21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Increasing Forward Error Correcting Codes Capabilities with Convolutional Codes over any finite field

Vicente Galiano[1] and Roberto Gandía[2] and Victoria Herranz[2]

***Abstract*— In this paper the authors introduce a model for evaluating the error correcting capabilities of convolutional codes. Until now, the developed models were limited to a binary field with only two elements. However, in this work we focus on the analysis of MDS, sMDS and not MDS codes defined over any finite field. We validate theoretical results of convolutional codes in a more realistic model to obtain the recovering error capability and we compare it with the results obtained simulating binary convolutional codes used currently in common communications systems as GSM or 802.11a.**

I. Introduction

Information is one of the most valuable goods of our time. However, the physical means used to transmit and store information are never perfect and they are subject to errors that might result in loss of important data. To avoid it, forward error correcting codes, such as convolutional codes, are used to correct them and guarantee an error-free communication.

Convolutional codes were considered the first time by Elias in [3] and the theory in context of finite fields (not necessarily binary fields) was formalized first by Forney [4]. These fields are frequently used to correct errors in noisy channels. Although convolutional encoding is a simple procedure, decoding a convolutional code is a much more complex task. There exist several classes of algorithms for this purpose, but Viterbi decoding [12] has the advantage that soft information can be processed. However, its main drawback is that it is too complex for codes with high degree, high memory or when the block length is large.

In recent literature on convolutional codes (see [6], [7], [9], [11]) several new classes of codes with optimal distance properties have been introduced and studied. These classes of codes are known as maximum distance separable (MDS) codes and strongly MDS (sMDS) codes. Such codes are particularly suited for applications where large alphabets are involved. However, they have not been tested in communications system and only theoretical results are obtained.

In this paper, authors complete the gap between the proposal of convolutional codes defined over any finite field and the current engineering models which are limited to the binary field. In section II, we introduce the fundamentals of convolutional coding theory from linear systems point of view. We describe the model we have developed to evaluate the efficiency of convolutional codes over any finite field in Section III. Once defined it, we propose some examples of convolutional codes with different complexities in Sections IV and V to compare its recovering error capability with the binary ones currently used in many communication systems . Finally, we take some conclusions in Section VI.

*The work of these authors was partially supported by Spanish grant MTM2011-24858 of the Ministerio de Economia y Competitividad of the Spanish Government and supported by IPZS01 Proyectos de Investigacion UMH-Bancaja.

[1]Vicente Galiano is Physics and Computer Architecture Department, Miguel Hernández University. Elche, Spain 03202 vgaliano@umh.es

[2]Roberto Gandía and Victoria Herranz are with Center of Operations Research, Departamento de Estadística, Matemáticas e Informática, Miguel Hernández University. Elche, Spain 03202 mavi.herranz@umh.es

II. Preliminary results

In this paper, we denote by $\mathbb{F} = GF(q)$ the Galois field of q elements and $\mathbb{F}[z]$ the polynomial ring on the variable z with coefficients in $\mathbb{F}$. A *convolutional code* of rate k/n is a submodule $\mathcal{C}$ of $\mathbb{F}^n[z]$ with rank k. So, there exists an $n \times k$ polynomial generator matrix $G(z) \in \mathbb{F}[z]^{n\times k}$, which is basic and minimal, such that

$$\mathcal{C} = \{\vec{v}(z) \in \mathbb{F}[z]^n : \vec{v}(z) = G(z)\vec{u}(z) \text{ with } \vec{u}(z) \in \mathbb{F}[z]^k\}.$$

The *degree* or *complexity* of $\mathcal{C}$ is the sum of the column degrees of one, and hence any, minimal basic generator matrix. Consider the matrices $A \in \mathbb{F}^{\delta\times\delta}$, $B \in \mathbb{F}^{\delta\times k}$, $C \in \mathbb{F}^{(n-k)\times\delta}$ and $D \in \mathbb{F}^{(n-k)\times k}$. Following [8], a rate k/n convolutional code $\mathcal{C}$ of complexity δ, called an (n, k, δ)-code, can be described also by the linear system governed by the equations

$$\left.\begin{array}{rcccl} x_{t+1} & = & Ax_t & + & Bu_t \\ y_t & = & Cx_t & + & Du_t \end{array}\right\}, \quad t = 0, 1, 2, \ldots \qquad (1)$$

$$v_t = \begin{pmatrix} y_t \\ u_t \end{pmatrix}, \quad x_0 = 0,$$

where for each time instant t, $x_t \in \mathbb{F}^\delta$ is the *state vector*, $u_t \in \mathbb{F}^k$ is the *information vector*, $y_t \in \mathbb{F}^{n-k}$ is the *parity vector* and $v_t \in \mathbb{F}^n$ is the *code vector*. In linear systems theory, this representation is known as the *input-state-output representation*. It was introduced by Rosenthal, Schumacher and York (see [8]) and it has been widely used in the last years to analyze and construct convolutional codes [1], [2], [10], [11].

In terms of an input-state-output representation, the free distance of a convolutional code $\mathcal{C}$ can be defined (see [10]) as

$$\mathrm{d}_{\mathrm{free}}(\mathcal{C}) = \min\left(\sum_{t=0}^{\infty} \mathrm{wt}(u_t) + \sum_{t=0}^{\infty} \mathrm{wt}(y_t)\right) \qquad (2)$$

where the minimum has to be taken over all possible nonzero codewords and where $\mathrm{wt}(v)$ denotes the Hamming weight of a vector v. As in the case of block codes, there is an upper

ISBN: 978-90-367-6321-9

bound, known as the *generalized Singleton bound*, for how large the free distance of an (n, k, δ)-code can be:

$$\mathrm{d_{free}}(\mathcal{C}) \leq (n-k)\left(\left\lfloor \frac{\delta}{k} \right\rfloor + 1\right) + \delta + 1. \qquad (3)$$

This bound was proved by Rosenthal and Smarandache in [9]. $\mathcal{C}$ is called an *MDS code* if $d_{free}(\mathcal{C})$ attains the generalized Singleton bound.

Another distance measure is the jth column distance of a convolutional code. It takes into account only truncations of codewords and is given by (see [7])

$$d_j^c(\mathcal{C}) = \min_{u_0 \neq 0} \left\{ \sum_{t=0}^{j} \mathrm{wt}(u_t) + \sum_{t=0}^{j} \mathrm{wt}(y_t) \right\}$$

for $j = 0, 1, 2, \ldots$. $\mathcal{C}$ is called a *strongly MDS code* if $d_M^c(\mathcal{C}) = (n-k)\left(\lfloor \frac{\delta}{k} \rfloor + 1\right) + \delta$ for $M = \lfloor \frac{\delta}{k} \rfloor + \lceil \frac{\delta}{n-k} \rceil$, that is, the sequence $\{d_j^c(\mathcal{C})\}_{j \geq 0}$ attains the generalized Singleton bound at the earliest possible step. So, in particular, a strongly MDS code is an MDS code.

III. Designing a Simulation Tool

In real communication systems, the data source is represented and transmitted in binary format composed by only two elements: $\{0, 1\}$. This is the main reason because communication systems use convolutional coding over a finite field with two elements, $GF(2) = \{0, 1\}$. However, as we have seen previously, finite fields with a high number of elements theoretically demonstrate a high corrective capability. In order to verify if this theoretical issue agrees with practical results, we define in Sections IV and V a set of examples of MDS convolutional codes with specific characteristics together WITH many examples where correcting capabilities are not so good.

In order to evaluate these codes in a more realistic environment, we define a simulation process represented in Figure 1 and detailed by steps (A), (B), etc. In the first step, step (A), we transform the bit sequence into a sequence of elements belonging to the finite field where is defined the convolutional code. Specifically, in our tests we use a Galois field with 8 elements, $GF(8)$. To perform this conversion, each sequence of three bits is collected and assigned to a field element u_t. The second step, (B), implements the convolutional coding generating the output element y_t (step (B1)), where $y_t = Cx_t + Du_t$, (see relation (1)) and getting the next state x_{t+1} (step (B2)) of the convolutional code. Observe that this step depends on the input u_t and the current state x_t, since $x_{t+1} = Ax_t + Bu_t$, (see relation (1)). As next state depends from the previous one, we must define an initial state $x_0 = 0$ for t_0. In the step (C), we concatenate and obtain $v_t = \begin{pmatrix} y_t & u_t \end{pmatrix}^T$. Elements of finite field $GF(8)$ are turned into a binary sequence in step (D) and transmitted in step (E). In this step, different modulations (FSK, QAM, etc.) can be performed to recover binary stream in the receiver. We assume that an Additive White Gaussian Noise (AWGN) affects to our communication channel. As we describe in in section IV, the signal quality in the receiver depends on the bit energy-Noise relation (E_b/N_0). In the receiver, after the demodulation, bits are grouped again and converted to an element v_t' (step (F)). Observe that v_t' is not necessarily equal to v_t because noise introduces an error in the bit stream received. Step (G) implements the Viterbi algorithm (see [12]) in order to obtain u_t'. This algorithm is widely used in digital communications systems and it will be implemented in our model. However, when the number of elements of the finite field is greater than 2, some authors [6] recommend and propose other algorithms more efficient than Viterbi decoding. Finally, in step (H), elements u_t' are transformed to a binary sequence and added to the output binary stream.

The output binary stream is compared with the input binary stream, and we obtain the bit error rate as $BER = \frac{\text{Errors Received}}{\text{Total Transmitted}}$. This software tool has been implemented using CML distribution according to the software review explained in [5].

IV. Simulating MDS codes with complexity $\delta = 1$

In the rest of the paper, we consider α as a primitive element of the field $\mathbb{F} = GF(8)$, with $\alpha^3 + \alpha + 1 = 0$. In the first example, we use the construction proposed by Smarandache and Rosenthal (see [11]) in order to obtain an MDS convolutional code.

Example 1: Let $\mathcal{C}_1(A_1, B_1, C_1, D_1)$ be the $(2, 1, 1)$-code with

$$A_1 = (\alpha), \qquad B_1 = (1), \qquad C_1 = (\alpha^4), \qquad D_1 = (1)$$

Then, $\mathcal{C}_1$ is an strongly MDS convolutional code. In particular, we get that $d_{free}(\mathcal{C}_1) = 4$.

Next example also shows two strongly MDS convolutional codes of rate $1/2$ and complexity $\delta = 1$.

Example 2: Let $\mathcal{C}_2(A_2, B_2, C_2, D_2)$ and $\mathcal{C}_3(A_3, B_3, C_3, D_3)$ be the $(2, 1, 1)$-codes described by matrices

$$A_2 = (\alpha), \qquad B_2 = (1), \qquad C_2 = (1), \qquad D_2 = (1)$$

$$A_3 = (\alpha), \qquad B_3 = (1), \qquad C_3 = (\alpha^2), \qquad D_3 = (\alpha^3)$$

respectively. Through computation, we get $d_2^c(\mathcal{C}_2) = \mathrm{d_{free}}(\mathcal{C}_2) = 4$ and $d_2^c(\mathcal{C}_3) = \mathrm{d_{free}}(C_3) = 4$, so $\mathcal{C}_2$ and $\mathcal{C}_3$ are strongly MDS convolutional codes.

Finally, we propose two convolutional codes that are not MDS codes.

Example 3: Let $\mathcal{C}_4(A_4, B_4, C_4, D_4)$ and $\mathcal{C}_5(A_5, B_5, C_5, D_5)$ be the $(2, 1, 1)$-codes described by matrices

$$A_4 = (1), \qquad B_4 = (\alpha^5), \qquad C_4 = (1), \qquad D_4 = (0)$$

$$A_5 = (\alpha), \qquad B_5 = (1), \qquad C_5 = (1), \qquad D_5 = (\alpha^6)$$

respectively. Through computation, we get the free distances of these codes, $\mathrm{d_{free}}(\mathcal{C}_4) = 3$ and $\mathrm{d_{free}}(C_5) = 3$. But in this case, the Singleton bound is equal to 4 so $\mathcal{C}_4$ and $\mathcal{C}_5$ are not MDS convolutional codes.

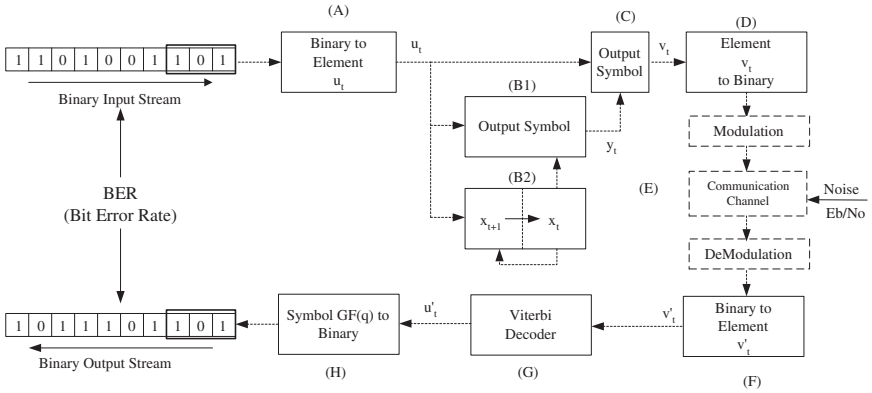

Fig. 1. Convolutional coding/decoding over $GF(8)$.

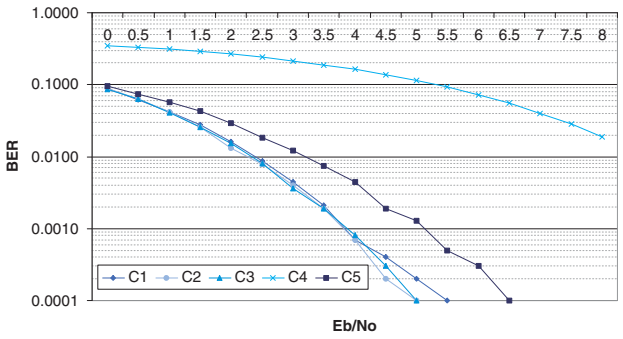

Fig. 2. Performance in BER with Complexity $\delta = 1$

The Figure 2 represents the BER obtained for each convolutional code transmitting 150kbits for different E_b/N_0 relations. As we can see, the code $\mathcal{C}_1$ proposed by Rosenthal and Smarandache (see [11]) has the best performance. Observe that $\mathcal{C}_2$ and $\mathcal{C}_3$ have good recovering factors too. Note that the E_b/N_0 values used in these tests represent a very poor signal levels. For example, a zero value represents that E_b and N_o has the same power level. With these poor signal levels, $\mathcal{C}_1$ is able to recover and obtain a BER minor than 0.089. On the other hand, $\mathcal{C}_4$ and $\mathcal{C}_5$ are not theoretically good error correcting codes, so the BER is bigger for these two codes for all the values of E_b/N_0. For example, we obtain a BER upper than 0.359 for a $E_b/N_0 = 0$ value.

As we see in Example 1, the convolutional code $\mathcal{C}_1$ is an MDS code, so it is one of the most efficient error correcting code among all convolutional codes with the same parameters. At this point, we believe interesting to compare this code $\mathcal{C}_1$ with other convolutional codes with the same rate used in digital systems. In this sense, we make simulations for the same number of bits, range of E_b/N_0 values and modulation QPSK. Table I shows the generator matrices and the free distances of the codes used to compare with the code $\mathcal{C}_1$. Observe that all these codes have rate 1/2 and complexity $\delta > 1$, so the generator matrix is $G(z) = \begin{pmatrix} g_{11}(z) & g_{12}(z) \end{pmatrix}$, where $g_{11}(z)$ and $g_{12}(z)$ are represented in octal. They are not MDS convolutional codes.

	$g_{11}(z)$	$g_{12}(z)$	d_{free}
NASA-SVP	171	133	10
GSM	31	33	7
802.11a	155	117	10
IS-95	657	435	12

TABLE I
RATE 1/2 CONVOLUTIONAL CODES IN FIELD $\mathbb{F} = GF(2)$

In Figure 3 we show the BER obtained for different noise levels and for each convolutional code. Observe that, for a very noisy channel, $\mathcal{C}_1$ obtains bigger error correcting capability. On the other hand, when E_b/N_0 relation increases, the BER obtained in binary systems are better than the BER obtained with $\mathcal{C}_1$.

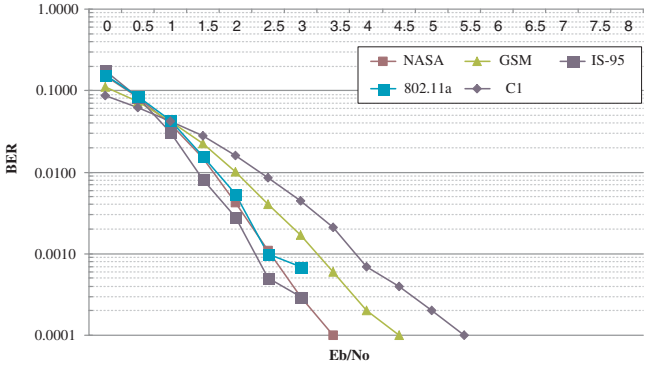

Fig. 3. Performance comparison of a 1-complexity code and binary convolutional codes

V. SIMULATING MDS CODES WITH COMPLEXITY $\delta > 1$

As we show in the Figure 3 of Section IV, the Bit Error Rate for codes with complexity δ=1 over the field $\mathbb{F} = GF(8)$ is not better than the one obtained with binary codes. Analyzing the results and comparing them with the results obtained for binary codes like IS-95, we reached the

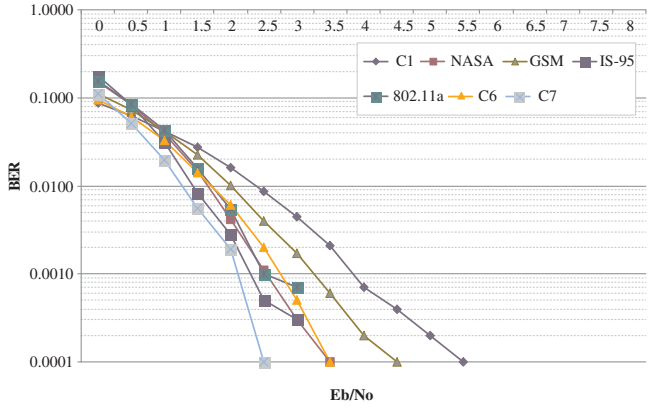

Fig. 4. Performance comparison of 1, 2, 3-complexity codes and binary convolutional codes

conclusion that it could be possible to improve them using codes over $GF(8)$ with greater complexity to obtain better results, because the bigger the complexity is then the stronger our code will be.

So that, with the aim of comparing codes of different fields, we consider α as a primitive element of the field $\mathbb{F} = GF(8)$, with $\alpha^3 + \alpha + 1 = 0$, as we already did in the section IV and use an MDS convolutional code with complexity δ=2 and then an MDS convolutional code with δ=3

Example 4: Let $\mathcal{C}_6(A_6, B_6, C_6, D_6)$ be the $(2, 1, 2)$-code with

$$A_6 = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^2 \end{pmatrix}, \qquad B_6 = \begin{pmatrix} 1 \\ \alpha \end{pmatrix},$$

$$C_6 = \begin{pmatrix} \alpha^3 & \alpha^4 \end{pmatrix}, \qquad D_6 = (1)$$

Let $\mathcal{C}_7(A_7, B_7, C_7, D_7)$ be the $(2, 1, 3)$-code with

$$A_7 = \begin{pmatrix} \alpha & \alpha^3 & \alpha^4 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha^2 \end{pmatrix}, \qquad B_7 = \begin{pmatrix} 1 \\ 1 \\ \alpha \end{pmatrix},$$

$$C_7 = \begin{pmatrix} \alpha^2 & \alpha^6 & 1 \end{pmatrix} \qquad D_7 = (\alpha^3)$$

Then, $\mathcal{C}_6$ and $\mathcal{C}_7$ are MDS convolutional code. In particular, we get that $d_{free}(\mathcal{C}_6) = 4$ and $d_{free}(\mathcal{C}_7) = 8$.

The Figure 4 represents the BER obtained for these convolutional code transmitting 150kbits for different E_b/N_0 relations. As we can see, the MDS code $\mathcal{C}_7$ is the most efficient error correcting code among all convolutional codes with the same parameters. That is, the codes defined in a large field and with a large complexity have the best performance. / this code $\mathcal{C}_1$ with other convolutional codes used in digital systems. For that, it is required a transformation to compare properly the codes with $\mathbb{F} = GF(2)$ and $\mathbb{F} = GF(8)$. As the number of bits to represent a symbol in $\mathbb{F} = GF(2)$ is one and the number of bits in $\mathbb{F} = GF(8)$ is three, when we say that a code in $\mathbb{F} = GF(8)$ has complexity $\delta = 1$ means that if we are comparing this code with a binary one, we will have to take a binary code with complexity three. Equally, if we take a $\mathbb{F} = GF(8)$ code with $\delta = 2$ it will be able to be compared with a binary code of complexity six. Under this requirement it will be possible to compare two codes with the same features but in different fields. Knowing that and observing the figure 4, it is possible to see that $\mathcal{C}_3$ is better than the binary code IS-95, which has exactly the same features as $\mathcal{C}_3$, but in another field.

VI. Conclusions

In this paper we present a first implementation of a software model to evaluate the error correcting capability of a convolutional code defined over any finite field. Experimental results show that strongly MDS codes with complexity $\delta > 1$ have better error correcting capability than the codes currently used in real systems with the same complexity. A future research line is to evaluate the memory requirements and the computational time needed to do the coding and the decoding process of convolutional codes defined over any finite field. Moreover, we want to apply different decoding algorithms for convolutional codes over any finite field instead of the Viterbi algorithm.

References

[1] J. -J. Climent, V. Herranz, C. Perea. "A first approximation of concatenated convolutional codes from linear systems theory viewpoint", *Linear Algebra and its Applications*, vol. 425, pp. 673–699, 2007.

[2] J. -J. Climent, V. Herranz, C. Perea. "Linear system modelization of concatenated block and convolutional codes", *Linear Algebra and its Applications*, vol. 429, pp. 1191–1212, 2008.

[3] P. Elias. "Coding for noisy channels". *IRE Conv. Rec.*, pp. 37–47, 1955.

[4] G. D. Forney. "Convoltuional codes I: Algebraic structure" . *IEEE Trans. Information Theory* (1970).

[5] V. Galiano, R. Gandia and V. Herranz. "Review of simulations tools for channel coding". *WIT Transactions on Information and Communication Technologies, WIT Press*, vol 45, pp. 3–12, 2013.

[6] H. Gluesing-Luerssen, J. Ronsethal, S. Smarandache, " Strongly-MDS convolutional codes", *IEEE Transactions on Information Theory*, vol 52, no. 2, pp. 584–598, 2006.

[7] R. Hutchinson, J. Rosenthal, and R. Smarandache. "Convolutional codes with maximum distance profile". *Systems and Control Letters*, vol 54 no. 1, pp. 53–63, 2005.

[8] J. Rosenthal, J.M. Schumacher and E.V. York. " On behaviors and convolutional codes". *IEEE Transactions on Information Theory,* vol 42, no. 6, pp. 1881-1891, 1996.

[9] J. Rosenthal and R. Smarandache. " Maximum distance separable convolutional codes". *Applicable Algebra in Engineering, Communication and Computing,* vol 10, pp. 15–32, 1999.

[10] J. Rosenthal and E.V. York. "BCH Convolutional Codes" *IEEE Transactions on Information Theory*, vol. 45, no. 6, 1999.

[11] R. Smarandache and J. Rosenthal. " A state space approach for constructing MDS rate $1/n$ convolutional codes". *Proceedings of the* 1998 *IEEE Workshop on Information Theory. Killarney, Kerry, Ireland.*, pp. 116–117, 1998.

[12] Viterbi AJ. " Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". *IEEE Transactions on Information Theory*, vol 13, no. 2, pp. 260-269 doi:10.1109/TIT.1967.1054010, 1967.