Input-State-Output Representation of Convolutional Product Codes

Joan-Josep Climent, Victoria Herranz, and Carmen Perea

Abstract In this paper, we present an input-state-output representation of a convolutional product code; we show that this representation is non minimal. Moreover, we introduce a lower bound on the free distance of the convolutional product code in terms of the free distance of the constituent codes.

Keywords Convolutional code • Product code • ISO representation • Free distance • Kronecker product

1 Introduction

The class of convolutional codes generalizes the class of linear block codes in a natural way. In comparison to the literature on linear block codes, there are only relatively few algebraic constructions of convolutional codes which have a good designed distance. There are several methods for constructing convolutional codes, for example by extending the constructions known for block codes to convolutional codes, such as the ones based on cyclic or quasi-cyclic constructions on block codes [7, 8, 10, 19].

Combining known codes is a powerful method to obtain new codes with better error correction capability avoiding the exponential increase of decoding complexity. For convolutional codes, we can find in the literature some powerful combining methods as woven convolutional codes [21, 22] and turbo codes [18]. More recently, as a natural extension of the direct product codes introduced by Elias [3], Bossert, Medina and Sidorenko [1] introduce the product of convolutional codes

J.-J. Climent (✉)
Departament d'Estadística i Investigació Operativa, Universitat d'Alacant, Ap. Correus 99, E-03080 Alacant, Spain
e-mail: jcliment@ua.es

V. Herranz • C. Perea
Departamento de Estadística, Matemáticas e Informática, Centro de Investigación Operativa, Universidad Miguel Hernández de Elche, Avenida del Ferrocarril, s/n. E-03202 Elche, Spain
e-mail: mavi.herranz@umh.es; perea@umh.es

R. Pinto et al. (eds.), *Coding Theory and Applications*, CIM Series in Mathematical Sciences 3, DOI 10.1007/978-3-319-17296-5_10

and they show that every convolutional product code can be represented as a woven convolutional code (see also [11]).

On the other hand, it is well-known that there exists a close connection between linear systems over finite fields and convolutional codes. Rosenthal [13] provides an excellent survey of the different points of view about convolutional codes. By using the input-state-output representation of convolutional codes introduced by Rosenthal and York [16], Climent, Herranz and Perea [2] and Herrnaz [4] introduce the input-state-output representation of different serial and parallel concatenated convolutional codes, and by using them, they also present a construction of new codes with prescribed distance.

The rest of the paper is structured as follows. In Sect. 2 we present the basic notions and previous results related to convolutional codes and convolutional product codes. Then, in Sect. 3, we introduce two input-state-output representations of a convolutional product code and prove that none of them is minimal. Moreover we introduce a lower bound on the free distance of the convolutional product code.

2 Preliminaries

Let $\mathbb{F}$ be a finite field and denote by $\mathbb{F}[z]$ the polynomial ring on the variable z with coefficients in $\mathbb{F}$. A *convolutional code* $\mathscr{C}$ of rate k/n is a submodule of $\mathbb{F}[z]^n$ that can be described as (see [17, 20])

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

where $\boldsymbol{u}(z)$ is the **information vector**, $\boldsymbol{v}(z)$ is the corresponding **codeword** and $G(z)$ is an $n \times k$ polynomial matrix with rank k called **generator** or **encoder matrix** of $\mathscr{C}$. Two full column rank matrices $G_1(z), G_2(z) \in \mathbb{F}[z]^{n\times k}$ are said to be **equivalent encoders** if and only if there exists a unimodular matrix $P(z) \in \mathbb{F}[z]^{k\times k}$ such that $G_2(z) = G_1(z)P(z)$. The **complexity** of a convolutional code $\mathscr{C}$ is the highest degree of the full size minors of any encoder of $\mathscr{C}$. A generator matrix of a convolutional code is called **minimal** if and only if the complexity is equal to the sum of the column degrees.

A generator matrix is said to be **catastrophic** [6] if there exists some input sequence $\boldsymbol{u}(z)$ with infinite nonzero entries which generates a codeword $\boldsymbol{v}(z) = G(z)\boldsymbol{u}(z)$ with a finite nonzero entries. A convolutional code $\mathscr{C}$ is **observable** if one, and therefore any, generator matrix $G(z)$ is right prime (see [14]). Furthermore, if $G(z)$ is a generator matrix of an observable convolutional code, then $G(z)$ is a noncatastrophic generator matrix (see [14]).

Let $\boldsymbol{v}(z) \in \mathscr{C}$ and assume that $\boldsymbol{v}(z) = \boldsymbol{v}_0 z^\gamma + \boldsymbol{v}_1 z^{\gamma-1} + \cdots + \boldsymbol{v}_{\gamma-1}z + \boldsymbol{v}_\gamma$ with $\boldsymbol{v}_t \in \mathbb{F}^n$, for $t = 0, 1, \ldots, \gamma-1, \gamma$. If we consider $\boldsymbol{v}_t = \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{u}_t \end{pmatrix}$, where $\boldsymbol{y}_t \in \mathbb{F}^{n-k}$ and $\boldsymbol{u}_t \in \mathbb{F}^k$, then the convolutional code $\mathscr{C}$ is equivalently described by the (A, B, C, D)

representation (see [13, 16, 17, 20])

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

For each instant t, we say that $\boldsymbol{x}_t$ is the **state vector**, $\boldsymbol{u}_t$ is the **information vector**, $\boldsymbol{y}_t$ is the **parity vector**, and $\boldsymbol{v}_t$ is the **codeword**. In the linear systems theory, this representation is known as the **input-state-output (ISO) representation**.

If $\mathscr{C}$ is a rate k/n convolutional code with complexity δ, we call $\mathscr{C}$ an (n, k, δ)-code, and in that case, it is possible (see [9]) to choose matrices A, B, C and D of sizes $\delta \times \delta$, $\delta \times k$, $(n-k) \times \delta$ and $(n-k) \times k$, respectively. In convolutional coding theory, an ISO representation (A, B, C, D) having the above sizes is called a **minimal representation** and it is characterized through the condition that the pair (A, B) is **controllable**, that is (see [16]),

$$\operatorname{rank}\begin{pmatrix} B & AB & \cdots & A^{\delta-1}B \end{pmatrix} = \delta.$$

Moreover, if (A, B) is controllable, then the convolutional code defined by the matrices (A, B, C, D) is an observable code if and only if (A, C) is an observable pair (see [12]). Recall that (A, C) is an **observable** pair if (A^T, C^T) is a controllable pair.

The **free distance** of a convolutional code $\mathscr{C}$ can be characterized (see [5]) as

$$d_{free}(\mathscr{C}) = \min\left(\sum_{t=0}^{\infty} \mathrm{wt}(\boldsymbol{u}_t) + \sum_{t=0}^{\infty} \mathrm{wt}(\boldsymbol{y}_t)\right)$$

where the minimum has to be taken over all possible nonzero codewords and where wt denotes the Hamming weight. The free distance of an (n, k, δ)-code $\mathscr{C}$ is always upper-bounded (see [15]) by the **generalized Singleton bound**

$$d_{free}(\mathscr{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1.$$

In addition, the convolutional code $\mathscr{C}$ is called **maximum-distance separable** (MDS) if its free distance is equal to the generalized Singleton bound.

To finish this section, we introduce the product of two convolutional codes called "horizontal" and "vertical" codes respectively. Assume that $\mathscr{C}_h$ and $\mathscr{C}_v$ are horizontal (n_h, k_h, δ_h) and vertical (n_v, k_v, δ_v) codes respectively. Then, the **product convolutional code** (see [1, 11]) $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$ is defined to be the convolutional code whose codewords consist of all $n_v \times n_h$ matrices in which columns belong to $\mathscr{C}_v$ and rows belongs to $\mathscr{C}_h$. It is an $(n_h n_v, k_h k_v, \delta_h k_v + k_h \delta_v)$.

Encoding of the product convolutional code $\mathscr{C}$ can be done as follows (see [1, 11]). Let $G_v(z)$ and $G_h(z)$ be generator matrices of the component convolutional codes $\mathscr{C}_v$ and $\mathscr{C}_h$, respectively. Denote by $U(z)$ a $k_v \times k_h$ 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 encoding 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. So, 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, we have

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

where vect $(\cdot)$ is the operator that transforms a matrix into a vector by stacking the column vectors of the matrix below one another. So, the generator matrix $G(z)$ of the product convolutional code $\mathscr{C}$ is the Kronecker product

$$G(z) = G_h(z) \otimes G_v(z)$$

of the generator matrices of the horizontal and vertical codes.

3 ISO Representation of a Product Convolutional Code

Assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are the ISO representations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathscr{C}_h$ and $\mathscr{C}_v$, respectively. Assume also that the $k_v \times k_h$ matrix U_t is the information matrix of the product code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$.

By using the ISO representation of the horizontal code $\mathscr{C}_h$ we can encode the information vector $\boldsymbol{u}_t = \operatorname{vect}(U_t)$ as

$$\left.\begin{aligned} \boldsymbol{x}_{t+1}^h &= (A_h \otimes I_{k_v})\boldsymbol{x}_t^h + (B_h \otimes I_{k_v})\boldsymbol{u}_t \\ \boldsymbol{y}_t^h &= (C_h \otimes I_{k_v})\boldsymbol{x}_t^h + (D_h \otimes I_{k_v})\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t^h = \begin{pmatrix} \boldsymbol{y}_t^h \\ \boldsymbol{u}_t \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}_0^h = \boldsymbol{0}. \tag{1}$$

Analogously, by using the ISO representation of the vertical code $\mathscr{C}_v$ we can encode the same information vector $\boldsymbol{u}_t$ as

$$\left.\begin{aligned} \boldsymbol{x}_{t+1}^v &= (I_{k_h} \otimes A_v)\boldsymbol{x}_t^v + (I_{k_h} \otimes B_v)\boldsymbol{u}_t \\ \boldsymbol{y}_t^v &= (I_{k_h} \otimes C_v)\boldsymbol{x}_t^v + (I_{k_h} \otimes D_v)\boldsymbol{u}_t \end{aligned}\right\}, \quad \boldsymbol{v}_t^v = \begin{pmatrix} \boldsymbol{y}_t^v \\ \boldsymbol{u}_t \end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \boldsymbol{x}_0^{|} = \boldsymbol{0}, \tag{2}$$

Then we encode the parity vector $\boldsymbol{y}_t^h$ (respectively, $\boldsymbol{y}_t^v$) by using the vertical code $\mathscr{C}_v$ (respectively, the horizontal code $\mathscr{C}_h$) as

$$\left.\begin{array}{rl}\mathfrak{x}_{t+1}^v &= (I_{n_h-k_h} \otimes A_v)\mathfrak{x}_t^v + (I_{n_h-k_h} \otimes B_v)\boldsymbol{y}_t^h \\ \mathfrak{y}_t^v &= (I_{n_h-k_h} \otimes C_v)\mathfrak{x}_t^v + (I_{n_h-k_h} \otimes D_v)\boldsymbol{y}_t^h\end{array}\right\}, \quad \mathfrak{v}_t^v = \begin{pmatrix}\mathfrak{y}_t^v \\ \boldsymbol{y}_t^h\end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \mathfrak{x}_0^v = \mathbf{0}, \tag{3}$$

$$\left.\begin{array}{rl}\mathfrak{x}_{t+1}^h &= (A_h \otimes I_{n_v-k_v})\mathfrak{x}_t^h + (B_h \otimes I_{n_v-k_v})\boldsymbol{y}_t^v \\ \mathfrak{y}_t^h &= (C_h \otimes I_{n_v-k_v})\mathfrak{x}_t^h + (D_h \otimes I_{n_v-k_v})\boldsymbol{y}_t^v\end{array}\right\}, \quad \mathfrak{y}_t^h = \begin{pmatrix}\mathfrak{y}_t^h \\ \boldsymbol{y}_t^v\end{pmatrix}, \quad t = 0, 1, 2, \ldots, \quad \mathfrak{x}_0^h = \mathbf{0}, \tag{4}$$

Then, by using properties of the Kronecker product we obtain the following result.

Theorem 1 *For the vectors $\mathfrak{y}_t^v$ and $\mathfrak{y}_t^h$ defined by expressions (3) and (4) respectively, it follows that $\mathfrak{y}_t^v = \mathfrak{y}_t^h$, for $t = 0, 1, 2, \ldots$*

Proof By induction over t. □

Next result establishes that the ISO representations defined by matrices in expressions (1)–(4) are minimal ISO representations.

Theorem 2 *Let us assume that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal ISO representations of the (n_h, k_h, δ_h) horizontal and (n_v, k_v, δ_v) vertical codes $\mathscr{C}_h$ and $\mathscr{C}_v$, respectively. Then*

1. *The matrices $(A_h \otimes I_{k_v}, B_h \otimes I_{k_v}, C_h \otimes I_{k_v}, D_h \otimes I_{k_v})$ in expression (1) define a minimal ISO representation of an $(n_h k_v, k_h k_v, \delta_h k_v)$ convolutional code $\mathscr{C}_h(k_v)$.*
2. *The matrices $(I_{k_h} \otimes A_v, I_{k_h} \otimes B_v, I_{k_h} \otimes C_v, I_{k_h} \otimes D_v)$ in expression (2) define a minimal ISO representation of an $(k_h n_v, k_h k_v, k_h \delta_v)$ convolutional code $\mathscr{C}_v(k_h)$.*
3. *The matrices $(I_{n_h-k_h} \otimes A_v, I_{n_h-k_h} \otimes B_v, I_{n_h-k_h} \otimes C_v, I_{n_h-k_h} \otimes D_v)$ in expression (3) define a minimal ISO representation of an $((n_h-k_h)n_v, (n_h-k_h)k_v, (n_h-k_h)\delta_v)$ convolutional code $\mathscr{C}_v(n_h - k_h)$.*
4. *The matrices $(A_h \otimes I_{n_v-k_v}, B_h \otimes I_{n_v-k_v}, C_h \otimes I_{n_v-k_v}, D_h \otimes I_{n_v-k_v})$ in expression (4) define a minimal ISO representation of an $(n_h(n_v-k_v), k_h(n_v-k_v), \delta_h(n_v-k_v))$ convolutional code $\mathscr{C}_h(n_v - k_v)$.*

Proof The result follows from the fact that (A_h, B_h, C_h, D_h) and (A_v, B_v, C_v, D_v) are minimal ISO representations and the properties of the Kronecker product of matrices. □

It is not difficult to show that the codes $\mathscr{C}_h(k_v)$ and $\mathscr{C}_h(n_v - k_v)$ (respectively, $\mathscr{C}_v(k_h)$ and $\mathscr{C}_v(n_h - k_h)$) correspond to the block parallel concatenation of convolutional codes described in [4, Section 5.3], and therefore

$$\begin{aligned} d_{free}\left(\mathscr{C}_h(k_v)\right) &= d_{free}\left(\mathscr{C}_h(n_v - k_v)\right) = d_{free}\left(\mathscr{C}_h\right), \\ d_{free}\left(\mathscr{C}_v(k_h)\right) &= d_{free}\left(\mathscr{C}_v(n_h - k_h)\right) = d_{free}\left(\mathscr{C}_v\right). \end{aligned} \tag{5}$$

Now, by using the second model of serial concatenated convolutional codes introduced in [2, 4] we have the following result.

Theorem 3 *With the same notation as in Theorem 2.*

1. *If $\mathscr{S}_1$ is the rate $k_h k_v/((n_h - k_h)n_v + k_h k_v)$ convolutional code defined by the serial concatenation of $\mathscr{C}_h(k_v)$ and $\mathscr{C}_v(n_h - k_h)$, then* $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$*, with*

$$\mathbf{A}_1 = \begin{bmatrix} I_{n_h-k_h} \otimes A_v & C_h \otimes B_v \\ O & A_h \otimes I_{k_v} \end{bmatrix}, \qquad \mathbf{B}_1 = \begin{bmatrix} D_h \otimes B_v \\ B_h \otimes I_{k_v} \end{bmatrix},$$

$$\mathbf{C}_1 = \begin{bmatrix} I_{n_h-k_h} \otimes C_v & C_h \otimes D_v \\ O & C_h \otimes I_{k_v} \end{bmatrix}, \qquad \mathbf{D}_1 = \begin{bmatrix} D_h \otimes D_v \\ D_h \otimes I_{k_v} \end{bmatrix},$$

   is an ISO representation of $\mathscr{S}_1$.
2. *If $\mathscr{S}_2$ is the rate $k_h k_v/(n_h(n_v - k_v) + k_h k_v)$ convolutional code defined by the serial concatenation of $\mathscr{C}_v(k_h)$ and $\mathscr{C}_h(n_v - k_v)$, then* $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$*, with*

$$\mathbf{A}_2 = \begin{bmatrix} A_h \otimes I_{n_v-k_v} & B_h \otimes C_v \\ O & I_{k_h} \otimes A_v \end{bmatrix}, \qquad \mathbf{B}_2 = \begin{bmatrix} B_h \otimes D_v \\ I_{k_h} \otimes B_v \end{bmatrix},$$

$$\mathbf{C}_2 = \begin{bmatrix} C_h \otimes I_{n_v-k_v} & D_h \otimes C_v \\ O & I_{k_h} \otimes C_v \end{bmatrix}, \qquad \mathbf{D}_2 = \begin{bmatrix} D_h \otimes D_v \\ I_{k_h} \otimes D_v \end{bmatrix},$$

   is an ISO representation of $\mathscr{S}_2$.

Proof The result follows from Theorem 9 of [2]. □

In general the ISO representations $(\mathbf{A}_1, \mathbf{B}_1, \mathbf{C}_1, \mathbf{D}_1)$ and $(\mathbf{A}_2, \mathbf{B}_2, \mathbf{C}_2, \mathbf{D}_2)$ introduced in the above theorem are not minimal (see [2, 4]). In [2, 4] we can find some sufficient conditions to ensure the minimality of the above ISO representations.

Now, by Theorem 15 of [2] we have that

$$d_{free}(\mathscr{S}_1) \geq d_{free}(\mathscr{C}_h) \quad \text{and} \quad d_{free}(\mathscr{S}_2) \geq d_{free}(\mathscr{C}_v). \tag{6}$$

As a consequence of Theorem 1, by using the second model of parallel concatenation (see [4, Section 5.2]) we have the following result.

Theorem 4 *With the same notation as in Theorems 2 and 3.*

1. *If $\mathscr{P}_1$ is the rate $(k_h k_v/n_h n_v)$ convolutional code defined by the parallel concatenation of $\mathscr{S}_1$ and $\mathscr{C}_v(k_h)$, then* $(\mathfrak{A}_1, \mathfrak{B}_1, \mathfrak{C}_1, \mathfrak{D}_1)$ *with*

$$\mathfrak{A}_1 = \left[\begin{array}{cc|c} I_{n_h-k_h} \otimes A_v & C_h \otimes B_v & O \\ O & A_h \otimes I_{k_v} & O \\ \hline O & O & I_{k_h} \otimes A_v \end{array}\right], \quad \mathfrak{B}_1 = \left[\begin{array}{c} D_h \otimes B_v \\ B_h \otimes I_{k_v} \\ \hline I_{k_h} \otimes B_v \end{array}\right]$$

$$\mathfrak{C}_1 = \left[\begin{array}{cc|c} I_{n_h-k_h} \otimes C_v & C_h \otimes D_v & O \\ O & C_h \otimes I_{k_v} & O \\ \hline O & O & I_{k_h} \otimes C_v \end{array}\right], \quad \mathfrak{D}_1 = \left[\begin{array}{c} D_h \otimes D_v \\ D_h \otimes I_{k_v} \\ \hline I_{k_h} \otimes D_v \end{array}\right]$$

   is an ISO representation of $\mathscr{P}_1$.

2. *If $\mathscr{P}_2$ is the rate $(k_h k_v / n_h n_v)$ convolutional code defined by the parallel concatenation of $\mathscr{S}_2$ and $\mathscr{C}_h(k_v)$, then $(\mathfrak{A}_2, \mathfrak{B}_2, \mathfrak{C}_2, \mathfrak{D}_2)$ with*

$$\mathfrak{A}_2 = \left[\begin{array}{cc|c} A_h \otimes I_{n_v-k_v} & B_h \otimes C_v & O \\ O & I_{k_h} \otimes A_v & O \\ \hline O & O & A_h \otimes I_{k_v} \end{array}\right], \quad \mathfrak{B}_2 = \left[\begin{array}{c} B_h \otimes D_v \\ I_{k_h} \otimes B_v \\ \hline B_h \otimes I_{k_v} \end{array}\right]$$

$$\mathfrak{C}_2 = \left[\begin{array}{cc|c} C_h \otimes I_{n_v-k_v} & D_h \otimes C_v & O \\ O & I_{k_h} \otimes C_v & O \\ \hline O & O & C_h \otimes I_{k_v} \end{array}\right], \quad \mathfrak{D}_2 = \left[\begin{array}{c} D_h \otimes D_v \\ I_{k_h} \otimes D_v \\ \hline D_h \otimes I_{k_v} \end{array}\right]$$

is an ISO representation of $\mathscr{P}_2$.

Note that, according to expressions (1)–(4) and Theorem 1, $\mathscr{P}_1$ is the product convolutional code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$. Moreover, since $\mathfrak{A}_1$ is a matrix of size $(n_h\delta_v + \delta_h k_v) \times (n_h\delta_v + \delta_h k_v)$ and the complexity of $\mathscr{C}$ is $k_h\delta_v + \delta_h k_v$, we can ensure that the ISO representation $(\mathfrak{A}_1, \mathfrak{B}_1, \mathfrak{C}_1, \mathfrak{D}_1)$ provided by part 1 of Theorem 4 is nonminimal. By an analogous argument $\mathscr{P}_2$ is the product convolutional code $\mathscr{C} = \mathscr{C}_h \otimes \mathscr{C}_v$ and the ISO representation $(\mathfrak{A}_2, \mathfrak{B}_2, \mathfrak{C}_2, \mathfrak{D}_2)$ provided by part 2 of Theorem 4 is nonminimal.

Next result introduces a lower bound on the free distance d_{free} of the convolutional product code in terms of the constituent convolutional codes.

Theorem 5 *If $\mathscr{C}_h$ and $\mathscr{C}_v$ are (n_h, k_h, δ_h) and (n_v, k_v, δ_v) codes, respectively, then,*

$$d_{free}(\mathscr{C}_h \otimes \mathscr{C}_v) \geq \max\left\{d_{free}(\mathscr{C}_v), d_{free}(\mathscr{C}_h)\right\}.$$

Proof With the same notation as in Theorem 4, as a consequence of Theorem 5.8 of [4] we have that

$$d_{free}(\mathscr{P}_1) \geq \max\left\{d_{free}(\mathscr{S}_1), d_{free}(\mathscr{C}_v)\right\},$$

$$d_{free}(\mathscr{P}_2) \geq \max\left\{d_{free}(\mathscr{S}_2), d_{free}(\mathscr{C}_h)\right\}.$$

The result follows now by expressions (5) and (6) an the fact that $\mathscr{P}_1 = \mathscr{P}_2 = \mathscr{C}_h \otimes \mathscr{C}_v$. □

Acknowledgements This work was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Ciencia e Innovación of the Gobierno de España.

References

1. 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, pp. 1063–1066. IEEE (2005). doi:10.1109/ISIT.2005.1523502

2. Climent, J.J., Herranz, V., Perea, C.: A first approximation of concatenated convolutional codes from linear systems theory viewpoint. Linear Algebr. Appl. **425**, 673–699 (2007). doi:10.1016/j.laa.2007.03.017
3. Elias, P.: Error free coding. Trans. IRE Prof. Group Inf. Theory **4**(4), 29–37 (1954)
4. Herranz, V.: Estudio y construcción de códigos convolucionales: Códigos perforados, códigos concatenados desde el punto de vista de sistemas. Ph.D. thesis, Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández, Elche (2007)
5. Hutchinson, R., Rosenthal, J., Smarandache, R.: Convolutional codes with maximum distance profile. Syst. Control Lett. **54**(1), 53–63 (2005). doi:10.1016/j.sysconle.2004.06.005
6. Johannesson, R., Wan, Z.X.: A linear algebra approach to minimal convolutional encoders. IEEE Trans. Inf. Theory **39**(4), 1219–1233 (1993). doi:10.1109/18.243440
7. Justesen, J.: New convolutional code constructions and a class of asymptotically good time-varying codes. IEEE Trans. Inf. Theory **19**(2), 220–225 (1973). doi:10.1109/TIT.1973.1054983
8. Justesen, J.: An algebraic construction of rate $1/\nu$ convolutional codes. IEEE Trans. Inf. Theory **21**(5), 577–580 (1975). doi:10.1109/TIT.1975.1055436
9. Kalman, R.E., Falb, P.L., Arbib, M.A.: Topics in Mathematical System Theory. McGraw-Hill, New York (1969)
10. Massey, J.L., Costello, D.J., Justesen, J.: Polynomial weights and code constructions. IEEE Trans. Inf. Theory **19**(1), 101–110 (1973). doi:10.1109/TIT.1973.1054936
11. Medina, C., Sidorenko, V.R., Zyablov, V.V.: Error exponents for product convolutional codes. Probl. Inf. Transm. **42**(3), 167–182 (2006). doi:10.1134/S003294600603001X
12. Rosenthal, J.: An algebraic decoding algorithm for convolutional codes. Prog. Syst. Control Theory **25**, 343–360 (1999). doi:10.1007/978-3-0348-8970-4_16
13. Rosenthal, J.: Connections between linear systems and convolutional codes. In: Marcus, B., Rosenthal, J. (eds.) Codes, Systems and Graphical Models. The IMA Volumes in Mathematics and its Applications, vol. 123, pp. 39–66. Springer, New York (2001). doi:10.1007/978-1-4613-0165-3_2
14. 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, Urbana, pp. 953–960. Allerton House, Monticello (1997)
15. Rosenthal, J., Smarandache, R.: Maximum distance separable convolutional codes. Appl. Algebra Eng. Commun. Comput. **10**(1), 15–32 (1999). doi:10.1007/s002000050120
16. Rosenthal, J., York, E.V.: BCH convolutional codes. IEEE Trans. Inf. Theory **45**(6), 1833–1844 (1999). doi:10.1109/18.782104
17. Rosenthal, J., Schumacher, J.M., York, E.V.: On behaviors and convolutional codes. IEEE Trans. Inf. Theory **42**(6), 1881–1891 (1996). doi:10.1109/18.556682
18. Sripimanwat, K. (ed.): Turbo Code Applications. A Journey from a Paper to Realization. Springer, Dordrecht (2005). doi:10.1007/1-4020-3685-X
19. Tanner, R.M.: Convolutional codes from quasicyclic codes: a link between the theories of block and convolutional codes. Technical Report USC-CRL-87-21, University of California, Santa Cruz (1987)
20. 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 (1997)
21. Zyablov, V., Shavgulidze, S., Skopintsev, O., Höst, S., Johannesson, R.: On the error exponent for woven convolutional codes with outer warp. IEEE Trans. Inf. Theory **45**(5), 1649–1653 (1999). doi:10.1109/18.771237
22. Zyablov, V.V., Shavgulidze, S., Johannesson, R.: On the error exponent for woven convolutional codes with inner warp. IEEE Trans. Inf. Theory **47**(3), 1195–1199 (2001). doi:10.1109/18.915681