mathematics

MDPI

Article

Minimal State-Space Representation of Convolutional Product Codes

Joan-Josep Climent [1], Diego Napp [1], Raquel Pinto [2] and Verónica Requena [1,*]

[1] Departament de Matemàtiques, Universitat d'Alacant, E-03690 Alacant, Spain; jcliment@ua.es (J.-J.C.); diego.napp@ua.es (D.N.)
[2] CIDMA—Center for Research and Development in Mathematics and Applications, Department of Mathematics, University of Aveiro, 3810-193 Aveiro, Portugal; raquel@ua.pt
* Correspondence: vrequena@ua.es

Abstract: In this paper, we study product convolutional codes described by state-space representations. In particular, we investigate how to derive state-space representations of the product code from the horizontal and vertical convolutional codes. We present a systematic procedure to build such representation with minimal dimension, i.e., reachable and observable.

Keywords: convolutional code; generator matrix; state-space representation; minimal representation; product code

check for updates

Citation: Climent, J.-J.; Napp, D.; Pinto, R.; Requena, V. Minimal State-Space Representation of Convolutional Product Codes. *Mathematics* **2021**, *9*, 1410. https://doi.org/10.3390/math9121410

Academic Editors: Michele Elia, J. Carmelo Interlando and Nadir Murru

Received: 31 March 2021
Accepted: 11 June 2021
Published: 17 June 2021

1. Introduction

It is well-known that the combination of codes can yield a new code with better properties than the single codes alone. Such combinations have been widely used in coding theory in different forms, e.g., concatenation, product codes, turbo codes, array codes, or using EVENODD and interleaving methods [1–8]. The advantages of the combination of codes can be due to, for instance, larger distance, lower decoding complexity or improved burst error correction. In this paper, we shall focus on the so-called product codes, which is a natural generalization of the interleaved schemes. More concretely, we will focus on product convolutional codes.

In the context of block product codes, the codewords are constant matrices with entries in a finite field. We may consider that both rows and columns are encoded into error-correcting codes. Hence, for encoding, first the row redundant symbols are obtained (horizontal encoding using $\mathcal{C}_h$), and then the column redundant symbols (vertical encoding using $\mathcal{C}_v$). If $\mathcal{C}_h$ has minimum distance d_h, and $\mathcal{C}_v$ has minimum distance d_v, it is easy to see that the product code, denoted by $\mathcal{C}_h \otimes \mathcal{C}_v$, has minimum distance $d_h d_v$. This class of product codes has been thoroughly studied and is widely used to correct burst and random errors using many possible different decoding procedures. However, the product of two *convolutional* codes has been less investigated and many properties that are known for block codes are still to be investigated in the convolutional context.

Naturally, the class of convolutional codes generalizes the class of linear block codes, and, therefore, they are mathematically more involved than block codes. In this context, the data are considered as a sequence in contrast with block codes which operate with fixed message blocks (matrices in this case). Even though they split the data into blocks of a fixed rate as block codes do, the relative position of each block in the sequence is taken into account. The blocks are not encoded independently and previously encoded data (matrices in this case) in the sequence have an effect over the next encoded node. Because of this, convolutional codes have memory and can be viewed as linear systems over a finite field (see, for instance, [5,9–18]). A description of convolutional codes can be provided by a time-invariant discrete linear system called discrete-time state-space system in control theory (see [19–21]). Hence, we consider product convolutional codes described by state-space representations. Convolutional codes have already been thoroughly investigated within this framework and fundamental system theoretical properties, such as observability, reachability, and minimality, have been derived in [11–14].

It is worth mentioning the results derived in [22,23] on fundamental algebraic properties of the encoders representing product convolutional codes. In addition, they showed that every product convolutional code can be represented as a woven code and introduced the notion of block distances. In [22], it is was shown that, if the generator matrices of the horizontal and vertical convolutional codes are minimal basic, then the generator matrix of the product code is also minimal basic. In this work, we continue this thread of research but within input-state-space framework instead of working with generator matrices. We present a constructive methodology to build a minimal state-space representations for these codes from two minimal state-space representation of the corresponding horizontal $\mathcal{C}_h$ and vertical $\mathcal{C}_v$ convolutional codes. These representations are, therefore, reachable and observable and are easily constructed by sorting and selecting some of the entries of a given matrix built upon the state-space representations of $\mathcal{C}_h$ and $\mathcal{C}_v$. This is done directly without using the encoder matrix representations of the convolutional codes. The derived representations are minimal and, therefore, are reachable and observable. Moreover, they are easily constructed by sorting and selecting some of the entries of a given matrix built upon the state-space representations of $\mathcal{C}_h$ and $\mathcal{C}_v$.

Recently, there have been new advances in the original idea of deriving an algebraic decoding algorithm of convolutional codes using state space representations. The idea was first proposed in [24] and heavily uses the structure of these representations to derive a general procedure, which will allow for extending known decoding algorithms for block codes (like, e.g., the Berlekamp–Massey algorithm) to convolutional codes. More concretely, the algorithm iteratively computes the state vector $\boldsymbol{x}_t$ inside the trellis diagram, and, once this state vector is constructed, the algorithm computes, in an algebraic manner, a new state vector $\boldsymbol{x}_{t+s}$, where s is related to the observability index of the state representation. Recently, these ideas have been further developed in [25,26]. Hence, the ideas of this paper can be used to built a minimal state space representation of a product convolutional code with the property that its decoding can be simplified by considering the simpler horizontal and vertical component codes and applying the decoding algorithms developed in [25,26].

In [27], an input–state–output representation of each one of the convolutional codes $\mathcal{C}_h$ and $\mathcal{C}_v$, two input–state–output representations of the product convolutional code $\mathcal{C}_h \otimes \mathcal{C}_v$ were introduced, but none of them are minimal, even if the two input–state–output representations are both minimal. In this paper, we give a solution to this problem.

The rest of the paper is organized as follows: In Section 2, we introduce the background on polynomial matrices and convolutional codes to understand the paper. In Section 3, we describe how product convolutional codes can be viewed as a convolutional code whose generator matrix is the Kronecker product of the corresponding generator matrices. In Section 4, we provide a state-space realization of the product convolutional code based on a state-space realization of each one of the convolutional codes involved in the product. Finally, in Section 5, we present the conclusions and future work.

2. Preliminaries

Let $\mathbb{F}$ be a finite field, $\mathbb{F}[z]$ the ring of polynomials in the variable z and coefficients in $\mathbb{F}$, and $\mathbb{F}(z)$ the set of rational functions in the variable z and coefficients in $\mathbb{F}$.

Assume that k and n are positive integers with $n > k$, denote by $\mathbb{F}[z]^{n\times k}$ the set of all $n \times k$ matrices with entries in $\mathbb{F}[z]$, and denote by $\mathbb{F}[z]^n$ the set $\mathbb{F}[z]^{n\times 1}$.

A matrix $U(z) \in \mathbb{F}[z]^{k\times k}$ is called **unimodular** if it admits a polynomial inverse; that is, its determinant is a nonzero element of $\mathbb{F}$ (see, for example [28,29]).

Assume that $G(z) \in \mathbb{F}[z]^{n\times k}$. The **internal degree** of $G(z)$ is the maximum degree of the $k \times k$ minors of $G(z)$. We said that $G(z)$ is **basic** if its internal degree is the minimum of the internal degrees of the matrices $G(z)U(z)$, for all invertible matrices $U(z) \in \mathbb{F}(z)^{k\times k}$; i.e., the internal degree of $G(z)$ is as small as possible (see, for instance, [17,30–33]. In particular, if $U(z)$ is unimodular, then $G(z)$ and $G(z)U(z)$ have the same internal degrees. $G(z)$ is called **right prime**, if, for every factorization, $G(z) = G'(z)U(z)$ with $G'(z) \in \mathbb{F}[z]^{n\times k}$ and $U(z) \in \mathbb{F}[z]^{k\times k}$; necessarily, $U(z)$ is unimodular (see, for instance, [28,33,34]). Furthermore, $G(z)$ is basic if and only if any (and therefore all) of the following equivalent conditions are satisfied: $G(z)$ is right prime, $G(z)$ has a polynomial left inverse [17,32]).

Assume that $G(z) = \left[g_{ij}(z)\right] \in \mathbb{F}[z]^{n\times k}$ and denote by $\nu_j = \max_{1\leq i\leq n} \deg\left(g_{ij}(z)\right)$ the j-th column degree of $G(z)$. We said that $G(z)$ is **column reduced** if the rank of the **high-order coefficient matrix** $G_\infty = \left[g_{ij}^{(\nu_j)}\right] \in \mathbb{F}^{n\times k}$ is k, where $g_{ij}^{(\nu_j)}$ is the coefficient of z^{ν_j} in $g_{ij}(z)$. Equivalently, $G(z)$ is column reduced if and only if its internal and external degrees coincide, where the **external degree** of $G(z)$ is the number $\sum_{j=1}^{k} \nu_j$. Note that the internal degree of a polynomial matrix is always less than or equal to its external degree [17]. For any $G(z) \in \mathbb{F}[z]^{n\times k}$, there exists a unimodular matrix $U(z) \in \mathbb{F}[z]^{k\times k}$ such that $G(z)U(z)$ is column reduced. Moreover, if $G(z), G(z)U(z) \in \mathbb{F}[z]^{n\times k}$ are column reduced matrices with $U(z) \in \mathbb{F}[z]^{k\times k}$ unimodular, then $G(z)$ and $G(z)U(z)$ have the same column degrees, up to a permutation. Column reduced matrices are also called **minimal matrices** [12,13,33]). Basic and reduced matrices are also called **minimal-basic matrices** [30–32] or **canonical matrices** [17].

A rate k/n **convolutional code** $\mathcal{C}$ is an $\mathbb{F}[z]$-submodule of rank k of the module $\mathbb{F}[z]^n$ (see [18,20,35]). Since $\mathbb{F}[z]$ is a Principle Ideal Domain, a convolutional code $\mathcal{C}$ has always a well-defined rank k, and there exists $G(z) \in \mathbb{F}[z]^{n\times k}$, of rank k, such that (see [35])

$$\mathcal{C} = \mathrm{im}_{\mathbb{F}[z]}(G(z)) = \left\{ \boldsymbol{v}(z) \in \mathbb{F}[z]^n \mid \boldsymbol{v}(z) = G(z)\boldsymbol{u}(z) \text{ with } \boldsymbol{u}(z) \in \mathbb{F}[z]^k \right\}$$

where $\boldsymbol{u}(z)$ is the **information vector**, $\boldsymbol{v}(z)$ is the corresponding **codeword**, and $G(z)$ is the **generator** or **encoder matrix** of $\mathcal{C}$.

If $G(z) \in \mathbb{F}[z]^{n\times k}$ is a generator matrix of $\mathcal{C}$ and $U(z) \in \mathbb{F}[z]^{k\times k}$ is unimodular, then $G(z)U(z)$ is also a generator matrix of $\mathcal{C}$. Therefore, all generator matrices of $\mathcal{C}$ have the same internal degree. The **degree** or **complexity** of $\mathcal{C}$ is the internal degree of one (and therefore any) generator matrix and, therefore, is also equal to the external degree of one (and therefore any) column reduced generator matrix (see [17,34]). The column degrees of a basic and column reduced generator matrix of $\mathcal{C}$ are called **Forney indices** of $\mathcal{C}$.

Since $\mathcal{C}$ always admits a generator matrix $G(z) \in \mathbb{F}[z]^{n\times k}$ which is column reduced, the row degrees $\nu_1, \nu_2, \ldots, \nu_k$ of $G(z)$ are the Forney indices of $\mathcal{C}$ and $\sum_{j=1}^{k} \nu_j = \delta$, the degree of $\mathcal{C}$. From now on, we refer to a rate k/n convolutional code with degree δ as an (n, k, δ) convolutional code.

An (n, k, δ) convolutional code $\mathcal{C}$ can be described by a time invariant linear system (see [11,14,16,17]), denoted by (A, B, C, D),

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{v}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}_0 = \boldsymbol{0}, \tag{1}$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{n\times m}$ and $D \in \mathbb{F}^{n\times k}$. For each instant t, we call $\boldsymbol{x}_t \in \mathbb{F}^m$ the **state vector**, $\boldsymbol{u}_t \in \mathbb{F}^k$ the **input vector**, and $\boldsymbol{v}_t \in \mathbb{F}^n$ the **output vector**, and we say that the system (A, B, C, D) has dimension m. In the literature of linear systems, the above representation is known as the **state-space representation** (see, for example, [28,29,36–38]). If we define $\boldsymbol{u}(z) = \sum_{t\geq 0} \boldsymbol{u}_t z^t$, and $\boldsymbol{v}(z) = \sum_{t\geq 0} \boldsymbol{v}_t z^t$, it follows from expression (1) that $\boldsymbol{v}(z) = G(z)\boldsymbol{u}(z)$ where

$$G(z) = C(I_m - zA)^{-1}B\,z + D \tag{2}$$

is the **transfer matrix** of the system. We say that (A, B, C, D) is a **realization** of $G(z)$ if $G(z)$ is the transfer matrix of (A, B, C, D).

For a given transfer matrix $G(z)$, there are, in general, many possible realizations. A realization (A, B, C, D) of $G(z)$ is called **minimal** if it has minimal dimension, and this happens if and only if the pair (A, B) is reachable and the pair (A, C) is observable (see, for instance, [28,29,36]). Recall that the pair (A, B) is called **reachable** if

$$\operatorname{rank}\left(\begin{bmatrix} B & AB & \cdots & A^{\delta-1}B \end{bmatrix}\right) = \delta$$

or equivalently (see [39]), $\operatorname{rank}\left(\begin{bmatrix} \lambda I_\delta - A & B \end{bmatrix}\right) = \delta$, for all $\lambda \in \bar{\mathbb{F}}$, where $\bar{\mathbb{F}}$ is the closure of $\mathbb{F}$. Analogously, the pair (A, C) is **observable** if and only if the pair (A^T, C^T) is reachable. The dimension of a minimal realization of a transfer matrix $G(z)$ is called the **McMillan degree** of $G(z)$. In the particular case that $G(z)$ is a column reduced generator matrix of a convolutional code $\mathcal{C}$, the McMillan degree of $G(z)$ coincides with the degree δ of $\mathcal{C}$.

Reachability and observability represent two major concepts of control system theory. They were introduced by Kalman in [40] in the context of systems theory and, in [35], the definitions of reachability and observability of convolutional codes were presented, see also [9,41–43]. These notions are not only important for characterizing minimality of our state-space realization but also to describe the possibility of driving the state everywhere with the appropriate selection of inputs (reachability) and the ability of computing the state after from the observation of output sequence.

A system (A, B, C, D) is a realization of a convolutional code $\mathcal{C}$ if $\mathcal{C}$ is equal to the set of outputs corresponding to polynomial inputs $\boldsymbol{u}(z) \in \mathbb{F}[z]^k$ and to zero initial conditions; i.e., $\boldsymbol{x}_0 = 0$. The minimal dimension of a realization of $\mathcal{C}$ is equal to the degree of $\mathcal{C}$ and the minimal realizations of the column reduced generator matrices of $\mathcal{C}$ are minimal realizations of the code.

If $(\bar{A}, \bar{B}, \bar{C}, \bar{D})$, with $\bar{A} \in \mathbb{F}^{m\times m}$, $\bar{B} \in \mathbb{F}^{m\times k}$, $\bar{C} \in \mathbb{F}^{n\times m}$, and $\bar{D} \in \mathbb{F}^{n\times k}$ is a non-minimal realization of a transfer matrix with McMillan degree δ, from the Kalman's decomposition theorem (see, for example, [28,29,36,38,40,44–47]), there exists an invertible matrix $S \in \mathbb{F}^{m\times m}$ such that

$$(S\bar{A}S^{-1}, S\bar{B}, \bar{C}S^{-1}, \bar{D}) = \left(\begin{bmatrix} A & O & \widetilde{A}_{13} & O \\ \widetilde{A}_{21} & \widetilde{A}_{22} & \widetilde{A}_{23} & \widetilde{A}_{24} \\ O & O & \widetilde{A}_{33} & O \\ O & O & \widetilde{A}_{43} & \widetilde{A}_{44} \end{bmatrix}, \begin{bmatrix} B \\ \widetilde{B} \\ O \\ O \end{bmatrix}, \begin{bmatrix} C & O & \widetilde{C} & O \end{bmatrix}, D \right),$$

where $A \in \mathbb{F}^{\delta\times\delta}$, $B \in \mathbb{F}^{\delta\times k}$, $C \in \mathbb{F}^{n\times\delta}$ and the pair (A, B) is reachable, the pair (A, C) is observable, and

$$\bar{C}(I_m - z\bar{A})^{-1}\bar{B}\,z + \bar{D} = C(I_\delta - zA)^{-1}B\,z + D.$$

That is, (A, B, C, D) is a minimal realization of the transfer matrix $G(z)$. Moreover, if (A', B', C', D') is another minimal realization of $G(z)$, then there exists a unique invertible matrix $P \in \mathbb{F}^{\delta\times\delta}$ such that

$$A' = PAP^{-1}, \quad B' = PB, C' = CP^{-1}, \quad \text{and} \quad D' = D.$$

The state-space representation in expression (1), also known as, **driving representation**, is different from the **input–state–output representation** (see [21]) given by

$$\left.\begin{aligned} \boldsymbol{x}_{t+1} &= A\boldsymbol{x}_t + B\boldsymbol{u}_t \\ \boldsymbol{y}_t &= C\boldsymbol{x}_t + D\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t = \begin{bmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{bmatrix}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}_0 = \boldsymbol{0},$$

where $A \in \mathbb{F}^{m\times m}$, $B \in \mathbb{F}^{m\times k}$, $C \in \mathbb{F}^{(n-k)\times m}$ and $D \in \mathbb{F}^{(n-k)\times k}$. This input–state–output representation has been thoroughly studied by many authors [9,10,18,19,21,27,33,35,48], and the codewords are the finite support input–output sequences $\{\boldsymbol{v}_t\}_{t\geq 0}$ corresponding to finite support state sequences $\{\boldsymbol{x}_t\}_{t\geq 0}$.

The next theorem (see [11,14]) provides a state-space realization for a given polynomial matrix, and it will be very useful in Section 4.

Theorem 1. *Let* $G(z) = \begin{bmatrix} \boldsymbol{g}_1(z) & \boldsymbol{g}_2(z) & \cdots & \boldsymbol{g}_k(z) \end{bmatrix} \in \mathbb{F}[z]^{n\times k}$ *be a matrix with column degrees* $\nu_1, \nu_2, \ldots, \nu_k$. *Assume that* $\boldsymbol{g}_j(z) = \sum_{\ell=0}^{\nu_j} \boldsymbol{g}_j^{(\ell)} z^\ell$, *for* $j = 1, 2, \ldots, k$, *and consider the matrices*

$$A_j = \begin{bmatrix} \mathbf{0}^T & 0 \\ I_{\nu_j - 1} & \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_j \times \nu_j}, \quad B_j = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_j}, \quad C_j = \begin{bmatrix} \boldsymbol{g}_j^{(1)} & \boldsymbol{g}_j^{(2)} & \cdots & \boldsymbol{g}_j^{(\nu_j)} \end{bmatrix} \in \mathbb{F}^{n \times \nu_j}.$$

If $\delta = \sum_{j=1}^{k} \nu_j$ *and*

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_k \end{bmatrix} \in \mathbb{F}^{\delta\times\delta}, \quad B = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_k \end{bmatrix} \in \mathbb{F}^{\delta\times k},$$

$$C = \begin{bmatrix} C_1 & C_2 & \cdots C_k \end{bmatrix} \in \mathbb{F}^{n\times\delta}, \quad D = \begin{bmatrix} \boldsymbol{g}_1^{(0)} & \boldsymbol{g}_2^{(0)} & \cdots & \boldsymbol{g}_k^{(0)} \end{bmatrix} \in \mathbb{F}^{n\times k},$$

then the pair (A, B) *is reachable. Moreover, if* $G(z)$ *is column reduced, then the pair* (A, C) *is observable, and, therefore,* (A, B, C, D) *is a minimal realization of* $G(z)$.

For the realization (A, B, C, D) of $G(z)$ introduced in the previous theorem, it follows from expression (2), that $G(z) = CE(z) + D$, where

$$E(z) = \begin{bmatrix} E_1(z) & & & \\ & E_2(z) & & \\ & & \ddots & \\ & & & E_k(z) \end{bmatrix} \quad \text{with} \quad E_j(z) = \begin{bmatrix} z \\ z^2 \\ \vdots \\ z^{\nu_j} \end{bmatrix}, \text{ for } j = 1, 2, \ldots, k. \tag{3}$$

The following example will help us to understand the previous theorem.

Example 1. *Let* $\mathbb{F} = GF(2)$ *be the Galois field of two elements and consider the polynomial matrix*

$$G(z) = \begin{bmatrix} z^2 & z+1 \\ z+1 & z \\ 1 & 1 \end{bmatrix} \in \mathbb{F}[z]^{3\times 2}.$$

Since $\nu_1 = 2$, $\nu_2 = 1$, *and* $\mathrm{rank}(G_\infty) = 2$. *It follows that* $G(z)$ *is column reduced. Now consider the matrices*

$$A_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Then, according to Theorem 1, it follows that (A, B, C, D) *is a minimal state-space realization of* $G(z)$ *with*

$$A = \begin{bmatrix} A_1 & \\ & A_2 \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & \\ & B_2 \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}.$$

Moreover, $E(z) = \begin{bmatrix} z & 0 \\ z^2 & 0 \\ 0 & z \end{bmatrix}$ *and* $G(z) = CE(z) + D$.

3. Product Convolutional Codes

In this section, we introduce the product of two convolutional codes called *horizontal* and *vertical* codes, respectively. Assume that $\mathcal{C}_h$ and $\mathcal{C}_v$ are horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v), respectively. Then, the **product convolutional code** (see [22,49]) $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$ is defined to be the convolutional code whose codewords consist of all $V(z) \in \mathbb{F}[z]^{n_v \times n_h}$ whose columns belong to $\mathcal{C}_v$ and whose rows belong to $\mathcal{C}_h$.

Encoding of the product convolutional code $\mathcal{C}$ can be done as follows (see [22,49]): Let $G_h(z) \in \mathbb{F}[z]^{n_h \times k_h}$ and $G_v(z) \in \mathbb{F}[z]^{n_v \times k_v}$ be generator matrices of the component convolutional codes $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively. Denote by $U(z) \in \mathbb{F}[z]^{k_v \times k_h}$ an information matrix. Now, we can apply **row-column** encoding; i.e., every column of $U(z)$ is encoded using $G_v(z)$, and then every row of the resulting matrix $G_v(z)U(z)$ is encoded using $G_h(z)$ as $(G_v(z)U(z))G_h(z)^T$. We can also apply **column-row** encoding; i.e., every row of $U(z)$ is encoded using $G_h(z)$, and then every column of the resulting matrix $U(z)G_h(z)^T$ is encoded using $G_v(z)$ as $G_v(z)(U(z)G_h(z)^T)$. As a consequence of the associativity of the product of matrices, we get the same matrix in both cases. Thus, the codeword matrix $V(z)$ is given by

$$V(z) = G_v(z)\, U(z)\, G_h(z)^T,$$

and by using properties of the Kronecker product (see [50,51]), we have

$$\text{vect}(V(z)) = (G_h(z) \otimes G_v(z))\, \text{vect}(U(z))$$

where $\text{vect}(\cdot)$ is the operator that transforms a matrix into a vector by stacking the column vectors of the matrix below one another. Now, since

$$G(z) = G_h(z) \otimes G_v(z) \in \mathbb{F}[z]^{n_h h_v \times k_h k_v}$$

and $\text{rank}(G(z)) = \text{rank}(G_h(z))\, \text{rank}(G_v(z)) = k_h\, k_v$, it follows that $G(z)$ is a generator matrix of the product convolutional code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$. Note that $\mathcal{C}$ is a rate $k_h k_v / n_h n_v$ convolutional code. We will compute its degree in Theorem 5 below.

The following two theorems were introduced in [22,49] without proof. We include them here, with proof, for completeness and further references. The first one establishes that the generator matrix of the product code is basic if the generator matrices of the constituent codes are also basic.

Theorem 2. *Assume that $G_h(z) \in \mathbb{F}[z]^{n_h \times k_h}$ and $G_v(z) \in \mathbb{F}[z]^{n_v \times k_v}$ are generator matrices of the horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v) convolutional codes C_h and C_v, respectively. If $G_h(z)$ and $G_v(z)$ are basic, then $G(z) = G_h(z) \otimes G_v(z)$ is basic.*

Proof. Since $G_h(z)$ and $G_v(z)$ are basic matrices, there exist $L_h(z) \in \mathbb{F}[z]^{k_h \times n_h}$ and $L_v(z) \in \mathbb{F}[z]^{k_v \times n_v}$ such that $L_h(z)G_h(z) = I_{k_h}$ and $L_v(z)G_v(z) = I_{k_v}$. Now, consider the polynomial matrix $L(z) = L_h(z) \otimes L_v(z) \in \mathbb{F}[z]^{k_h k_v \times n_h n_v}$. From the properties of the Kronecker product, it follows that $L(z)G(z) = I_{k_h k_v}$. Consequently, $G(z)$ is basic. □

The next theorem gives us the column degrees of a generator matrix of the product code as a function of the column degrees of the generator matrices of the constituent codes.

Theorem 3. *Assume that $G_h(z) \in \mathbb{F}[z]^{n_h \times k_h}$ and $G_v(z) \in \mathbb{F}[z]^{n_v \times k_v}$ are generator matrices of the horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v) convolutional codes C_h and C_v, respectively. If $\nu_1^{(h)}, \nu_2^{(h)}, \ldots, \nu_{k_h}^{(h)}$, and $\nu_1^{(v)}, \nu_2^{(v)}, \ldots, \nu_{k_v}^{(v)}$, are the column degrees of $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively, then the column degrees of $G(z) = G_h(z) \otimes G_v(z)$ are*

$$\nu_1, \nu_2, \ldots, \nu_{k_v}, \nu_{k_v+1}, \nu_{k_v+2}, \ldots, \nu_{2k_v}, \nu_{2k_v+1}, \ldots, \nu_{(k_h-1)k_v+1}, \nu_{(k_h-1)k_v+2}, \ldots, \nu_{k_h k_v},$$

with $\nu_l = \nu_i^{(h)} + \nu_j^{(v)}$, with $l = (i-1)k_v + j$ and $j = 1, 2, \ldots, k_v$.

Proof. Assume that

$$G_h(z) = \begin{bmatrix} g_1^{(h)}(z) & g_2^{(h)}(z) & \cdots & g_{k_h}^{(h)}(z) \end{bmatrix}$$
$$G_v(z) = \begin{bmatrix} g_1^{(v)}(z) & g_2^{(v)}(z) & \cdots & g_{k_v}^{(v)}(z) \end{bmatrix}.$$

From the properties of the Kronecker product, it follows that

$$G(z) = \begin{bmatrix} M_1 & M_2 & \cdots & M_{k_h} \end{bmatrix}$$

where

$$M_i = \begin{bmatrix} g_i^{(h)}(z) \otimes g_1^{(v)}(z) & g_i^{(h)}(z) \otimes g_2^{(v)}(z) & \cdots & g_i^{(h)}(z) \otimes g_{k_v}^{(v)}(z) \end{bmatrix}, \quad \text{for } i = 1, 2, \ldots, k_h.$$

Now, since the column degrees of $g_i^{(h)}(z)$ and $g_j^{(v)}(z)$ are $\nu_i^{(h)}$ and $\nu_j^{(v)}$, respectively, it follows that the column degree of $g_i^{(h)}(z) \otimes g_j^{(v)}(z)$ is $\nu_i^{(h)} + \nu_j^{(v)}$, and the theorem holds. □

As an immediate consequence of the previous theorem, we have the following theorem:

Theorem 4. *Assume that $G_h(z) \in \mathbb{F}[z]^{n_h \times k_h}$ and $G_v(z) \in \mathbb{F}[z]^{n_v \times k_v}$. If $G_h(z)$ and $G_v(z)$ are column reduced, then $G(z) = G_h(z) \otimes G_v(z)$ is column reduced.*

Proof. Let $G_h^{(\infty)}$ and $G_v^{(\infty)}$ the high-order coefficient matrices of $G_h(z)$ and $G_v(z)$, respectively. If G_∞ is the high-order coefficient matrix of $G(z)$, from Theorem 3, it follows that

$$G_\infty = G_h^{(\infty)} \otimes G_v^{(\infty)},$$

and, from the properties of the Kronecker product,

$$\operatorname{rank}(G_\infty) = \operatorname{rank}\left(G_h^{(\infty)} \otimes G_v^{(\infty)}\right) = \operatorname{rank}\left(G_h^{(\infty)}\right) \operatorname{rank}\left(G_v^{(\infty)}\right) = k_h\, k_v.$$

Therefore, $G(z)$ is column reduced. □

Finally, as a consequence of Theorems 2 and 4, we obtain the following theorem that gives us the degree of the product code as a function of the degrees of the constituent codes.

Theorem 5. *Assume that C_h and C_v are horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v) convolutional codes, respectively. Then, the degree of $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$ is $\delta_h k_v + k_h \delta_v$.*

Proof. Assume that $G_h(z) \in \mathbb{F}[z]^{n_h \times k_h}$ and $G_v(z) \in \mathbb{F}[z]^{n_v \times k_v}$ are basic and column reduced generator matrices of $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively. With the notation of Theorem 3, $\nu_1^{(h)}, \nu_2^{(h)}, \ldots, \nu_{k_h}^{(h)}$ and $\nu_1^{(v)}, \nu_2^{(v)}, \ldots, \nu_{k_v}^{(v)}$, are the Forney indices of $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively, and, therefore, $\delta_h = \sum_{i=1}^{k_h} \nu_i^{(h)}$ and $\delta_v = \sum_{j=1}^{k_v} \nu_j^{(v)}$. Moreover, from Theorems 2 and 4, $G(z) = G_h(z) \otimes G_v(z)$ is a basic and column reduced generator matrix for $\mathcal{C}$. Again, with the notation of Theorem 3, $\nu_{(i-1)k_v+j} = \nu_i^{(h)} + \nu_j^{(v)}$, for $i = 1, 2, \ldots, k_h$ and $j = 1, 2, \ldots, k_v$, are the Forney indices of $\mathcal{C}$, and, therefore,

$$\sum_{i=1}^{k_h} \sum_{j=1}^{k_v} \left(\nu_i^{(h)} + \nu_j^{(v)}\right) = \sum_{i=1}^{k_h} \left(\nu_i^{(h)} k_v + \delta_v\right) = \delta_h k_v + k_h \delta_v$$

is the degree of $\mathcal{C}$. □

We will use the above theorems in the next section to obtain a minimal state-space realization of the product convolutional code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$.

4. State-Space Realizations of Product Convolutional Codes

More specifically, let us assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal realizations of column reduced generator matrices of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively. In this section, we will obtain a minimal state-space realization (A, B, C, D) of the (n, k, δ) product convolutional code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$, where $n = n_h n_v$, $k = k_h k_v$ and $\delta = \delta_h k_v + k_h \delta_v$. This means that we must find matrices $A \in \mathbb{F}^{\delta \times \delta}$, $B \in \mathbb{F}^{\delta \times k}$, $C \in \mathbb{F}^{n \times \delta}$ and $D \in \mathbb{F}^{n \times k}$, such that the pair (A, B) is reachable, the pair (A, C) is observable, and $C(I_\delta - zA)^{-1}Bz + D$ is a basic and column reduced generator matrix for $\mathcal{C}$.

We can assume, without loss of generality, that matrices A_h, A_v, and B_h, B_v have the form of matrices A and B in Theorem 1. That is,

$$A_h = \begin{bmatrix} A_1^{(h)} & & & \\ & A_2^{(h)} & & \\ & & \ddots & \\ & & & A_{k_h}^{(h)} \end{bmatrix} \quad \text{with} \quad A_i^{(h)} = \begin{bmatrix} \mathbf{0}^T & 0 \\ I_{\nu_i^{(h)}-1} & \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_i^{(h)} \times \nu_i^{(h)}}, \tag{4}$$

$$B_h = \begin{bmatrix} B_1^{(h)} & & & \\ & B_2^{(h)} & & \\ & & \ddots & \\ & & & B_{k_h}^{(h)} \end{bmatrix} \quad \text{with} \quad B_i^{(h)} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_i^{(h)}}, \tag{5}$$

$$A_v = \begin{bmatrix} A_1^{(v)} & & & \\ & A_2^{(v)} & & \\ & & \ddots & \\ & & & A_{k_v}^{(v)} \end{bmatrix} \quad \text{with} \quad A_j^{(v)} = \begin{bmatrix} \mathbf{0}^T & 0 \\ I_{\nu_j^{(v)}-1} & \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_j^{(h)} \times \nu_j^{(h)}}, \tag{6}$$

$$B_v = \begin{bmatrix} B_1^{(v)} & & & \\ & B_2^{(v)} & & \\ & & \ddots & \\ & & & B_{k_v}^{(v)} \end{bmatrix} \quad \text{with} \quad B_j^{(v)} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_j^{(v)}}. \tag{7}$$

The next theorem allows us to obtain a reachable pair (A, B) from the reachable pairs (A_h, B_h) and (A_v, B_v).

Theorem 6. *Assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal state-space realizations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively, with A_h, B_h, A_v, and B_v as in expressions (4)–(7). For $i = 1, 2, \ldots, k_h$ and $j = 1, 2, \ldots, k_v$, let $\nu_{(i-1)k_v+j} = \nu_i^{(h)} + \nu_j^{(v)}$ and consider*

$$A_{(i-1)k_v+j} = \left[\begin{array}{cc|cc} \mathbf{0}^T & 0 & & \\ I_{\nu_i^{(h)}-1} & \mathbf{0} & & \\ \hline & 1 & \mathbf{0}^T & 0 \\ & & I_{\nu_j^{(v)}-1} & \mathbf{0} \end{array}\right] = \begin{bmatrix} \mathbf{0}^T & 0 \\ I_{\nu_{(i-1)k_v+j}-1} & \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_{(i-1)k_v+j} \times \nu_{(i-1)k_v+j}},$$

$$B_{(i-1)k_v+j} = \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix} \in \mathbb{F}^{\nu_{(i-1)k_v+j}},$$

and define

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_k \end{bmatrix} \in \mathbb{F}^{\delta \times \delta}, \quad B = \begin{bmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_k \end{bmatrix} \in \mathbb{F}^{\delta \times k}.$$

Then, (A, B) *is a reachable par.*

Proof. It is easy to see that $\mathrm{rank}\left(\begin{bmatrix} \lambda I_\delta - A & B \end{bmatrix}\right) = \delta$, for all $\lambda \in \bar{\mathbb{F}}$. Thus, the pair (A, B) is reachable. □

Assume again that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal state-space realizations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively, with A_h, B_h, A_v, and B_v as in expressions (4)–(7). From Theorem 1 and expressions (1) and (3), it follows that

$$G_h(z) = C_h E_h(z) + D_h \quad \text{and} \quad G_v(z) = C_v E_v(z) + D_v \tag{8}$$

where

$$E_h(z) = \begin{bmatrix} E_1^{(h)}(z) & & & \\ & E_2^{(h)}(z) & & \\ & & \ddots & \\ & & & E_{k_h}^{(h)}(z) \end{bmatrix} \quad \text{with} \quad E_i^{(h)}(z) = \begin{bmatrix} z \\ z^2 \\ \vdots \\ z^{\nu_i^{(h)}} \end{bmatrix}, \text{ for } i = 1, 2, \ldots, k_h. \tag{9}$$

$$E_v(z) = \begin{bmatrix} E_1^{(v)}(z) & & & \\ & E_2^{(v)}(z) & & \\ & & \ddots & \\ & & & E_{k_v}^{(v)}(z) \end{bmatrix} \quad \text{with} \quad E_j^{(v)}(z) = \begin{bmatrix} z \\ z^2 \\ \vdots \\ z^{\nu_j^{(v)}} \end{bmatrix}, \text{ for } j = 1, 2, \ldots, k_v. \tag{10}$$

Now, since $G(z) = G_h(z) \otimes G_v(z)$, from expression (8) and the properties of the Kronecker product, we have that

$$\begin{aligned} G(z) &= (C_h E_h(z) + D_h) \otimes (C_v E_v(z) + D_v) \\ &= (C_h \otimes C_v)(E_h(z) \otimes E_v(z)) + (C_h \otimes D_v)(E_h(z) \otimes I_{k_v}) \\ &\qquad + (D_h \otimes C_v)\left(I_{k_h} \otimes E_v(z)\right) + D_h \otimes D_v \\ &= \begin{bmatrix} C_h \otimes C_v & C_h \otimes D_v & D_h \otimes C_v \end{bmatrix} \begin{bmatrix} E_h(z) \otimes E_v(z) \\ E_h(z) \otimes I_{k_v} \\ I_{k_h} \otimes E_v(z) \end{bmatrix} + D_h \otimes D_v \\ &= \bar{C}\bar{E}(z) + \bar{D}. \end{aligned} \tag{11}$$

Note that $\bar{D} = D_h \otimes D_v$ is a matrix of size $n_h n_v \times k_h k_v$; that is, $n \times k$. Thus, we can take $D = \bar{D}$. However, since $\bar{C} = \begin{bmatrix} C_h \otimes C_v & C_h \otimes D_v & D_h \otimes C_v \end{bmatrix}$ is a matrix of size $n_h n_v \times (\delta_h \delta_v + \delta_h k_v + k_h \delta_v)$, that is, $n \times (\delta_h \delta_v + \delta)$, we cannot take the above matrix as matrix C. The following example will help us to understand how we should proceed to obtain the matrix C from the matrix $\bar{C}$ in expression (11).

Example 2. *Let $\mathbb{F} = GF(2)$ be the Galois field of two elements and consider $G_h(z)$, the column reduced matrix, and the minimal state-space realization (A_h, B_h, C_h, D_h) of $G_h(z)$ given in Example 1. That is,* $G_h(z) = \begin{bmatrix} z^2 & z+1 \\ z+1 & z \\ 1 & 1 \end{bmatrix} \in \mathbb{F}[z]^{3\times 2}$ *and*

$$A_h = \left[\begin{array}{cc|c} 0 & 0 & \\ 1 & 0 & \\ \hline & & 0 \end{array}\right], \quad B_h = \left[\begin{array}{c|c} 1 & \\ 0 & \\ \hline & 1 \end{array}\right], \quad C_h = \left[\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right], \quad \text{and} \quad D_h = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Moreover, $E_h(z) = \begin{bmatrix} z & 0 \\ z^2 & 0 \\ 0 & z \end{bmatrix}$.

Let $G_v(z) = \begin{bmatrix} 1+z+z^2 & 1+z \\ z & 1 \\ 1+z^3 & z \\ 1 & 1+z^2 \end{bmatrix} \in \mathbb{F}[z]^{4\times 2}$. *Since* $\nu_1^{(v)} = 3$, $\nu_2^{(v)} = 2$, *and* $\operatorname{rank}\left(G_v^{(\infty)}\right) = 2$, *it follows that $G_v(z)$ is column reduced. Now, consider the matrices*

$$A_v = \left[\begin{array}{ccc|cc} 0 & 0 & 0 & & \\ 1 & 0 & 0 & & \\ 0 & 1 & 0 & & \\ \hline & & & 0 & 0 \\ & & & 1 & 0 \end{array}\right], \quad B_v = \left[\begin{array}{c|c} 1 & \\ 0 & \\ 0 & \\ \hline & 1 \\ & 0 \end{array}\right],$$

$$C_v = \left[\begin{array}{ccc|cc} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right], \quad \text{and} \quad D_v = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}.$$

Moreover, $E_v(z) = \begin{bmatrix} z & 0 \\ z^2 & 0 \\ z^3 & 0 \\ 0 & z \\ 0 & z^2 \end{bmatrix}$.

Now, from expression (11), the generator matrix $G(z) = G_h(z) \otimes G_v(z)$ of the product convolutional code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$ is given by

$$G(z) = \bar{C}\bar{E}(z) + \bar{D}$$

with

$$\bar{C} = \begin{bmatrix} C_h \otimes C_v & C_h \otimes D_v & D_h \otimes C_v \end{bmatrix}, \quad \bar{E}(z) = \begin{bmatrix} E_h(z) \otimes E_v(z) \\ E_h(z) \otimes I_{k_v} \\ I_{k_h} \otimes E_v(z) \end{bmatrix}, \quad \text{and} \quad \bar{D} = D_h \otimes D_v,$$

where

$$
C_h \otimes C_v = \left[\begin{array}{ccccc:ccccc:ccccc}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hdashline
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right],
$$

$$
C_h \otimes D_v = \left[\begin{array}{cc:cc:cc}
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\ \hdashline
1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 \\ \hdashline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{array}\right], \quad
D_h \otimes C_v = \left[\begin{array}{ccccc:ccccc}
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hdashline
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hdashline
1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\right],
$$

$$
E_h(z) \otimes E_v(z) = \left[\begin{array}{cc:cc}
z^2 & 0 & & \\
z^3 & 0 & & \\
z^4 & 0 & & \\
0 & z^2 & & \\
0 & z^3 & & \\ \hdashline
z^3 & 0 & & \\
z^4 & 0 & & \\
z^5 & 0 & & \\
0 & z^3 & & \\
0 & z^4 & & \\ \hdashline
 & & z^2 & 0 \\
 & & z^3 & 0 \\
 & & z^4 & 0 \\
 & & 0 & z^2 \\
 & & 0 & z^3
\end{array}\right],
$$

$$
E_h(z) \otimes I_{k_v} = \left[\begin{array}{cc:cc}
z & 0 & & \\
0 & z & & \\ \hdashline
z^2 & 0 & & \\
0 & z^2 & & \\ \hdashline
 & & z & 0 \\
 & & 0 & z
\end{array}\right], \quad
I_{k_h} \otimes E_v(z) = \left[\begin{array}{cc:cc}
z & 0 & & \\
z^2 & 0 & & \\
z^3 & 0 & & \\
0 & z & & \\
0 & z^2 & & \\ \hdashline
 & & z & 0 \\
 & & z^2 & 0 \\
 & & z^3 & 0 \\
 & & 0 & z \\
 & & 0 & z^2
\end{array}\right].
$$

As we can observe, $\bar{C}$ *has* 31 *columns, but we need a matrix with* 16 *columns. Furthermore,* $\bar{E}(z)$ *does not have the structure given by expression* (3)*.*

However, considering the rows of $\bar{E}(z)$ whose elements have been written in red, we can move these rows to the appropriate positions and then, by Gaussian elimination from those rows, we can transform the matrix $\bar{E}(z)$ into the matrix $\begin{bmatrix} E(z) \\ O \end{bmatrix}$, with

$$E(z) = \begin{bmatrix} z & 0 & 0 & 0 \\ z^2 & 0 & 0 & 0 \\ z^3 & 0 & 0 & 0 \\ z^4 & 0 & 0 & 0 \\ z^5 & 0 & 0 & 0 \\ 0 & z & 0 & 0 \\ 0 & z^2 & 0 & 0 \\ 0 & z^3 & 0 & 0 \\ 0 & z^4 & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & z^2 & 0 \\ 0 & 0 & z^3 & 0 \\ 0 & 0 & z^4 & 0 \\ 0 & 0 & 0 & z \\ 0 & 0 & 0 & z^2 \\ 0 & 0 & 0 & z^3 \end{bmatrix}$$

and O the zero matrix of the appropriate size. This means that we can find an invertible matrix $P \in \mathbb{F}^{31\times 31}$ such that

$$P\bar{E}(z) = \begin{bmatrix} E(z) \\ O \end{bmatrix}$$

and, therefore $\bar{C}\bar{E}(z) = CE(z)$, with $C \in \mathbb{F}^{12\times 16}$ such that $\bar{C}P^{-1} = \begin{bmatrix} C & \widetilde{C} \end{bmatrix}$.

We can use the argument introduced in the above example to prove the following theorem.

Theorem 7. *Assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal state-space realizations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathcal{C}_h$ and $\mathcal{C}_v$, respectively, with A_h, B_h, A_v, and B_v as in expressions (4)–(7). Let A be the matrix defined in Theorem 6 and let $\bar{C}$ be the matrix in expression (11). Moreover, assume that*

$$E(z) = \begin{bmatrix} E_1(z) & & & \\ & E_2(z) & & \\ & & \ddots & \\ & & & E_k(z) \end{bmatrix} \quad \text{with} \quad E_\ell(z) = \begin{bmatrix} z \\ z^2 \\ \vdots \\ z^{\nu_\ell} \end{bmatrix}, \text{ for } \ell = 1, 2, \dots, k,$$

where $\nu_\ell = \nu_i^{(h)} + \nu_j^{(v)}$, with $\ell = (i-1)k_v + j$, for $i = 1, 2, \dots, k_h$ and $j = 1, 2, \dots, k_v$, and consider the matrices $E_h(z)$ and $E_v(z)$ in expressions (9) and (10). If $\bar{E}(z) = \begin{bmatrix} E_h(z) \otimes E_v(z) \\ E_h(z) \otimes I_{k_v} \\ I_{k_h} \otimes E_v(z) \end{bmatrix}$, then there exists an invertible matrix $P \in \mathbb{F}^{(\delta+\delta_h\delta_v)\times(\delta+\delta_h\delta_v)}$ such that

$$P\bar{E}(z) = \begin{bmatrix} E(z) \\ O \end{bmatrix}.$$

Moreover, if $\bar{C}P^{-1} = \begin{bmatrix} C & \widetilde{C} \end{bmatrix}$, with $C \in \mathbb{F}^{n\times\delta}$, then the pair (A, C) is observable.

Proof. Note that the submatrix of $\bar{E}(z)$ given by

$$\widehat{E}(z) = \left[\begin{array}{cccc} \multicolumn{4}{c}{E_h(z) \otimes I_{k_v}} \\ \hdashline z^{\nu_1^{(h)}} E_v(z) & O & \cdots & O \\ O & z^{\nu_2^{(h)}} E_v(z) & \cdots & O \\ \vdots & \vdots & & \vdots \\ O & O & \cdots & z^{\nu_{k_h}^{(h)}} E_v(z) \end{array} \right] \quad (12)$$

contains the necessary rows to construct the matrix $E(z)$. Thus, by using an appropriate permutation matrix $Q \in \mathbb{F}^{(\delta+\delta_h\delta_v)\times(\delta+\delta_h\delta_v)}$, we have that

$$Q\bar{E}(z) = \begin{bmatrix} E(z) \\ \widetilde{E}(z) \end{bmatrix}.$$

Now, the entries in the first column of $\widetilde{E}(z)$ are 0 or z^t with $1 \le t \le \nu_1^{(h)} + \nu_1^{(v)} - 1$; therefore, by using Gaussian elimination, we can transform these entries in 0. Once this operation is completed, the entries in the second column of the modified $\widetilde{E}(z)$ are, again, 0 or z^t with $1 \le t \le \nu_1^{(h)} + \nu_2^{(v)} - 1$ and, therefore, we can transform these entries in 0. We continue with this argument, until we transform matrix $\widetilde{E}(z)$ into the zero matrix. In other words, we have found an invertible matrix $R \in \mathbb{F}^{(\delta+\delta_h\delta_v)\times(\delta+\delta_h\delta_v)}$ such that

$$R \begin{bmatrix} E(z) \\ \widetilde{E}(z) \end{bmatrix} = \begin{bmatrix} E(z) \\ O \end{bmatrix}.$$

Thus, we can take $P = RQ$ and, from expression (11), it follows that $\bar{C}\bar{E}(z) = CE(z)$.

Now, by a similar argument to the argument used in the proof of Theorem 1, it follows that the pair (A, C) is observable. □

The proof of the previous theorem tells us which are the rows of matrix $\bar{E}(z)$ that we must consider to obtain matrix $E(z)$. Therefore, it also tells us which are the columns of matrix $\bar{C}$ that we must consider. Specifically, the submatrix $\widehat{E}(z)$ given in expression (12) will help us to determine a submatrix of $\bar{C}$, which contains the necessary columns to construct the matrix C. For that, on the one hand, the block $E_h(z) \otimes I_{k_v}$ of $\widehat{E}(z)$ means that we take all the columns of $C_h \otimes D_v$. On the other hand, if we assume that $C_h = \begin{bmatrix} C_1^{(h)} & C_2^{(h)} & \cdots & C_{k_h}^{(h)} \end{bmatrix}$, with

$$C_i^{(h)} = \begin{bmatrix} \left(g_i^{(h)}\right)^{(1)} & \left(g_i^{(h)}\right)^{(2)} & \cdots & \left(g_i^{(h)}\right)^{\left(\nu_i^{(h)}\right)} \end{bmatrix}, \quad \text{for } i = 1, 2, \ldots, k_h,$$

then, from the properties of the Kronecker product,

$$C_h \times C_v = \begin{bmatrix} C_1^{(h)} \otimes C_v & C_2^{(h)} \otimes C_v & \cdots & C_{k_h}^{(h)} \otimes C_v \end{bmatrix}$$

with

$$C_i^{(h)} \otimes C_v = \begin{bmatrix} \left(g_i^{(h)}\right)^{(1)} \otimes C_v & \left(g_i^{(h)}\right)^{(2)} \otimes C_v & \cdots & \left(g_i^{(h)}\right)^{\left(\nu_i^{(h)}\right)} \otimes C_v \end{bmatrix}, \quad \text{for } i = 1, 2, \ldots, k_h.$$

Therefore, the rest of the rows of matrix $\widehat{E}(z)$ in expression (12) means that we must take the columns $\left(g_i^{(h)}\right)^{\left(\nu_i^{(h)}\right)} \otimes C_v$, for $i = 1, 2, \ldots, k_h$. Thus, by using the matrix P^{-1}, we have that

$$
\begin{aligned}
\bar{C}P^{-1} &= \left[\begin{array}{c|c|c|c|c} \cdots \quad \left(g_1^{(h)}\right)^{\nu_1^{(h)}} \otimes C_v & \cdots \quad \left(g_2^{(h)}\right)^{\nu_2^{(h)}} \otimes C_v & \cdots \quad \left(g_{k_h}^{(h)}\right)^{\nu_{k_h}^{(h)}} \otimes C_v & C_h \otimes D_v & D_h \otimes C_v \end{array} \right] P^{-1} \\
&= \left[\begin{array}{cccc|cccc|c|cccc|c} c_1^{(1)} & c_1^{(2)} & \cdots & c_1^{(\nu_1)} & c_2^{(1)} & c_2^{(2)} & \cdots & c_2^{(\nu_2)} & \cdots & c_k^{(1)} & c_k^{(2)} & \cdots & c_k^{(\nu_k)} & \widetilde{C} \end{array} \right] \\
&= \left[\begin{array}{cc} C & \widetilde{C} \end{array} \right],
\end{aligned}
$$

with $k = k_h k_v$ and ν_ℓ as in Theorem 7.

Now, as a consequence of Theorems 6 and 7, we obtain a minimal state-space realization of the convolutional product code.

Corollary 1. *With the notation of Theorems 6 and 7, the system (A, B, C, D), with $D = D_h \otimes D_v$, is a minimal realization of the convolutional product code $\mathcal{C} = \mathcal{C}_h \otimes \mathcal{C}_v$.*

Example 3. *For the matrices in Example 2, it follows that*

$$
\bar{C}P^{-1} = \left[\begin{array}{c|ccccc|ccccc|cccccc|c}
 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \\
 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \\
 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & \\ \cline{2-17}
 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\
\cdots & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \cdots \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & \\ \cline{2-17}
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 &
\end{array} \right]
$$

$$
= \left[\begin{array}{cccccccccccccccc|c}
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & \cdots \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & \\
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & \\
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 &
\end{array} \right] = \left[\begin{array}{c|c} C & \widetilde{C} \end{array} \right]
$$

5. Conclusions and Future Work

In this paper, we presented a constructive methodology to obtain a minimal state-space representation (A, B, C, D) of a convolutional product code from two minimal state-space representations, (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) of an horizontal and a vertical convolutional code, respectively. In this work, we have considered driven variable representations and showed that, even if the matrices A, B, and D of the product convolutional code can be built in a straightforward way from the given matrix representations (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v), the matrix C requires further analysis. We showed, however, that C can still be computed if one properly selects the appropriate entries of a matrix that depends on C_h, C_v, D_h and D_v. In this way, the produced representation is minimal and can be computed in a relatively easy way.

An interesting line for future research would be to consider input–state–output representations instead of driven variables and study these different state space representations in the context of convolutional product codes.

Author Contributions: Investigation, J.-J.C., D.N., R.P. and V.R.; writing–original draft, J.-J.C., D.N., R.P. and V.R.; writing–review and editing, J.-J.C., D.N., R.P. and V.R. All authors have read and agreed to the published version of the manuscript.

Funding: The research of the first, second, and fourth authors was supported by Spanish grants PID2019-108668GB-I00 of the Ministerio de Ciencia e Innovación of the Gobierno de España and VIGROB-287 of the Universitat d'Alacant. The research of the third author was supported by The Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT—Fundação para a Ciência e a Tecnologia), references UIDB/04106/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Blaum, M.; Brady, J.; Bruck, J.; Menon, J. EVENODD: An efficient scheme for tolerating double disk failures in RAID architectures. *IEEE Trans. Comput.* **1995**, *42*, 192–202. [CrossRef]
2. Blaum, M.; Roth, R.M. New array codes for multiple phased burst correction. *IEEE Trans. Inf. Theory* **1993**, *39*, 66–77. [CrossRef]
3. Cardell, S.D.; Climent, J.J. An approach to the performance of SPC product codes under the erasure channel. *Adv. Math. Commun.* **2016**, *10*, 11–28. [CrossRef]
4. Climent, J.J.; Napp, D.; Pinto, R.; Simões, R. Series concatenation of 2D convolutional codes by means of input–state–output representations. *Int. J. Control* **2018**, *91*, 2682–2691. [CrossRef]
5. DeCastro-García, N.; García-Planas, M. Concatenated linear systems over rings and their application to construction of concatenated families of convolutional codes. *Linear Algebra Its Appl.* **2018**, *542*, 624–647. [CrossRef]
6. Elias, P. Error-free coding. *Trans. Ire Prof. Group Inf. Theory* **1954**, *4*, 29–37. [CrossRef]
7. Napp, D.; Pinto, R.; Sidorenko, V. Concatenation of convolutional codes and rank metric codes for multi-shot network coding. *Des. Codes Cryptogr.* **2018**, *86*, 237–445. [CrossRef]
8. Sidorenko, V.; Jiang, L.; Bossert, M. Skew-feedback shift-register synthesis and decoding interleaved Gabidulin codes. *IEEE Trans. Inf. Theory* **2011**, *57*, 621–632. [CrossRef]
9. Climent, J.J.; Herranz, V.; Perea, C. Linear system modelization of concatenated block and convolutional codes. *Linear Algebra Its Appl.* **2008**, *429*, 1191–1212. [CrossRef]
10. Climent, J.J.; Herranz, V.; Perea, C. Parallel concatenated convolutional codes from linear systems theory viewpoint. *Syst. Control Lett.* **2016**, *96*, 15–22. [CrossRef]
11. Fornasini, E.; Pinto, R. Matrix fraction descriptions in convolutional codes. *Linear Algebra Its Appl.* **2004**, *392*, 119–158. [CrossRef]
12. Forney, G.D., Jr. Minimal bases of rational vector spaces, with applications to multivariable linear systems. *SIAM J. Control* **1975**, *13*, 493–520. [CrossRef]
13. Forney, G.D., Jr.; Johannesson, R.; Wan, Z.X. Minimal and canonical rational generator matrices for convolutional codes. *IEEE Trans. Inf. Theory* **1996**, *42*, 1865–1880. [CrossRef]
14. Gluesing-Luerssen, H.; Schneider, G. State space realizations and monomial equivalence for convolutional codes. *Linear Algebra Its Appl.* **2007**, *425*, 518–533. [CrossRef]
15. Herranz, V.; Napp, D.; Perea, C. $1/n$ turbo codes from linear system point of view. *Rev. Real Acad. Cienc. Exactas Fis. Nat. Ser. A. Mat.* **2020**, *114*. [CrossRef]
16. Massey, J.L.; Sain, M.K. Codes, automata, and continuous systems: Explicit interconnections. *IEEE Trans. Autom. Control* **1967**, *12*, 644–650. [CrossRef]
17. McEliece, R.J. The algebraic theory of convolutional codes. In *Handbook of Coding Theory*; Pless, V.S., Huffman, W.C., Eds.; Elsevier: North-Holland, The Netherlands, 1998; pp. 1065–1138.
18. Rosenthal, J. Connections between linear systems and convolutional codes. In *Codes, Systems and Graphical Models*; Marcus, B., Rosenthal, J., Eds.; Springer: New York, NY, USA, 2001; Volume 123, The IMA Volumes in Mathematics and its Applications, pp. 39–66. [CrossRef]
19. Rosenthal, J. Some interesting problems in systems theory which are of fundamental importance in coding theory. In Proceedings of the IEEE Conference on Decision and Control, San Diego, CA, USA, 12 December 1997; pp. 1–6.

20. Rosenthal, J.; Schumacher, J.M.; York, E.V. On behaviors and convolutional codes. *IEEE Trans. Inf. Theory* **1996**, *42*, 1881–1891. [CrossRef]
21. Rosenthal, J.; York, E.V. BCH convolutional codes. *IEEE Trans. Inf. Theory* **1999**, *45*, 1833–1844. [CrossRef]
22. Bossert, M.; Medina, C.; Sidorenko, V. Encoding and distance estimation of product convolutional codes. In Proceedings of the 2005 IEEE International Symposium on Information Theory (ISIT 2005), Adelaide, SA, Australia, 4–9 September 2005; pp. 1063–1066. [CrossRef]
23. Höst, S.; Johannesson, R.; Sidorenko, V.; Zigangirov, K.S.; Zyablov, V.V. Woven convolutional codes I: Encoder properties. *IEEE Trans. Inf. Theory* **2002**, *48*, 149–161. [CrossRef]
24. Rosenthal, J. An algebraic decoding algorithm for convolutional codes. *Prog. Syst. Control Theory* **1999**, *25*, 343–360. [CrossRef]
25. Lieb, J.; Rosenthal, J. Erasure decoding of convolutional codes using first order representations. *Math. Control. Signals Syst.* **2021**, 1–15. [CrossRef]
26. Muñoz Castañeda, A.L.; Muñoz-Porras, J.M.; Plaza-Martín, F.J. Rosenthal's decoding algorithm for certain 1-dimensional convolutional codes. *IEEE Trans. Inf. Theory* **2019**, *65*, 7736–7741. [CrossRef]
27. Climent, J.J.; Herranz, V.; Perea, C. input–state–output representation of convolutional product codes. In *Coding Theory and Applications—Proceedings of the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA)*; CIM Series in Mathematical Sciences; Pinto, R., Rocha Malonek, P., Vettori, P., Eds.; Springer: Berlin, Germany, 2015; Volume 3, pp. 107–114. [CrossRef]
28. Fuhrmann, P.A.; Helmke, U. *The Mathematics of Networks of Linear Systems*; Springer International Publishing AG: Cham, Switzerland, 2015.
29. Kailath, T. *Linear Systems*; Prentice-Hall: Upper Saddle River, NJ, USA, 1980.
30. Forney, G.D., Jr. Convolutional codes I: Algebraic structure. *IEEE Trans. Inf. Theory* **1970**, *16*, 720–738. [CrossRef]
31. Johannesson, R.; Wan, Z.X. A linear algebra approach to minimal convolutional encoders. *IEEE Trans. Inf. Theory* **1993**, *39*, 1219–1233. [CrossRef]
32. Johannesson, R.; Zigangirov, K.S. *Fundamentals of Convolutional Coding*; IEEE Press: New York, NY, USA, 1999.
33. Smarandache, R.; Gluesing-Luerssen, H.; Rosenthal, J. Constructions of MDS-convolutional codes. *IEEE Trans. Inf. Theory* **2001**, *47*, 2045–2049. [CrossRef]
34. Piret, P. *Convolutional Codes, an Algebraic Approach*; MIT Press: Boston, MA, USA, 1988.
35. York, E.V. Algebraic Description and Construction of Error Correcting Codes: A Linear Systems Point of View. Ph.D. Thesis, Department of Mathematics, University of Notre Dame, Notre Dame, IN, USA, 1997.
36. Antsaklis, P.J.; Michel, A.N. *A Linear Systems Primer*; Birkhäuser: Boston, MA, USA, 2007.
37. Chen, C.T. *Linear Systems Theory and Design*, 3rd ed.; Oxford University Press: New York, NY, USA, 1999.
38. Kalman, R.E. Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. Ser. A Control* **1963**, *1*, 152–192. [CrossRef]
39. Hautus, M.L.J. Controllability and observability condition for linear autonomous systems. *Proc. Ned. Akad. Voor Wet. (Ser. A)* **1969**, *72*, 443–448.
40. Kalman, R.E. Lectures on Controllabilitty and Observability. In *Controllabilitty and Observability*; Evangelisti, E., Ed.; Espringer: Berlin, Germangy, 1968; pp. 1–149.
41. Climent, J.J.; Herranz, V.; Perea, C. A first approximation of concatenated convolutional codes from linear systems theory viewpoint. *Linear Algebra Its Appl.* **2007**, *425*, 673–699. [CrossRef]
42. Hutchinson, R.; Rosenthal, J.; Smarandache, R. Convolutional codes with maximum distance profile. *Syst. Control Lett.* **2005**, *54*, 53–63. [CrossRef]
43. Zerz, E. On multidimensional convolutional codes and controllability properties of multidimensional systems over finite rings. *Asian J. Control* **2010**, *12*, 119–126. [CrossRef]
44. Delchamps, D.F. *State Space and Input-Output Linear Systems*; Springer: New York, NY, USA, 1988.
45. de Schutter, B. Minimal state-space realization in linear system theory: An overview. *J. Comput. Appl. Math.* **2000**, *121*, 331–354. [CrossRef]
46. Gilbert, E.G. Controllability and observability in multivariable control systems. *J. Soc. Ind. Appl. Math. Ser. A Control* **1963**, *1*, 128–151. [CrossRef]
47. Kalman, R.E.; Falb, P.L.; Arbib, M.A. *Topics in Mathematical System Theory*; McGraw-Hill: New York, NY, USA, 1969.
48. Rosenthal, J.; Smarandache, R. Construction of convolutional codes using methods from linear systems theory. In Proccedings of the 35th Allerton Conference on Communications, Control and Computing, Monticello, IL, USA, 29 September–1 October 1997; pp. 953–960.
49. Medina, C.; Sidorenko, V.R.; Zyablov, V.V. Error exponents for product convolutional codes. *Probl. Inf. Transm.* **2006**, *42*, 167–182. [CrossRef]
50. Brewer, J.W. Kronecker products and matrix calculus in system theory. *IEEE Trans. Circuits Syst.* **1978**, *cas-25*, 772–781. [CrossRef]
51. Graham, A. *Kronecker Products and Matrix Calculus with Applications*; Ellis Horwood Limited: Chischester, West Sussex, UK, 1981.