Decoding of a class of convolutional codes

Heide Gluesing-Luerssen, Uwe Helmke and José Ignacio Iglesias Curto

***Abstract*— A general decoding algorithm for convolutional codes will be exposed. Under certain conditions this algorithm will allow to correct a high number of errors in an interval of fixed length. We will show that a class of Cyclic Convolutional Codes is particularly well suited for this iterative algorithm in two aspects: first, it satisfies the conditions so that the error correcting capacities of the algorithm are optimized; secondly, the computations needed at each iteration are feasible. Consequently, the application of the algorithm to this class of codes results in an efficient decoding method.**

I. INTRODUCTION

As well as for block codes, the main objective in convolutional coding is to construct codes with good properties, i.e. that allows to carry the biggest amount of information and to correct the maximum number of errors, and that can be decoded in practice. It is well known that no decoding algorithm exists that can be used in practice for any code.

Similarly to the case of syndrome decoding for block codes, the Viterbi decoding algorithm for convolutional codes [16] can be theoretically applied to any code. However for high values on the degree of the code its use in practice becomes unfeasible. As a result, particular algorithms are devised for each family of convolutional codes, as it is also the case for block codes. Such algorithms make use of the particular characteristics of the corresponding family. Hence, they way in which a class of codes is constructed has an effect in both the main aspects commented before: the performance characteristics of the code and the possibility of efficient decoding.

Early algebraic constructions of convolutional codes date back to the 70's, as for example [8], [10]. In particular, an analogy to the notion of a cyclic code for the convolutional context has been investigated in [10], [1], [2]. From another perspective, the relationship of convolutional codes and linear systems has been exploited [11], [13], [14], [15], and even algebraic geometric constructions have been developed [4], [5], [9]. However, the number of decoding algorithms is still quite reduced. Examples can be found e.g. in [7], [12].

It is the aim of this paper to propose a general decoding algorithm that works for a wide class of convolutional codes, but is particularly well adapted to the class of doubly-cyclic convolutional codes. It will turn out that such adaptation is feasible in practice.

The structure of the paper is as follows. In the rest of Section I we will introduce the basic notions on convolutional codes that will be needed for the understanding of the paper. In Section II a general decoding algorithm for convolutional codes will be exposed. In Section III we will show how doubly cyclic codes are constructed, as presented in [2], since these are the codes that our algorithm will decode. Section IV is devoted to the adaptation of the general algorithm for the particular case of doubly cyclic codes. Finally Section V contains the conclusions of this work and points to some directions for future research in this topic.

Throughout our paper $\mathbb{F}$ will denote a finite field of q elements. Recall that a *convolutional code $\mathcal{C}$ of length n and dimension k* is a k-rank $\mathbb{F}[z]$-submodule of $\mathbb{F}[z]^n$, that can be obtained as

$$\mathcal{C} = \{uG | u \in \mathbb{F}[z]^k\} \tag{1}$$

where G is a polynomial matrix known as the *generator matrix* of $\mathcal{C}$. We will require that G is *basic*, i.e. $rank(G(\alpha)) = k$ for all α in an algebraic closure of $\mathbb{F}$, and *reduced*, i.e. the sum of its row degrees equals the maximal degree of any k-minor. Since G is polynomial we can write $G = \sum_{i=0}^{m} G_i z^i$, $G_i \in \mathbb{F}^{k\times n}$, with m the memory of the code (which is introduced later).

A convolutional codeword is a polynomial vector $c(z) = \sum_{i=0} c_i z^i$ with $c_i \in \mathbb{F}^n$, that can be also thought of as the sequence $(c_0, c_1, c_2, \ldots)$. Further, if $u(z) = \sum_{i=0} u_i z^i$ with $u_i \in \mathbb{F}^n$ is the information vector such that $c(z) = u(z)G$, that can be thought of as the sequence $(u_0, u_1, u_2, \ldots)$, then

$$(c_0, c_1, c_2, \ldots) = (u_0, u_1, u_2, \ldots)\mathfrak{G} \tag{2}$$

where $\mathfrak{G}$ is the constant sliding matrix

$$\mathfrak{G} = \begin{pmatrix} G_0 & G_1 & \ldots & G_m & 0 & \ldots & \ldots \\ & G_0 & \ldots & G_{m-1} & G_m & 0 & \ldots \\ & & \ddots & \vdots & & & \\ & & & G_0 & G_1 & & \ldots \\ & & & & \ddots & \ddots & \end{pmatrix} \tag{3}$$

Convolutional codes are characterized by their length, dimension and free distance, together with two more parameters that measure the interdependency of the sequence components: the *memory*, i.e. the maximal degree of any entry in a reduced generator matrix, and the *degree*, i.e. the sum of the row degrees in a reduced generator matrix. The row degrees of a reduced matrix are invariants of the code

Heide Gluesing-Luerssen is with the Department of Mathematics, University of Kentucky, 715 Patterson Office Tower, Lexington, KY 40506-0027, USA heidegl@ms.uky.edu

Uwe Helmke is with the Institut für Mathematik, Universität Würzburg, Am Hubland, 97074 Würzburg, Germany helmke@mathematik.uni-wuerzburg.de

José Ignacio Iglesias Curto is with the Departamento de Matemáticas, Universidad de Salamanca, Plaza de la Merced 1–4, 37008 Salamanca, Spain joseig@usal.es

ISBN 978-963-311-370-7

and they are commonly known as the *Forney indices* of the code.

The *free distance* of a convolutional code is the analogy to the minimum distance of a block code. Let us define the overall Hamming weight of a polynomial vector $v(z) = \sum_{i=0} v_i z^i$ as $\mathrm{w}(v(z)) = \sum_{i=0} \mathrm{w}(v_i)$. Then the free distance of the code $\mathcal{C}$ is defined as

$$d_{free}(\mathcal{C}) = \min_{c \in \mathcal{C}} \mathrm{w}(c)\,. \tag{4}$$

II. A DECODING ALGORITHM FOR CONVOLUTIONAL CODES

In this Section we expose a decoding algorithm for convolutional codes, presented in a different way in [6], [3]. This algorithm would be theoretically of application to any convolutional code. It works sequentially on the received codeword by decoding each "piece" with the information from the previous ones, which are assumed to be correctly decoded. The idea for the recursion is rather simple: if a codeword is correctly decoded up to a certain instant, one can recover the corresponding information subsequence and use it to remove its influence on the later time instants. Hence, decoding the resulting sequence is equivalent to decoding a received sequence from the initial time instant.

For that we will need to fix the length of the subsequence that we will decode at each step, L, and the length of the decoding window that will be used for it, $N > L$. Both parameters will be fixed for each particular case in order to get the best performance. (Note that higher values for L will result in faster decoding while lower values for L will allow, as we will see later, to decode more errors per step.)

Let us consider a convolutional code $\mathcal{C}$ with memory m and generator matrix $G = \sum_{i=0}^{m} G_i z^i$, which we wish to decode with our algorithm. The performance of this algorithm will be based on the decoding of a constant vector with respect to the block code $\mathcal{C}_N := \operatorname{im} \mathcal{G}_N \subseteq \mathbb{F}^{Nn}$, being

$$\mathcal{G}_N := \begin{pmatrix} G_0 & G_1 & \dots & G_{N-1} \\ & G_0 & \dots & G_{N-2} \\ & & \ddots & \vdots \\ & & & G_0 \end{pmatrix} \in \mathbb{F}^{Nk \times Nn}, \tag{5}$$

where G_i is assumed to be 0 whenever $i > m$. Since G_0 is a full row rank matrix this holds also for $\mathcal{G}_N$.

Let $\tilde{d} \in \mathbb{N}$ be such that for all $c = (c_0, \dots, c_{N-1}) \in \mathcal{C}_N$, if $\mathrm{w}(v) \leq \tilde{d}$ then $(v_0, \dots, v_{L-1}) = 0$. The following is a direct consequence of this definition of $\tilde{d}$.

Lemma 1. *For any vector $v = (v_0, \dots, v_{N-1})$, if there are two codewords $c = (c_0, \dots, c_{N-1}), c' = (c'_0, \dots, c'_{N-1}) \in \mathcal{C}_N$ such that $d(v, c) \leq \lfloor \frac{\tilde{d}}{2} \rfloor$, $d(v, c') \leq \lfloor \frac{\tilde{d}}{2} \rfloor$, then $c_i = c'_i$ for all $0 \leq i \leq L-1$.*

This means that we can correct $\lfloor \frac{\tilde{d}}{2} \rfloor$ errors with respect to $\mathcal{C}_N$ if we need correct decoding only in the L first components. In the context of our algorithm this will mean that we will decode correctly as long as at most $\lfloor \frac{\tilde{d}}{2} \rfloor$ errors occur on each window of size N. In the rest of the paper we will be referring to this condition as the *error constraint*. Note that for lower values of L we can have higher values for $\tilde{d}$ and hence, as mentioned before, we will be able to correct more errors.

To carry out our algorithm we will also make use of the matrix $\widetilde{\mathcal{G}_N} \in \mathbb{F}^{mk \times Nn}$, where

$$\widetilde{\mathcal{G}_N} = \begin{pmatrix} G_m & & & \\ G_{m-1} & G_m & & \\ G_{m-2} & G_{m-1} & \ddots & \\ \vdots & \vdots & & G_m \\ G_2 & G_3 & & \vdots \\ G_1 & G_2 & \dots & G_N \end{pmatrix} \text{ if } N \leq m$$

$$\widetilde{\mathcal{G}_N} = \begin{pmatrix} G_m & & & & 0 \\ G_{m-1} & G_m & & & \vdots \\ \vdots & \vdots & \ddots & & \vdots \\ G_1 & G_2 & \dots & G_m & 0 \end{pmatrix} \text{ if } N > m \tag{6}$$

Let us describe now the decoding algorithm for convolutional codes.

Algorithm 2. *Let $v = (v_0, v_1, \dots)$ be the received sequence, we want to obtain the decoded sequence $\hat{c} = (\hat{c}_0, \hat{c}_1, \dots)$ and the corresponding information sequence $\hat{u} = (\hat{u}_0, \hat{u}_1, \dots)$. For all calculations we assume that the vectors with negative index are 0.*

1. *Initialize the sequences $\hat{u} = ()$, $\hat{c} = ()$ and $i = 0$.*
2. *While $iL - 1 < \mathrm{length}(v)$ repeat:*
   - 2.1. *Let $\tilde{v} := (v_{iL}, \dots, v_{iL+N-1})$, and let $\tilde{w} := \tilde{v} - (\tilde{u}_{iL-m}, \dots, \tilde{u}_{iL-1})\widetilde{\mathcal{G}_N}$.*
   - 2.2. *Decode $\tilde{w}$ with respect to the code $\mathcal{C}_N$ to obtain $\hat{w}$.*
   - 2.3. *Obtain $(\hat{u}_{iL}, \dots, \hat{u}_{iL+N-1})$ the information sequence corresponding to $\hat{w}$.*
   - 2.4. *Compute $(\hat{c}_{iL}, \dots, \hat{c}_{iL+N-1}) = \hat{w} + (\tilde{u}_{iL-m}, \dots, \tilde{u}_{iL-1})\widetilde{\mathcal{G}_N}$.*
   - 2.5. *Update $\hat{u}$ and $\hat{c}$ by appending $(\hat{u}_{iL}, \dots, \hat{u}_{(i+1)L-1})$ and $(\hat{c}_{iL}, \dots, \hat{c}_{(i+1)L-1})$ respectively at then ends of $\hat{u}$ and $\hat{c}$.*
   - 2.6. *Update i with $i + 1$*
3. *Return $\hat{u}$ and $\hat{c}$.*

Remark 3. The key step of algorithm is 2.2, and the usability of this algorithm over a particular class of convolutional codes depends on the existence of a practical decoding algorithm for the corresponding block code $\mathcal{C}_N$ correcting up to $\lfloor \frac{\tilde{d}}{2} \rfloor$ errors and allowing decoding errors only in the last $N-L$ components. In Section IV we will study in detail how it runs over a class of codes for which such block decoding algorithm exists.

Let us prove that the algorithm returns indeed the sent codeword as long as the error constraint is fulfilled.

Theorem 4. *The sequence $\hat{c}$ is a codeword from the convolutional code $\mathcal{C}$ and it is the closest one to the received*

sequence v.

Proof: Note that for all i

$$\begin{aligned}(\hat{c}_{iL},\dots,\hat{c}_{iL+N-1}) &= \hat{w} + (\tilde{u}_{iL-m},\dots,\tilde{u}_{iL-1})\widetilde{\mathcal{G}_N} \\ &= (\hat{u}_{iL},\dots,\hat{u}_{iL+N-1})\mathcal{G}_N + (\tilde{u}_{iL-m},\dots,\tilde{u}_{iL-1})\widetilde{\mathcal{G}_N} \\ &= (\tilde{u}_{iL-m},\dots,\tilde{u}_{iL-1},\hat{u}_{iL},\dots,\hat{u}_{iL+N-1})\begin{pmatrix}\widetilde{\mathcal{G}_N}\\ \mathcal{G}_N\end{pmatrix}\end{aligned}$$

and $\left(\begin{smallmatrix}\widetilde{\mathcal{G}_N}\\ \mathcal{G}_N\end{smallmatrix}\right)$ is a column block submatrix of the sliding generator matrix $\mathfrak{G}$. Further, the components $((\hat{u}_{(i+1)L},\dots,\hat{u}_{iL+N-1})$ that are discarded at the i-th step only affect to the components $((\hat{c}_{(i+1)L},\dots,\hat{c}_{iL+N-1})$, which are also discarded at that step. Hence all the resulting sequence $\hat{c}$ can be obtained by the product of the sequence $\hat{u}$ with the sliding matrix $\mathfrak{G}$ and as a consequence $\hat{c}$ belongs to the code.

The fact that $\hat{c}$ is the closest codeword to v is a direct consequence of Lemma 1, since if the error constraint is fulfilled then for any $\tilde{w}$ returned in Step 2.2 of the algorithm the first L components, which are the only ones kept to update $\hat{u}$ and $\hat{c}$, are uniquely determined. □

III. DOUBLY CYCLIC CONVOLUTIONAL CODES

As any other kind of algorithm, the one presented here for decoding convolutional codes has some interest beyond theory only if it is usable in practice, i.e., if there exist codes for which the algorithm is efficient. Similarly, the importance of codes in practice relies upon the existence of a practical decoding algorithm.

We present in this Section the family of doubly cyclic convolutional codes [2] that, as we will see in the following Section, can be efficiently decoded by means of our algorithm.

In [1] the notion of cyclicity is studied for convolutional codes. This is done by considering the polynomial ring $A[z]$, with $A = \mathbb{F}[x]/ < x^n - 1 >$, as well as the natural isomorphisms

$$\begin{array}{ccc}\mathbb{F}^n & \overset{\mathfrak{p}}{\underset{\mathfrak{v}}{\rightleftarrows}} & A \\ (v_0,\dots,v_{n-1}) & \longleftrightarrow & \sum_{i=0}^{n-1} v_i x^i\end{array} \quad . \tag{7}$$

However, the natural step of defining cyclic convolutional codes as ideals of $A[z]$ leads to the fact that in that case the only cyclic codes would have zero memory, i.e., they would be block codes. Consequently a more general notion of cyclicity is needed. With this purpose an automorphism $\sigma \in Aut_{\mathbb{F}}(A)$ is also considered, and a multiplication in $A[z]$ defined by

$$az = z\sigma(a) \ \text{ for all } a \in A \tag{8}$$

$A[z]$ has the structure of a ring (only commutative when $\sigma = Id$) denoted by $A[z;\sigma]$, and it turns out that A and $\mathbb{F}[z]$ are subrings of $A[z;\sigma]$. Hence, $A[z;\sigma]$ inherits the $\mathbb{F}[z]$-module structure from $A[z]$, and the map

$$\mathfrak{p} : \mathbb{F}[z]^n \longrightarrow A[z;\sigma] \tag{9}$$

gives an isomorphism of left $\mathbb{F}[z]$-modules. Then, a convolutional code $\mathcal{C} \subseteq \mathbb{F}[z]^n$ is said to by σ-cyclic if $\mathfrak{p}(\mathcal{C})$ is a left ideal in $A[z;\sigma]$.

A particular subfamily of cyclic convolutional codes is studied in [2] where the chosen automorphism of A is of the form $\sigma = \alpha^k x$, with α a primitive element of $\mathbb{F}$ and $0 \le k \le n-1$. The cyclic convolutional codes of the family are the left ideals of $A[z;\sigma]$ generated by polynomials of the form $g = \sum_{i=0}^{m} z^i \sigma^i(f)$, where $f = \prod_{j=0}^{n-k-1}(x - \alpha^i)$. Note that f is the generator polynomial of a k-dimensional cyclic block code. It is because of this additional cyclic component that the codes so obtained are called *doubly-cyclic convolutional codes*.

The properties of these codes have been studied, and in particular the following results have been proven ([2]).

Theorem 5. *Let $n = q-1$, $k \le \lfloor n/2 \rfloor$, $m \le \lfloor n/k \rfloor - 1$, and let $\mathcal{C}$ denote the doubly-cyclic convolutional code generated by the polynomial g. Then,*

1) *$\mathcal{C}$ is generated by the matrix $G = \sum_{i=0}^{m} G_i z^i \in \mathbb{F}[z]^{k\times n}$, where*

$$G_i = \begin{pmatrix}\mathfrak{v}(\sigma^i(f))\\ \mathfrak{v}(\sigma^i(xf))\\ \vdots \\ \mathfrak{v}(\sigma^i(x^{k-1}f))\end{pmatrix} \in \mathbb{F}^{k\times n}. \tag{10}$$

G is both a basic and reduced polynomial matrix, and all Forney indices are equal to the memory of the code, m.

2) *$\mathcal{C}$ has free distance*

$$d_{free}(\mathcal{C}) = (m+1)(n-k+1). \tag{11}$$

3) *The codes generated by the matrices*

$$G_{j,0} = \begin{pmatrix}G_j\\ G_{j-1}\\ \vdots \\ G_0\end{pmatrix} \in \mathbb{F}^{(j+1)k\times n} \tag{12}$$

are Reed-Solomon codes.

As it will be apparent in next Section, the third statement of the Theorem is the key to adapt our general decoding algorithm to doubly-cyclic convolutional codes.

IV. DECODING DOUBLY-CYCLIC CONVOLUTIONAL CODES

As we saw in the description of the general decoding algorithm, the fundamental step is the block decoding of a given vector with respect to the code $\mathcal{C}_N$ so that $\lfloor \frac{d}{2} \rfloor$ errors can be corrected and only the first L components are needed to be correctly decoded.

Note that in the case of doubly-cyclic convolutional codes each column block submatrix of $\mathcal{G}_N$ is a matrix $G_{i,0}, i \le N$–1, which generates a Reed-Solomon code by Theorem 5.3, and there are efficient decoding methods for decoding such codes. We will make use of this fact to describe the way in which Step 2.2 of Algorithm 2 is carried out.

First, we need to fix N and L.

The length for the decoding window that we will be considering is $N = m + 1$, while we will take $L = 1$ to maximize the number of correctable errors at a decoding window.

Remark 6. Since the matrices $G_{i,0}$ generate Reed-Solomon codes, and they have therefore full row rank, the code $\mathcal{C}_{m+1}$ (generated by the matrix $\mathcal{G}_{m+1}$) has the property that for every codeword $c = (c_0, \ldots, c_m)$ with $c_0 = \ldots = c_{L-1} = 0$ and $c_L \neq 0$, then $c_j \neq 0$ for all $j > L$. As a consequence, $\mathrm{w}(c) \geq \sum_{i=0}^{m-L} d_i$, where $d_i = n - (i+1)k + 1$ is the minimum distance of the Reed-Solomon code generated by $G_{i,0}$.

Reciprocally, if $c \in C_{m+1}$ and $\mathrm{w}(c) \leq \sum_{i=0}^{m-L+1} d_i - 1$ then $c_0 = \ldots = c_{L-1} = 0$. Then for such L we can take $\tilde{d} := \sum_{i=0}^{m-L+1} d_i - 1$ to be in the conditions of Lemma 1.

We are now ready to describe the block decoding algorithm that will perform the fundamental part in the decoding of doubly-cyclic convolutional codes via Algorithm 2.

Algorithm 7. *Let $v = (v_0, \ldots, v_m) \in \mathbb{F}^{(m+1)n}$ be any vector. The algorithm looks for any codeword $c \in \mathcal{C}_{m+1}$ such that $d(c, v) \leq \tilde{d}$. As pointed out in the previous Remark, all such codewords will coincide in their first $L = 1$ components. Hence we will only ask the algorithm to return the first component of the codeword, c_0, and the corresponding information component $\hat{u}_0$.*

1. *Initialize $i = m + 1$,* (boolean)*STOP*=false.
2. *While STOP*=false *and $i > 0$ repeat:*
   - 2.1. *Update i with $i - 1$.*
   - 2.2. *Decode v_i with respect to the Reed-Solomon code generated by $G_{i,0}$ to obtain $\hat{c}_i$.*
   - 2.3. *If $d(v_i, \hat{c}_i) > \lfloor (d_i - 1)/2 \rfloor$, go to the beginning of the loop.*
     Else STOP=true. *Go on till the end of the loop.*
   - 2.4. *Obtain $(\hat{u}_0, \ldots, \hat{u}_i)$ the information sequence corresponding to $\hat{c}_i$ with respect to $G_{i,0}$.*
   - 2.5. *Compute*

$$(c_0, \ldots, c_i) = (\hat{u}_0, \ldots, \hat{u}_i) \begin{pmatrix} G_0 & G_1 & \ldots & G_i \\ & G_0 & \ldots & G_{i-1} \\ & & \ddots & \vdots \\ & & & G_0 \end{pmatrix}$$

   - 2.6. *If*

$$d((c_0, \ldots, c_i), (v_0, \ldots, v_i)) > \lfloor (\sum_{j=0}^{i} d_j - 1)/2 \rfloor,$$

     STOP=false. *Go to the beginning of the loop.*
3. *If $i < 0$, then perform repeatedly list decoding with respect to G_0 to obtain a codeword $(c_0, \ldots, c_m) = (\hat{u}_0, \ldots, \hat{u}_m)\mathcal{G}_m$ closest to v.*
4. *Return c_0 and $\hat{u}_0$.*

Remark 8. The algorithm can be easily generalized for arbitrary values of $L < N$.

V. CONCLUSIONS AND FUTURE WORK

The general decoding algorithm exposed in Section II becomes attractive since, as shown in Section IV, there is a family of codes for which it can be used in practice. The adaptation of the general algorithm to this particular class of codes is based on the decoding of Reed-Solomon codes, for which efficient algorithmS are well-known. As a consequence the convolutional decoding of such codes by means of the presented algorithm is shown to be feasible in practice. In addition, also the family of doubly-cyclic codes gains in interest due to the possibility of practical decoding.

A natural expansion of this work would be to find other families of convolutional codes for which we could adapt the general algorithm. This involves the need for a block decoding algorithm that corrects up to $\lfloor \tilde{d}/2 \rfloor$ errors while imposing the need of correct decoding only on the first L components. A possible approach on this direction would be to investigate the use of partial decoding algorithms.

VI. ACKNOWLEDGMENTS

Research by U.H. has been partially supported by grant HE 1858/12-1 within the DFG SPP 1305. Research by J.I.I.C. has been partially supported by Junta de Castilla y León through research project SA029A08.

REFERENCES

[1] H. Gluesing-Luerssen and W. Schmale, *On cyclic convolutional codes*, Acta Applicandae Mathematicae (2004), no. 82, 183–237.

[2] H. Gluesing-Luerssen and W. Schmale, *On doubly-cyclic convolutional codes*, Appl. Algebra Engrg. Comm. Comput. (2006), no. 17, 151–170.

[3] U. Helmke H. Gluesing-Luerssen and J. I. Iglesias Curto, *Algebraic decoding for doubly cyclic convolutional codes*, Advances in Mathematics of Communications **4** (2010), no. 1, 83–99.

[4] José Ignacio Iglesias Curto, *AAECC 2009*, ch. On Elliptic Convolutional Goppa Codes, pp. 83–91.

[5] José Ignacio Iglesias Curto, *Generalized AG convolutional codes*, Advances in Mathematics of Communications **3** (2009), no. 4, 317–328.

[6] José Ignacio Iglesias Curto and Uwe Helmke, *Receding horizon decoding of convolutional codes*, Preprint, 2009.

[7] R. Johannesson and K. Sh. Zigangirov, *Fundamentals of convolutional coding*, IEEE Press, New York, 1999.

[8] J. Justesen, *Algebraic construction of rate $1/\nu$ convolutional codes*, IEEE Trans. Inform. Theory **IT-21** (1975), 577–580.

[9] J.M. Muñoz Porras, J.A. Domínguez Perez, J.I. Iglesias Curto, and G. Serrano Sotelo, *Convolutional Goppa codes*, IEEE Trans. Inform. Theory **52** (2006), no. 1, 340–344.

[10] P. Piret, *Structure and constructions of cyclic convolutional codes*, IEEE Trans. Inform. Theory **IT-22** (1976), 147–155.

[11] Joachim Rosenthal, *Some interesting problems in systems theory which are of fundamental importance in coding theory*, Proceedings of the 36th IEEE Conference on Decision and Control.

[12] Joachim Rosenthal, *Dynamical systems, control, coding, computer vision: New trends, interfaces, and interplay*, ch. An algebraic decoding algorithm for convolutional codes, pp. 343–360, Birkuser, 1999.

[13] J. Rosenthal, J. M. Schumacher, and E. V. York, *On behaviors and convolutional codes*, IEEE Trans. Inform. Theory **42** (1996), no. 6, 1881–1891.

[14] J. Rosenthal and R. Smarandache, *Construction of convolutional codes using methods from linear systems theory* (1997), 953–960.

[15] R. Smarandache, J. Rosenthal, and H. Gluesing-Luerssen, *Constructions of MDS-convolutional codes*, IEEE Transactions on Information Theory **47** (2001), no. 5, 2045–2049.

[16] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Trans. Inform. Theory **13** (1967), no. 2, 260–267.