathcal{Q}$:

$$u(D) = \left(u^{(1)}(D),\, u^{(2)}(D) \ldots,\, u^{(k)}(D)\right), \quad u^{(i)}(D) \in \mathcal{Q}, \quad 1 \le i \le k \tag{5}$$

and is called an information word, $G(D)$ is a generator matrix of $\mathcal{C}$, $G(D) \in \mathcal{Q}^{k\times n}$ or $G(D) \in \mathcal{R}^{k\times n}$. Relation (4) already provides an encoder. This encoder (4) is just an encoder of a block code over $\mathcal{Q}$ and the skew code $\mathcal{C}$ can be considered as the set of n-tuples $v(D)$ over $\mathcal{Q}$ that satisfy (4), i.e., $\mathcal{C} = \{v(D)\}$.

However, to use the code in practice we need to write components of $u(D)$ and $v(D)$ as skew Laurent series, i.e., we have

$$u^{(i)}(D) = u_0^{(i)} + u_1^{(i)} D + u_2^{(i)} D^2 + \ldots, \quad i = 1, \ldots, k \tag{6}$$

and

$$v^{(j)}(D) = v_0^{(j)} + v_1^{(j)} D + u_2^{(j)} D^2 + \ldots, \quad j = 1, \ldots, n. \tag{7}$$

Actually, in a Laurent series, the lower (time) index of coefficients can be a negative integer, but, in practice, the information sequence $u^{(i)}(D)$ should be causal for every component i, that is, the coefficients $u_t^{(i)}$ are zeros for time $t < 0$. Causal information sequences should be encoded into causal code sequences; otherwise, an encoder can not be implemented, since it should output code symbols before it receives an information symbol.

Denote the block of information symbols that enters an encoder at time $t = 0, 1, \ldots$ by

$$u_t = \left(u_t^{(1)}, u_t^{(2)}, \ldots u_t^{(k)}\right) \in \mathbb{F}^k. \tag{8}$$

The block of code symbols that leaves the encoder at time $t = 0, 1, \ldots$ is denoted by

$$v_t = \left(v_t^{(1)}, v_t^{(2)}, \ldots v_t^{(n)}\right) \in \mathbb{F}^n. \tag{9}$$

Combining (5), (6), and (8), we obtain the following information series with vector coefficients:

$$u(D) = u_0 + u_1 D + \ldots + u_t D^t + \ldots, \quad u(D) \in \mathbb{F}((D))^k. \tag{10}$$

Using (3), (7), and (9), we write a codeword as a series

$$v(D) = v_0 + v_1 D + \ldots + v_t D^t + \ldots, \quad v(D) \in \mathbb{F}((D))^n. \tag{11}$$

One can write a skew polynomial generator matrix $G(D) = \left(g_{ij}(D)\right) \in \mathcal{R}^{k\times n}$ as a skew polynomial with matrix coefficients:

$$G(D) = G_0 + G_1 D + G_2 D^2 + \ldots + G_\mu D^\mu \tag{12}$$

where μ is the maximum degree of polynomials $g_{ij}(D)$. The matrices G_i are $k \times n$ matrices over the field $\mathbb{F}$ and μ is called the generator matrix *memory*.

From (4), (10), and (11), we obtain that v_t is a coefficient in the product of skew series $u(D)$ and skew polynomial $G(D)$, which is the following *skew convolution* (see Figure 1)

$$v_t = u_t \theta^t(G_0) + u_{t-1}\theta^{t-1}(G_1) + \cdots + u_{t-\mu}\theta^{t-\mu}(G_\mu) \tag{13}$$

where $u_t = 0$ for $t < 0$. This encoding rule explains the title *skew convolutional code*, which can be also seen as the set $\mathcal{C} = \{v(D)\}$ of series $v(D)$ defined in (11).

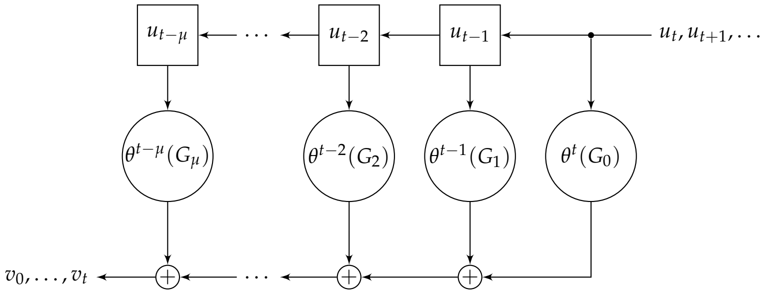

Figure 1. Encoder of a skew convolutional code.

At time t, the decoder observes an information block u_t of k symbols from $\mathbb{F}$ and outputs the code block v_t of n code symbols from $\mathbb{F}$ using (13); hence, the *code rate* is $R = k/n$. The encoder (13) uses u_t and also μ previous information blocks $u_{t-1}, u_{t-2}, \dots, u_{t-\mu}$, which should be stored in the encoder's memory; this is why μ is also the encoder *memory*.

The coefficients $\theta^{t-i}(G_i)$, $i = 0, 1, \dots, \mu$ in the encoder (13) depend on the time t. Hence, the *skew convolutional code is a time-varying classical convolutional code*. Denote

$$\tau = \min \left\{ i > 0 \; : \; \theta^i(G_j) = G_j \;\; \forall j = 0, 1, \dots, \mu \right\}. \tag{14}$$

For the field $\mathbb{F} = \mathbb{F}_{q^m}$, we have $\theta^m = \theta^0$; hence, the coefficients in (13) are periodic with period $\tau \le m$, and the *skew convolutional code is periodic with period* $\tau \le m$. The period τ can be less than m if all the matrices $G_0, \dots G_\mu$ are over a subfield of $\mathbb{F}_{q^m}$.

3.2. Scalar Form of Encoding

The input of the encoder can also be written as an information sequence of k-blocks over $\mathbb{F}$

$$u = u_0,\ u_1,\ u_2, \dots, u_t, \dots \tag{15}$$

and the output as a code sequence of n-blocks over $\mathbb{F}$

$$v = v_0,\ v_1,\ v_2, \dots,\ v_t,\ \dots\ . \tag{16}$$

Then, the encoding rule (13) can be written in a scalar form

$$v = uG \tag{17}$$

with semi-infinite scalar generator block matrix

$$G = \begin{pmatrix} G_0 & G_1 & G_2 & \dots & G_\mu & & \\ & \theta(G_0) & \theta(G_1) & \dots & \theta(G_{\mu-1}) & \theta(G_\mu) & \\ & & \theta^2(G_0) & \dots & \theta^2(G_{\mu-2}) & \theta^2(G_{\mu-1}) & \theta^2(G_\mu) \\ & & & \dots & & & \end{pmatrix}. \tag{18}$$

Thus, a skew convolutional code can be equivalently represented in a scalar form as the set $\mathcal{C} = \{v\}$ of sequences v defined in (16) that satisfy (17).

3.3. Relations between Skew and Classical Convolutional Codes

In case of identity automorphism, $\theta(a) = a$, the scalar generator matrix of the skew code becomes

$$G' = \begin{pmatrix} G_0 & G_1 & G_2 & \dots & G_\mu & & \\ & G_0 & G_1 & \dots & G_{\mu-1} & G_\mu & \\ & & G_0 & \dots & G_{\mu-2} & G_{\mu-1} & G_\mu \\ & & & \dots & & & \end{pmatrix}, \tag{19}$$

which is a generator matrix of a fixed convolutional code [20]. For fixed convolutional codes, polynomial generator matrices with G_0 of full rank k are of particular interest [20] (Chapter 3). The skew convolutional codes use the following nice property: if G_0 has full rank, then $\theta^i(G_0)$ has full rank as well for all i.

Time-varying classical convolutional codes are defined by the following generator matrices in [20],

$$G_{var} = \begin{pmatrix} G_{0,0} & G_{1,1} & & G_{\mu,\mu} & & \\ & G_{1,0} & \vdots & G_{\mu,\mu-1} & G_{\mu+1,\mu} & \\ & & & \vdots & G_{\mu+1,\mu-1} & \vdots \\ & & & G_{\mu,0} & \vdots & \\ & & & & G_{\mu+1,0} & \end{pmatrix} \tag{20}$$

where the first index t in $G_{t,i}$ is time index. The code defined by the generator matrix G_{var} in (20) is called τ-*periodic* if the columns $(G_{t,\mu}^T, \dots, G_{t,0}^T)^T$, $t \geq \mu$, repeat with period τ.

Lemma 2. *A scalar generator matrix (18) of a skew code can be written in the following equivalent form:*

$$G = \begin{pmatrix} \widetilde{G}_0 & \theta(\widetilde{G}_1) & & \theta^\mu(\widetilde{G}_\mu) & & \\ & \theta(\widetilde{G}_0) & \vdots & \theta^\mu(\widetilde{G}_{\mu-1}) & \theta^{\mu+1}(\widetilde{G}_\mu) & \\ & & & \vdots & \theta^{\mu+1}(\widetilde{G}_{\mu-1}) & \vdots \\ & & & \theta^\mu(\widetilde{G}_0) & \vdots & \\ & & & & \theta^{\mu+1}(\widetilde{G}_0) & \end{pmatrix}. \tag{21}$$

Proof. The statement follows from the change of variables $G_i = \theta^i(\widetilde{G}_i)$ for $i = 1, 2, \dots, \mu$. □

From (14), (20) and (21), we see again that a skew code defined by a generator matrix (21) is a τ-periodic classical convolutional code. Thus, above we proved the following theorem.

Theorem 1. *Given a field $\mathbb{F} = \mathbb{F}_{q^m}$ with automorphism θ in (1), any skew convolutional $[n, k]$ code $\mathcal{C}$ over $\mathbb{F}$ is equivalent to a periodic time-varying (classical) convolutional $[n, k]$ code over $\mathbb{F}$, with period $\tau \leq m$ (14). If $G(D)$ is a skew polynomial generator matrix (12) of the code $\mathcal{C}$, then the scalar generator matrix G of the time-varying code is given by (18) or (21).*

Not every periodic classical convolutional code can be represented as a skew code. Indeed, e.g., the submatrix $G_{1,0}$ in (20) can be selected independently of $G_{0,0}$ while corresponding submatrix $\theta(\widetilde{G}_0)$ in (21) is completely determined by $\widetilde{G}_0$. Hence, a class of skew convolutional codes is a subclass of periodic classical convolutional codes.

Given the field $\mathbb{F}_{q^m}$, the automorphism θ in (1), and the code memory μ, an $[n, k]$ skew convolutional code is defined by a generator matrix G in (21). To specify the matrix, we should fix elements of $\mu + 1$ matrices $\widetilde{G}_0, \dots, \widetilde{G}_\mu$ of size $k \times n$. Hence, we should define $(\mu + 1)kn$ field elements. Since a classical convolutional code corresponds to the identity automorphism $\theta = \mathrm{id}$, the *description of skew and classical codes require the same number of field elements*.

The number of skew $[n,k]$ convolutional codes over $\mathbb{F}_Q = \mathbb{F}_{q^m}$ of memory μ with fixed automorphism $\theta(a) = a^q$ has order $q^{(\mu+1)mkn}$. The number of τ-periodic classical convolutional codes has order $q^{(\mu+1)mkn\tau}$, which is much larger. As a result, the search of good periodic time-varying convolutional codes is much simpler in the class of skew codes in comparison with periodic classical codes. The search among skew convolutional codes has the same complexity as the search among fixed classical codes.

How many more skew codes can we obtain by considering all possible automorphisms? Denote $q = p^s$, where p is the field characteristic then $\mathbb{F}_{q^m} = \mathbb{F}_{p^{sm}} = \mathbb{F}_{p^M}$, i.e., our field $\mathbb{F}_Q$ is an M-extension of the prime field $\mathbb{F}_p$. The parameter $q = p^s$, we should select such that $\mathbb{F}_q$ is a subfield of $\mathbb{F}_{p^M}$, hence s should divide M. Denote by $\delta(M)$ the *number-of-divisors function* that is the number of divisors i of M, $1 \le i \le M$. Given the field $\mathbb{F}_Q = \mathbb{F}_{p^M}$, we can select $q = p^i$ and the automorphism $\theta(a) = a^q$ in $\delta(M)$ ways.

Lemma 3. *For a fixed field $\mathbb{F}_Q = \mathbb{F}_{p^M}$, there are $\delta(M)$ sub-classes of skew convolutional codes, each defined by a fixed automorphism θ.*

In Example 2, we have $\mathbb{F}_Q = \mathbb{F}_{2^2}$, i.e., $p = q = 2, s = 1, m = 2$. $M = sm = 2$ with divisors 1 and 2. For $i = 1$, we have $q = p^i = 2$ and $\theta(a) = a^2$ considered in Example 2. For $i = 2$, we have $q = p^2 = 4$ that corresponds to $\theta = \mathrm{id}$ and gives a constant classical convolutional code. For the field $\mathbb{F}_2^6$, we have $\delta(6) = 4$, and there are four sub-classes of skew codes with $q = 2^1, 2^2, 2^3$, and $q = 2^6$ (for fixed code).

4. Properties of Skew Convolutional Codes

4.1. Extension of Fixed Convolutional Codes

To show properties of skew convolutional codes, we will use the following example.

Example 2. *Consider a [2,1] skew convolutional code $\widehat{\mathcal{C}}$ over the field $\mathbb{F}_Q = \mathbb{F}_{q^m} = \mathbb{F}_{2^2}$ with automorphism $\theta(a) = a^q = a^2$ (see Example 1). Let the generator matrix of the code $\widehat{\mathcal{C}}$ in polynomial form be*

$$G(D) = (1 + \alpha D,\ \alpha + \alpha^2 D) = G_0 + G_1 D, \tag{22}$$

where

$$G_0 = (1, \alpha),\ G_1 = (\alpha, \alpha^2). \tag{23}$$

The generator matrix in scalar form (18) is

$$G = \begin{pmatrix} 1\ \alpha & \alpha\ \alpha^2 & & & \\ & 1\ \alpha^2 & \alpha^2\ \alpha & & \\ & & 1\ \alpha & \alpha\ \alpha^2 & \\ & & & 1\ \alpha^2 & \alpha^2\ \alpha \\ & & \dots & & \end{pmatrix}. \tag{24}$$

Here, $\mu = 1$; hence, it is a unit memory code. The encoding rule is $v = uG$, or, from (13), it is

$$v_t = u_t \theta^t(G_0) + u_{t-1}\theta^{t-1}(G_1), \qquad t = 0, 1, \dots . \tag{25}$$

In some applications, it is preferable to have a generator matrix in systematic form. For our example, a systematic fractional matrix can be obtained using the left division of its components by the first component

$$G_{syst}(D) = \left(1, \frac{\alpha + \alpha^2 D}{1 + \alpha D}\right) = \Big(1, (1 + \alpha D)^{-1}(\alpha + \alpha^2 D)\Big). \tag{26}$$

Let us show that the matrices $G_{syst}(D)$ and $G(D)$ in (22) encode the same code $\widehat{\mathcal{C}}$. Denote $g_1(D) = 1 + \alpha D$. Then, for any information sequence $u(D) \in \mathcal{Q}$, we have the code word

$$u(D)G(D) = u(D)g_1(D)g_1^{-1}(D)G(D) = u(D)g_1(D)G_{syst}(D) = u'(D)G_{syst}(D)$$

and the statement follows since there is a one to one mapping between $u(D)$ and $u'(D) = u(D)g_1(D)$; hence, both information sequences $u(D)$ and $u'(D)$ run over all possible causal sequences.

Theorem 2. *The class of skew convolutional codes extends the class of fixed convolutional codes.*

Proof. By Lemma 1, the class of fixed codes is included in the class of skew convolutional codes. Hence, it is sufficient to show that there exists a codeword in a skew convolutional $[n, k]$ code that can not belong to any fixed $[n, k]$ code with the same memory. Indeed, consider the unit memory, $\mu = 1$, skew $[2, 1]$ code $\widehat{\mathcal{C}}$ defined by the generator matrix (24). By encoding the information sequence $u = 1, 0, 0, 1$, we obtain the codeword $v = v_0,\ v_1,\ v_2,\ v_3,\ v_4 = (1, \alpha), (\alpha, \alpha^2), (0, 0), (1, \alpha^2), (\alpha^2, \alpha) \in \widehat{\mathcal{C}}$.

Suppose for the sake of contradiction that the codeword v belongs to a fixed unit memory $[2, 1]$ convolutional code $\mathcal{C}'$. A general form of generator matrix of a unit memory fixed $[2, 1]$ code $\mathcal{C}'$ is (19):

$$G' = \begin{pmatrix} a\,b & c\,d & & & \\ & a\,b & c\,d & & \\ & & a\,b & c\,d & \\ & & & a\,b & c\,d \\ & & \dots & & \end{pmatrix}$$

where $a, b, c, d, \in \mathbb{F}_{2^2}$. Assume that the word $v = (e, f, g, h, \dots)G' \in \mathcal{C}'$ where $e, f, g, h \in \mathbb{F}_{2^2}$. From $v_2 = f(c, d) + g(a, b) = (0, 0)$, it follows that either i) $f = g = 0$, or ii) vectors (c, d) and (a, b) are $\mathbb{F}_{2^2}$-linearly dependent. In case i), $v_0 = e(a, b) = (1, \alpha)$ and $v_3 = h(a, b) = (1, \alpha^2)$, $e^{-1}(1, \alpha) = h^{-1}(1, \alpha^2)$, which is impossible since the vectors $(1, \alpha)$ and $(1, \alpha^2)$ are linearly independent. In case ii), linear combinations of linearly dependent vectors (c, d) and (a, b) should give two linearly independent vectors $v_0 = (1, \alpha)$ and $v_3 = (1, \alpha^2)$, which is impossible as well. □

4.2. Canonical Encoders and Generator Matrices

The encoder in a controller canonical form [20] for the code $\widehat{\mathcal{C}}$ in Example 2 with generator matrix (22) is shown in Figure 2a for even t and in Figure 2b for odd t. The encoder has one shift register, since $k = 1$. There is one Q-ary memory element in the shift register shown as a rectangle, where $Q = q^m = 4$ is the order of the field. We need only one memory element since a maximum degree of items in $G(D)$, which consists of a single row in our example, is 1. A large circle means multiplication by the coefficient shown inside.

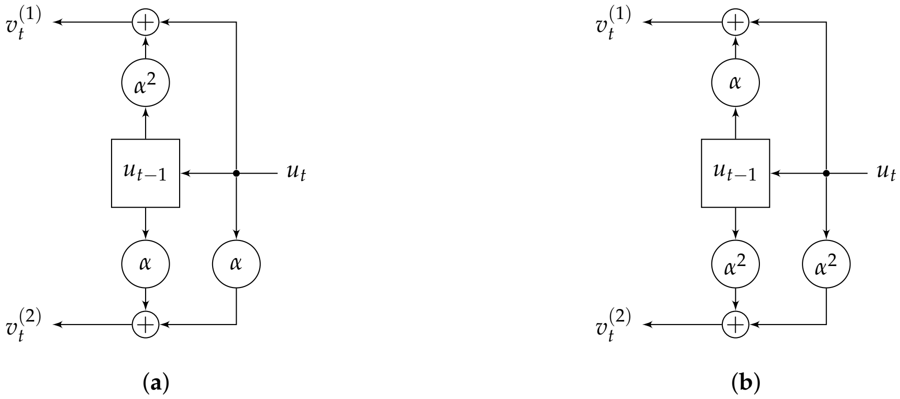

Figure 2. Encoder of the skew code $\widehat{\mathcal{C}}$ from Example 2: (**a**) for even t, (**b**) for odd t.

In the general case of a $k \times n$ matrix $G(D)$, we define the degree ν_i of its ith row as the maximum degree of its components, and *external degree* ν of $G(D)$ is the sum of its row degrees [21]. The controller-canonical-form encoder of $G(D)$ over $\mathbb{F}_Q$ has k shift registers, the ith register has ν_i memory elements, and the total number of Q-ary memory elements in the encoder is ν. Different generator matrices for a code $\mathcal{C}$ may have different external degrees.

Definition 2. *Among all skew polynomial generator matrices (PGM) for a given skew convolutional code $\mathcal{C}$, those for which the external degree is as small as possible are called canonical PGMs. This minimal external degree is called the degree or overall constraint length of the code $\mathcal{C}$, and denoted as $\nu = \deg \mathcal{C}$.*

4.3. Code Trellises

For the code $\widehat{\mathcal{C}}$ in Example 2, the code trellis is shown in Figure 3. The trellis consists of sections periodically repeated with period $\tau = m = 2$. Every section has $Q^\nu = 4^1 = 4$ states labeled by elements of the field $\mathbb{F}_Q$. For the t-th section for time $t, t = 0, 1, \ldots$, an edge connects the states u_{t-1} and u_t and is labeled by the code block v_t. Every codeword is represented by a path in the trellis that starts from the zero state 0 and goes to the right. The edge label v_t is computed according to the encoding rule (25) as follows:

$$v_t = \begin{cases} u_{t-1}(\alpha, \alpha^2) + u_t(1, \alpha^2) & \text{for odd } t, \\ u_{t-1}(\alpha^2, \alpha) + u_t(1, \alpha) & \text{for even } t \end{cases} \tag{27}$$

where we assume that $u_{-1} = 0$, i.e., the initial state of the shift register is 0.

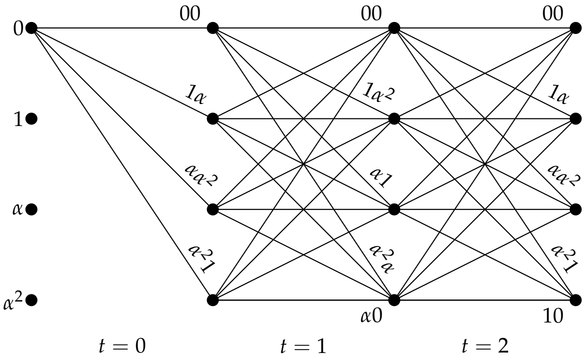

Figure 3. Time-varying trellis of the skew code $\widehat{\mathcal{C}}$.

4.4. Code Distances

There are two important characteristics of a convolutional code: the free distance d_f and the slope σ of the increase of the active burst distance, defined as follows [20]. The weight of a branch labeled by a vector v_t is defined to be the weight $w(v_t)$ of v_t. The weight of a path is the sum of its branch weights. A path in the trellis that diverges from zero state, which does not use edges of weight 0 from zero state to zero state, and that returns to zero state after ℓ edges is called a loop of length ℓ or *ℓ-loop*.

The *ℓth order active burst distance* d_ℓ^{burst} is defined [20] to be the minimum weight of ℓ-loops in the code trellis. The slope is defined as $\sigma = \lim_{\ell \to \infty} d_\ell^{\text{burst}} / \ell$. The free distance is $d_f = \min_\ell d_\ell^{\text{burst}}$.

Theorem 3 (Singleton bound). *The free Hamming distance of $[n, k]$ skew convolutional code $\mathcal{C}$ of degree $\nu = \deg \mathcal{C}$ is upper bounded as follows:*

$$d_f \leq (n-k) \left\lfloor \frac{\nu}{k} + 1 \right\rfloor + \nu + 1. \tag{28}$$

Proof. We adopted the proof given in [22] for time-invariant finite state codes. The trellis of the code $\mathcal{C}$ is time-varying with Q^ν states at every level. Consider $Q^{\ell k}$ information sequences $u_0, \ldots, u_{\ell-1}$ of length ℓ blocks. For each of them, the code path in the trellis starts at the state 0 and terminates in one of Q^{ν} states. From the pigeon-hole principle, it follows that there must be at least $Q^{\ell k-\nu} = Q^K$ of these paths that have the same final state. The code sequences corresponding to these paths can be thought of as a block code with length $N = \ell n$ with at least Q^K codewords. We should select ℓ such that $K = \ell k - \nu > 0$. The Hamming distance d of the block code is upper bounded by the Singleton bound $d \le N - K + 1 = \ell(n-k) + \nu + 1$. On the other hand, d is an upper bound on the free Hamming distance d_f of the code $\mathcal{C}$. Since this is true for all $\ell > \nu/k$, we have

$$d_f \le \min_{\ell:\, \ell > \nu/k} \ell(n-k) + \nu + 1$$

that gives the upper bound (28). □

To obtain (28), we use the Singleton bound for block codes; therefore, the bound (28) is *Singleton-type bound for skew convolutional codes*. In fact, this bound and the proof are valid for arbitrary (also for nonlinear) time-varying trellis codes. In [23], codes that reach the Singleton-type bound are called maximum distance separable (MDS) codes. Any other upper bound for the Hamming distance of block codes can be used to obtain another upper bound for d_f of skew convolutional codes (also for time-varying trellis codes). Using the Plotkin bound for block codes, we obtain the following bound.

Corollary 1 (Heller bound)**.** *The free Hamming distance of $[n,k]$ skew convolutional code $\mathcal{C}$ over $\mathbb{F}_Q$ of degree $\nu = \deg \mathcal{C}$ and memory μ is upper bounded as follows:*

$$d_f \le \min_{i \in \widehat{\mathbb{N}}} \left\lfloor \frac{n(\mu+i)Q^{k(\mu+i)-\nu-1}(Q-1)}{Q^{k(\mu+i)-\nu}-1} \right\rfloor, \tag{29}$$

where $\widehat{\mathbb{N}} = \{1,2,\dots\}$ if $k\mu = \nu$ and $\widehat{\mathbb{N}} = \{0,1,\dots\}$, otherwise.

The bound is named Heller since it was obtained for fixed binary convolutional codes in 1968, see [20,24]. The bound (29) is valid for for time-varying (nonlinear or linear) trellis codes.

In the Hamming metric, the upper bound

$$\sigma \le n - k \tag{30}$$

for the slope σ was obtained in [25] for fixed binary convolutional codes. We conjecture that this bound is true also for non-binary time-varying convolutional codes, and, hence, for skew convolutional codes.

Another interesting metric for convolutional codes is the *sum-rank metric*, which can be applied for multi-shot network coding [26]. The metric is defined as follows. The *rank weight* $w_R(v_t)$ of a vector v_t over the extension field $\mathbb{F}_{q^m}$ is the rank of the vector over the base field $\mathbb{F}_q$, i.e., $w_R(v_t)$ is the maximum number of $\mathbb{F}_q$-linearly independent components of v_t. The sum-rank weight of a sequence v in (16) is the sum of weights of its items v_t. The sum-rank distance between two sequences is the weight of their difference.

The rank of a vector $v_t \in \mathbb{F}_{q^m}^n$ is upper bounded by the Hamming weight $w_H(v_t)$ of the vector, i.e.,

$$w_R(v_t) \le w_H(v_t). \tag{31}$$

Hence, any upper bound for the Hamming metric is an upper bound for the sum-rank metric, and, from Theorem 3, we have the following corollary.

Corollary 2. *The free sum-rank distance d_f of $[n,k]$ skew convolutional code $\mathcal{C}$ of degree $\nu = \deg \mathcal{C}$ is upper bounded by (28) or by* (29)*.*

On the other hand, from (31), we obtain the following lemma.

Lemma 4. *Let the distance of a code $\mathcal{C}$ in the sum-rank metric be d; then, in the Hamming metric, the code distance is at least d.*

We next compute the free distance, active burst distances, and the slope for the code $\widehat{\mathcal{C}}$ from Example 2 for the Hamming and for the sum-rank metrics.

Lemma 5. *In the sum-rank metric, the skew convolutional code $\widehat{\mathcal{C}}$ defined by $G(D)$ in (22) has the ℓ-th active burst distance $d_\ell^{burst} = \ell + 2$ for $\ell = 2, 3, \ldots$, the slope of the active distance is $\sigma = 1$, and the free distance is $d_f = 4$.*

Proof. For this code, the shortest length of a loop is $\ell = 2$; hence, we should consider loops of length $\ell = 2, 3, \ldots$. It follows from (27) that the weight $w_R(v_t) = \text{rank}\, v_t$ of a branch in the code trellis that diverges from or merges to zero state is 2. A branch connecting two nonzero states has weight at least 1. Indeed, for odd t, the branch label v_t in (27) is a linear combination of vectors (α, α^2) and $(1, \alpha^2)$ that are $\mathbb{F}_{q^m}$-linearly independent, and v_t can't be 0 for nonzero coefficients u_{t-1}, u_t. The same is true for even t. Hence, $d_\ell^{\text{burst}} \geq \ell + 2$. On the other hand, the path, corresponding to the information sequence $u_0 = u_1 = \cdots = u_{\ell-1} \neq 0$, $u_\ell = 0$, is an ℓ-loop of weight $\ell + 2$. Hence, $d_\ell^{\text{burst}} \leq \ell + 2$, and the statement of the lemma follows. □

Combining Lemmas 3 and 4, we obtain the following corollary.

Corollary 3. *In the Hamming metric, the skew convolutional code $\widehat{\mathcal{C}}$ defined by $G(D)$ in (22) has the ℓ-th active burst distance $d_\ell^{burst} = \ell + 2$ for $\ell = 2, 3, \ldots$, the slope of the active distance is $\sigma = 1$ and free distance is $d_f = 4$.*

For both metrics, the Hamming and the sum-rank, the upper bounds for the free distance (28) and for the slope (30) for the unit memory $[2, 1]$ code $\widehat{\mathcal{C}}$ become

$$d_f \leq 2n - k + 1 = 4, \text{ and } \sigma \leq n - k = 1.$$

Hence, the skew code $\widehat{\mathcal{C}}$ defined by (22) achieves the Singleton-type upper bound on d_f and can be called MDS codes like in [23]. The Heller bound (29) gives $d_f \leq 4$ as well. The slope of $\widehat{\mathcal{C}}$ also reaches the upper bound (30).

A generator matrix $G(D)$ of a skew convolutional code (and corresponding encoder) is called *catastrophic* if there exists an information sequence $u(D)$ of infinite weight such that the code sequence $v(D) = u(D)G(D)$ has finite weight. The generator matrix $G(D)$ in (22) of skew convolutional code $\widehat{\mathcal{C}}$ with $\theta = (.)^q$ is non-catastrophic, since the slope $\sigma > 0$. Note that, in case of fixed convolutional code $\mathcal{C}'$, i.e., for $\theta = id$, the generator matrix (22) is catastrophic, and the code has $d_f = 2$ and $\sigma = 0$.

4.5. Blocking of Skew Convolutional Codes

A skew convolutional code $\mathcal{C}$, represented as a τ-periodic $[n, k]$ code, can be considered as a $[\tau n, \tau k]$ *fixed* code $\mathcal{C}^{(\tau)}$ by τ*-blocking*, described in [21]. The only difference between $\mathcal{C}$ and $\mathcal{C}^{(\tau)}$ is that the code symbols are grouped in blocks v_t of different lengths in these codes. In this way, known methods to analyze fixed codes can be applied to skew convolutional codes.

For example, the $[2, 1]$ skew code $\widehat{\mathcal{C}}$ with generator matrix (24) has period $\tau = m = 2$ and can be written as $[4, 2]$ *fixed* code $\mathcal{C}^{(\tau)} = \mathcal{C}^{(2)}$ defined by the scalar generator matrix

$$G^{(2)} = \left(\begin{array}{c|c|c} 1\,\alpha\,\alpha\,\alpha^2 & 0\,0\,0\,0 & \\ 0\,0\,1\,\alpha & \alpha^2\,\alpha\,0\,0 & \\ \hline & 1\,\alpha\,\alpha\,\alpha^2 & 0\,0\,0\,0 \\ & 0\,0\,1\,\alpha & \alpha^2\,\alpha\,0\,0 \\ \hline & \dots & \end{array} \right)$$

which coincides with the matrix G in (24) but is written in 2-blocked form. From $G^{(2)}$, we obtain the generator polynomial matrix of the $[4,2]$ blocked code $\widehat{\mathcal{C}}^{(2)}$ as

$$G^{(2)}(D) = \begin{pmatrix} 1 & \alpha & \alpha & \alpha^2 \\ \alpha^2 D & \alpha D & 1 & \alpha \end{pmatrix}.$$

In general, for any skew convolutional code $\mathcal{C}$ and for any i-blocking $\mathcal{C}^{(i)}$, $i \in \mathbb{N}_1$, the codewords, represented by sequences v of elements from $\mathbb{F}_{q^m}$ in (16), are the same for codes $\mathcal{C}$ and $\mathcal{C}^{(i)}$. Hence, the codes have the same properties, e.g., we have

$$\deg \mathcal{C} = \deg \mathcal{C}^{(i)}. \tag{32}$$

5. Dual Skew Convolutional Codes

5.1. Definitions of Duality

The duality of skew convolutional codes can be defined in different ways.

First, consider a skew convolutional code $\mathcal{C}$ over $\mathbb{F}$ in a scalar form as a set of sequences as in (16). For two sequences v and v', where at least one of them is finite, define the scalar product (v, v') as the sum of products of corresponding components, where missing components are assumed to be zeros. We say that the sequences are orthogonal if $(v, v') = 0$.

Definition 3. *The dual code $\mathcal{C}^\perp$ to a skew convolutional $[n,k]$ code $\mathcal{C}$ is an $[n, n-k]$ skew convolutional code $\mathcal{C}^\perp$ such that $(v, v^\perp) = 0$ for all finite length words $v \in \mathcal{C}$ and for all words $v^\perp \in \mathcal{C}^\perp$.*

Another way to define orthogonality is, for example, as follows. Consider two n-words $v(D)$ and $v^\perp(D)$ over $\mathcal{Q}^n$. We say that $v^\perp(D)$ is left-orthogonal to $v(D)$ if $v^\perp(D)v(D) = 0$ and right-orthogonal if $v(D)v^\perp(D) = 0$. A left dual code to a skew convolutional code $\mathcal{C}$ can be defined as

$$\mathcal{C}^\perp_{\text{left}} = \{v^\perp \in \mathcal{Q}^n : v^\perp(D)v(D) = 0 \text{ for all } v \in \mathcal{C}\}.$$

The dual code $\mathcal{C}^\perp_{\text{left}}$ is a left submodule of $\mathcal{Q}^n$, hence it is a skew convolutional code.

Later on, we consider dual codes according to Definition 3 only, since it is more interesting for practical applications.

5.2. Parity Check Matrices

Given a code $\mathcal{C}$ with generator matrix G, we next show how to find a parity check matrix H, such that $GH^T = 0$.

Let a skew $[n,k]$ code $\mathcal{C}$ of memory μ be defined by a polynomial generator matrix $G(D)$ in (12), which corresponds to the scalar generator matrix G in (18). For the dual $[n, n-k]$ code $\mathcal{C}^\perp$, we write a transposed parity check matrix H^T of memory $\mu^\perp$, similar to classical convolutional codes, as

$$H^T = \begin{pmatrix} H_0^T & H_1^T & \dots & H_{\mu^\perp}^T & \\ & \theta(H_0^T) & \dots & \theta(H_{\mu^\perp-1}^T) & \theta(H_{\mu^\perp}^T) \\ & & \dots & & \end{pmatrix} \tag{33}$$

where $\mathrm{rank}(H_0) = n - k$. Similar to [20], we call the matrix $H^{\perp}$ the *syndrome former* and write it in polynomial form as

$$H^T(D) = H_0^T + H_1^T D + \cdots + H_{\mu^{\perp}}^T D^{\mu^{\perp}}. \tag{34}$$

Then, we have the following *parity check matrix* of the causal code $\mathcal{C}$ with the generator matrix (21)

$$H = \begin{pmatrix} H_0 & & \\ H_1 & \theta(H_0) & \\ \vdots & \vdots & \\ H_{\mu^{\perp}} & \theta(H_{\mu^{\perp}-1}) & \vdots \\ & \theta(H_{\mu^{\perp}}) & \end{pmatrix} \tag{35}$$

which, in the case of $\theta = \mathrm{id}$, coincides with the check matrix of a classical fixed convolutional code.

From Definition 3, we have that $vH^T = 0$ for all sequences $v \in \mathcal{C}$ over $\mathbb{F}$. On the other hand, from (4), we have that every codeword $v(D) \in \mathcal{C}$ can be written as $v(D) = u(D)G(D)$. Hence, if we find an $n \times (n-k)$ matrix $H^T(D)$ over $\mathcal{R}$ of full rank such that $G(D)H^T(D) = 0$, then every codeword satisfies $v(D)H^T(D) = u(D)G(D)H^T(D) = 0$ and, vice versa, i.e., if $v(D)H^T(D) = 0$ then $v(D)$ is a codeword of $\mathcal{C}$.

Theorem 4. *We have* $G(D)H^T(D) = 0$ *if and only if* $GH^T = 0$.

Proof. We show the proof for the code of memory $\mu = 1$ (like in Example 2). For the general memory case, the proof follows similarly. Consider the code with generator matrices $G(D)$ and G given by (12) and (18). Let us find a check matrix with memory $\mu^{\perp} = \mu = 1$. Then, we have

$$H^T(D) = H_0^T + H_1^T D \tag{36}$$

and

$$H^T = \begin{pmatrix} H_0^T & H_1^T & & \\ & \theta(H_0^T) & \theta(H_1^T) & \\ & & \theta^2(H_0^T) & \theta^2(H_1^T) \\ & & \cdots & \end{pmatrix}. \tag{37}$$

From the condition $G(D)H^T(D) = 0$, we have the following system of equations for unknowns H_0^T, H_1^T

$$\begin{cases} G_0 H_0^T = 0 \\ G_0 H_1^T + G_1\theta(H_0^T) = 0 \\ G_1\theta(H_1^T) = 0. \end{cases} \tag{38}$$

From the condition $GH^T = 0$, we obtain the following equations: by multiplying the first row of G by H^T we get the system (38), by multiplying the second row of of G by H^T we get the system

$$\begin{cases} \theta(G_0)\theta(H_0^T) = 0 \\ \theta(G_0)\theta(H_1^T) + \theta(G_1)\theta^2(H_0^T) = 0 \\ \theta(G_1)\theta^2(H_1^T) = 0 \end{cases}$$

which is equivalent to (38). Multiplication of other rows of G by H^T does not give new equations. Hence, conditions $G(D)H^T(D) = 0$ and $GH^T = 0$ give the same system (38). □

Example 3. *For the code $\widehat{\mathcal{C}}$ from Example 2, we write $H_0 = (a, c)$ and $H_1 = (b, d)$. Using G_0, G_1 from (22) and solving the system (38), we obtain $H_0 = (\alpha, 1)$ and $H_1 = (1, \alpha)$. Hence, $H(D) = (\alpha + D, 1 + \alpha D)$ and a parity check matrix H of the code $\widehat{\mathcal{C}}$, which is a generator matrix for the dual code $\widehat{\mathcal{C}}^{\perp}$, is as follows:*

$$H = \left(\begin{array}{cccccccc} \alpha & 1 & & & & & & \\ 1 & \alpha & \alpha^2 & 1 & & & & \\ & & 1 & \alpha^2 & \alpha & 1 & \vdots & \\ & & & & 1 & \alpha & & \end{array} \right).$$

5.3. Trellises of Dual Codes

Similar to fixed convolutional codes, we have the following theorem:

Theorem 5. *For a skew convolutional code $\mathcal{C}$ and its dual $\mathcal{C}^{\perp}$, we have* $\deg \mathcal{C} = deg\mathcal{C}^{\perp}$.

Proof. Denote by τ and $\tau^{\perp}$ periods of the codes $\mathcal{C}$ and $\mathcal{C}^{\perp}$, respectively. Let ℓ be the least common multiple of periods τ and $\tau^{\perp}$; then, the ℓ-blocked codes $\mathcal{C}^{(\ell)}$ and $(\mathcal{C}^{\perp})^{(\ell)}$ are both fixed convolutional codes. The fixed codes $\mathcal{C}^{(\ell)}$ and $(\mathcal{C}^{\perp})^{(\ell)}$ are dual to each other, since blocking does not change code sequences, hence $\deg \mathcal{C}^{(\ell)} = \deg(\mathcal{C}^{\perp})^{(\ell)}$, see, e.g., Theorem 2.69, [20] for fixed dual convolutional codes. From (32), we have $\deg \mathcal{C} = \mathcal{C}^{(\ell)}$, $\deg \mathcal{C}^{\perp} = \deg(\mathcal{C}^{\perp})^{(\ell)}$, hence $\deg \mathcal{C} = \deg \mathcal{C}^{\perp}$. □

It follows from Theorem 5 that the number of states at one level of the code trellis (trellis complexity) is the same for an original code $\mathcal{C}$ and for its dual $\mathcal{C}^{\perp}$ and equals $Q^{\deg \mathcal{C}}$.

The trellis of the dual code $\widehat{\mathcal{C}}^{\perp}$ obtained from the matrix H in Example 3 is shown in Figure 4. The trellis has $Q^{\deg \widehat{\mathcal{C}}} = 4^1 = 4$ states labeled by elements of the set $\mathcal{S} = \{0, 1, \alpha, \alpha^2\}$. Every word of the dual code $\widehat{\mathcal{C}}^{\perp}$ is represented by a path in the trellis that starts from a state $s_{-1} \in \mathcal{S}$ and goes to the right. For the trellis section corresponding to time $t = 0, 1, \ldots$, the edge connecting states s_{t-1} and s_t are labeled by $v_t^{\perp}$ computed as follows:

$$v_t^{\perp} = \begin{cases} s_{t-1}(\alpha^2, 1) + s_t(1, \alpha^2) & \text{for odd } t, \\ s_{t-1}(\alpha, 1) + s_t(1, \alpha) & \text{for even } t. \end{cases}$$

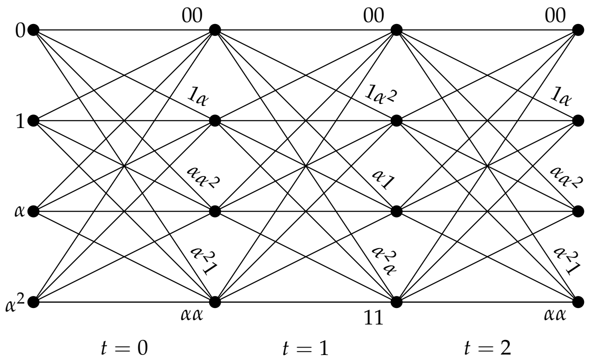

Figure 4. Time-varying trellis of the dual skew code $\widehat{\mathcal{C}}^{\perp}$ from Example 3.

6. Trellis Decoding of Skew Convolutional Codes

For a given skew convolutional code $\mathcal{C}$, we showed how to obtain a code trellis using a generator matrix of the code. Another way to obtain a code trellis of $\mathcal{C}$ using a parity check matrix H was proposed in [27]. Having a code trellis, one can use the Viterbi decoder [4] for maximum likelihood sequence decoding or the BCJR decoder [28] for symbol-wise decoding.

For an $[n,k]$ skew convolutional code, the complexity of the Viterbi decoder has order $\varkappa = nQ^k Q^{\deg \mathcal{C}}$ operations (additions and binary selections), which exponentially increases in k and might be high for high rate codes. Using detailed code trellis [27,29], where every edge is labeled by a single field element, the decoding complexity can be reduced to

$$\varkappa = nQ^{\min\{k,n-k\}} Q^{\deg \mathcal{C}}. \tag{39}$$

Another advantage of the method in [29] is that it can be applied to every trellis section separately, which is convenient for time-varying codes. The decoding complexity of a particular code can also be decreased using methods in [30]. The complexity of the BCJR decoding algorithm has the same order as in (39) as well.

Symbol-wise decoding of a skew convolutional code $\mathcal{C}$ can be implemented using a trellis of the dual code $\mathcal{C}^\perp$, see [31–33]. The order of decoding complexity in this case is also given in (39).

7. Conclusions

A new class of non-binary skew convolutional codes was defined that extends the class of fixed convolutional codes. The skew convolutional codes are equivalent to periodic time-varying classical convolutional codes but have as compact a description as fixed convolutional codes.

Given a field $\mathbb{F} = \mathbb{F}_{p^M} = \mathbb{F}_{q^m}$ of characteristic p and code parameters n, k and μ; for every authomorphism $\theta(a) = q^a$ of the field, the subclass $SCC(\theta)$ of skew convolutional $[n,k]$ codes of memory μ over the field is defined. All the subclasses have the same number of codes. In case of the identity automorphism $\theta = \text{id}$, we obtain the subclass $SCC(\text{id})$ of classical fixed convolutional codes. Any other automorphism θ of the field gives a subclass $SCC(\theta)$ of skew convolutional codes that can be represented as a periodic time-varying convolutional code with typical period m . The total number of the subclasses $SCC(\theta)$ is equal to the number of divisors of M, which is usually not a large number. The class of m-periodic time-varying convolutional codes is larger than the class of skew convolutional codes. Every code in the subclass $SCC(\theta)$ is defined by a $k \times n$ polynomial generator matrix $G(D)$ over the ring of θ-skew polynomials; hence, the descriptions of skew codes and fixed codes are the same, and the description is given by the same matrix $G(D)$.

Every τ-periodic convolutional $[n,k]$ code can be written as a fixed $[\tau n, \tau k]$ code; hence, skew convolutional codes can be analyzed by methods known for fixed codes. We showed how to design generator and parity check matrices in polynomial and scalar forms, encoders and code trellises for skew convolutional codes, and their duals. Using code trellises for original and dual codes, in the case of channels without memory, one can apply Viterbi or BCJR decoding algorithms, or the dualized BCJR algorithm.

Future work. We gave just a first encounter with skew convolutional codes. There are many open problems remaining. The algebraic structure of classical fixed convolutional codes is well understood, see, e.g., [20,21] and references therein. The questions such as how to obtain a canonical generator matrix of a skew convolutional code and its dual, or how to design encoders of a fractional generator matrix can be considered in the future. Another open problem is to find good skew convolutional codes reaching an upper bound on the free distance. One possibility to obtain skew convolutional codes is based on unwrapping skew quasi-cyclic (QC) block codes (see such codes in [17]) in a way similar to [34] or [35], where it is shown how fixed classical convolutional codes can be obtained by unwrapping QC block codes and vice versa.

Author Contributions: Conceptualization, V.S., W.L., O.G. and G.K.; Investigation, V.S., W.L., O.G. and G.K.; Writing—original draft, V.S., W.L., O.G. and G.K. All authors have read and agreed to the published version of the manuscript.

Funding: The work of V. Sidorenko was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 801434) and by the Chair of Communications Engineering at the Technical University of Munich. The work of O. Günlü was supported by the German Federal Ministry of Education and Research (BMBF) within the national initiative for "Post Shannon

Communication (NewCom)" under Grant 16KIS1004. The work of G. Kramer was supported in part by the German Research Foundation through DFG Grant KR 3517/9-1.

Acknowledgments: The authors are grateful to the guest editor, to the assistant editor, and especially to anonymous reviewers for their comments and advice that allowed us to improve the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Elias, P. Coding for noisy channels. *IRE Conv. Rec.* **1955**, *4*, 37–46.
2. Fano, R. A heuristic discussion of probabilistic decoding. *IEEE Trans. Inf. Theory* **1963**, *9*, 64–74. [CrossRef]
3. Massey, J.L. *Threshold Decoding*; MIT Press: Cambridge, MA, USA, 1963.
4. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [CrossRef]
5. Berrou, C.; Glavieux, A.; Thitimajshima, P. Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1. In Proceedings of the ICC '93—IEEE International Conference on Communications, Geneva, Switzerland, 23–26 May 1993; Volume 2, pp. 1064–1070. [CrossRef]
6. IEEE Standard for Telecommunications and Information Exchange between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High Speed Physical Layer in the 5 GHz Band. Available online: https://ieeexplore.ieee.org/document/815305 (accessed on 1 December 2020).
7. Filler, T.; Judas, J.; Fridrich, J. Minimizing Additive Distortion in Steganography Using Syndrome-Trellis Codes. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 920–935. [CrossRef]
8. Ouahada, K. Nonbinary convolutional codes and modified M-FSK detectors for power-line communications channel. *J. Commun. Netw.* **2014**, *16*, 270–279. [CrossRef]
9. Holzbaur, L.; Freij-Hollanti, R.; Wachter-Zeh, A.; Hollanti, C. Private Streaming With Convolutional Codes. *IEEE Trans. Inf. Theory* **2020**, *66*, 2417–2429. [CrossRef]
10. Mooser, M. Some periodic convolutional codes better than any fixed code (Corresp.). *IEEE Trans. Inf. Theory* **1983**, *29*, 750–751. [CrossRef]
11. Lee, P.J. There are many good periodically time-varying convolutional codes. *IEEE Trans. Inf. Theory* **1989**, *35*, 460–463. [CrossRef]
12. Gabidulin, E.M. Theory of Codes with Maximum Rank Distance. *Probl. Inform. Trans.* **1985**, *21*, 1–12.
13. Boucher, D.; Ulmer, F. Coding with skew polynomial rings. *J. Symb. Comput.* **2009**, *44*, 1644–1656. [CrossRef]
14. Boucher, D.; Ulmer, F. *Codes as Modules over Skew Polynomial Rings*; Springer: Berlin/Heidelberg, Germany, 2009. [CrossRef]
15. Martínez-Peñas, U. Sum-Rank BCH Codes and Cyclic-Skew-Cyclic Codes. *arXiv* **2020**, arXiv:2009.04949.
16. Gluesing-Luerssen, H. Skew-Polynomial Rings and Skew-Cyclic Codes. *arXiv* **2019**, arXiv:1902.03516.
17. Abualrub, T.; Ghrayeb, A.; Aydin, N.; Siap, I. On the Construction of Skew Quasi-Cyclic Codes. *IEEE Trans. Inf. Theory* **2010**, *56*, 2081–2090. [CrossRef]
18. Ore, O. Theory of Non-Commutative Polynomials. *Ann. Math.* **1933**, *34*, 480–508. [CrossRef]
19. Clark, P. *Non-Commutative Algebra*; University of Georgia: Athens, GA, USA, 2012.
20. Johannesson, R.; Zigangirov, K.S. *Fundamentals of Convolutional Coding*; John Wiley and Sons, Ltd.: Hoboken, NJ, USA, 2015.
21. McEliece, R.J. The Algebraic Theory of Convolutional Codes. In *Handbook of Coding Theory*; Chapter 12; Pless, V.S., Huffman, W.C., Eds.; Elsevier Science: Amsterdam, The Netherlands, 1998; Volume I, pp. 1065–1138.
22. Pollara, F.; McEliece, R.J.; Abdel-Ghaffar, K. Finite-state codes. *IEEE Trans. Inf. Theory* **1988**, *34*, 1083–1089. [CrossRef]
23. Rosenthal, J.; Smarandache, R. Maximum Distance Separable Convolutional Codes. *Appl. Algebra Eng. Commun. Comput.* **1998**, *10*, 15–32. [CrossRef]
24. Gluesing-Luerssen, H.; Schmale, W. Distance bounds for convolutional codes and some optimal codes. *arXiv* **2003**, arXiv:math/0305135.

25. Jordan, R.; Pavlushkov, V.; Zyablov, V.V. An upper bound on the slope of convolutional codes. In Proceedings of the 2002 IEEE International Symposium on Information Theory, Lausanne, Switzerland, 30 June–5 July 2002; p. 424. [CrossRef]
26. Wachter-Zeh, A.; Stinner, M.; Sidorenko, V. Convolutional Codes in Rank Metric with Application to Random Network Coding. *IEEE Trans. Inf. Theory* **2015**, *61*, 3199–3213. [CrossRef]
27. Sidorenko, V.; Zyablov, V. Decoding of convolutional codes using a syndrome trellis. *IEEE Trans. Inf. Theory* **1994**, *40*, 1663–1666. [CrossRef]
28. Bahl, L.; Cocke, J.; Jelinek, F.; Raviv, J. Optimal decoding of linear codes for minimizing symbol error rate (Corresp.). *IEEE Trans. Inf. Theory* **1974**, *20*, 284–287. [CrossRef]
29. Li, W.; Sidorenko, V.; Jerkovits, T.; Kramer, G. On Maximum-Likelihood Decoding of Time-Varying Trellis Codes. In Proceedings of the 2019 XVI International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY), Moscow, Russia, 21–25 October 2019; pp. 104–109. [CrossRef]
30. Lafourcade, A.; Vardy, A. Optimal sectionalization of a trellis. *IEEE Trans. Inf. Theory* **1996**, *42*, 689–703. [CrossRef]
31. Hartmann, C.; Rudolph, L. An optimum symbol-by-symbol decoding rule for linear codes. *IEEE Trans. Inf. Theory* **1976**, *22*, 514–517. [CrossRef]
32. Berkmann, J.; Weiss, C. On dualizing trellis-based APP decoding algorithms. *IEEE Trans. Commun.* **2002**, *50*, 1743–1757. [CrossRef]
33. Srinivasan, S.; Pietrobon, S.S. Decoding of High Rate Convolutional Codes Using the Dual Trellis. *IEEE Trans. Inf. Theory* **2010**, *56*, 273–295. [CrossRef]
34. Esmaeili, M.; Gulliver, T.A.; Secord, N.P.; Mahmoud, S.A. A link between quasi-cyclic codes and convolutional codes. *IEEE Trans. Inf. Theory* **1998**, *44*, 431–435. [CrossRef]
35. Kudryashov, B.D.; Zakharova, T.G. Block Codes from Convolution Codes. *Probl. Peredachi Inf.* **1989**, *25*, 98–102.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.