Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream

Mélanie Marazin, Roland Gautier, Gilles Burel

Algebraic method for blind recovery of punctured convolutional encoders from an erroneous bitstream

Mélanie Marazin[1-2] Roland Gautier[1-2] Gilles Burel[1-2]
[1]Université Européenne de Bretagne, France.
[2]Université de Brest; CNRS, UMR 3192 Lab-STICC, ISSTB, 6 avenue Victor Le Gorgeu, CS 93837, 29238 Brest cedex 3, France

***Abstract*—To enhance the quality of transmissions, all digital communication systems use error-correcting codes. By introducing some redundancy in the informative binary data stream, they allow one to better withstand channel impairments. The design of new coding schemes leads to a perpetual evolution of the digital communication systems and, thus, cognitive radio receivers have to be designed. Such receivers will be able to blind estimate the transmitter parameters. In this study, an algebraic method dedicated to the blind identification of punctured convolutional encoders is presented. The blind identification of such encoders is of great interest, because convolutional encoders are embedded in most digital transmission systems where the puncturing principle is used to increase the code rate in order to reduce the loss of the information data rate due to the redundancy introduced by the encoder. After a brief recall of the principle of puncturing codes and the construction of the equivalent punctured code, a new method dedicated to the blind identification of both the mother code and the puncturing pattern is developed when the received bits are erroneous. Finally, case-studies are presented to illustrate the performances of our blind identification method.**

I. Introduction

In most digital transmission systems, error-correcting codes are used for enhancement of the communication quality. These codes introduced some redundancy in the informative binary data stream to better withstand channel impairments. In this paper the problem of the blind identification of error correcting codes is treated. This problem has for a long time been reserved for military applications such as passive listening. In such a context, the adversary has only access to the intercepted noise bits stream with no knowledge of the parameters of the code. Therefore, these parameters must be blindly estimated. Such methods are also used in cryptanalysis systems. In this context, the objective is to recover the original message. But, many methods are based on the hypothesis that they have access to a decoded message. So, to obtain this decoded message it is necessary to know the encoder. Moreover, in [1], the author proposed a library which allows a good security level for peer to peer data transmission using convolutional codes punctured (or not) with controlled additive noise.

Currently, with the aim of enhancing the quality of transmission, new coding schemes are constantly being developed. In such a context, it is more and more difficult for users to follow all the changes to stay up-to-date and also to have an electronic communication device which is always compatible with every standard in use all around the world. Consequently, it becomes necessary to design cognitive receivers. In literature a lot of works deal with the domain of cognitive radio. Generally, a cognitive radio device has the ability to dynamically select their configuration parameters, on the transmitter side. Thus, a cognitive transmitter is able to adapt the encoder to the transmission channel, among a set of encoders. Here, the problem of a self-reconfigurable cognitive receiver is treated. Such receivers will be able to blindly estimate the transmitter parameters simply from the knowledge of the received data.

Email: {melanie.marazin, roland.gautier, gilles.burel}@univ-brest.fr

In this paper, the blind identification problem of error-correcting codes is considered for cognitive radio receiver design. Here, among the error-correcting codes, the blind recovery of convolutional encoders is dealt with. In a noisy environment, the first approach to identify the parameters of a code is related in [2]. At the same time, methods to recover a block code are developed in [3], [4] whereas [5] deals with to the blind identification of linear scramble. In [6], an iterative algorithm dedicated to the blind identification of a rate $(n-1)/n$ convolutional encoder is presented. In [7], [8], the authors are interested in the identification of the interleaver of a turbo-code. An algorithm to decide if the sequence is coded by a linear code is presented in [9].

The redundancy introduced by a convolutional encoder produces a decrease in the transmission rate. A simple technique, called puncturing, allows an increase in the code rate. It consists of deleting some symbols from an encoded word. This technique is usually used in digital communication systems. In a noisy environment (i.e when the received bits are erroneous), this paper deals with the blind identification of the punctured convolutional encoder. The first approaches developed to recover the punctured code in a noiseless context were proposed in [10], [11]. The new method presented in [12] deals with the specific case of a rate $1/n$ mother code in a noisy environment. Here, this paper describes a new iterative method for blind recognition of a rate k/n punctured convolutional code. In this context, the blind identification of a punctured code corresponds to the blind identification of the mother code and the puncturing pattern.

This paper is organized as follows. In Sect. II, the principle of punctured convolutional encoders is explained. The description of the blind recognition of this punctured code is developed in Sect. III. Finally, the performances of the blind identification method are discussed in Sect. IV. Conclusions and prospects are drawn in Sect. V.

II. Punctured convolutional code

The concept of punctured convolutional code was introduced by Cain [13] in 1979. Such a high-rate code is obtained

through a periodic elimination of specific code symbols at the output of a low-rate encoder. One should note that it depends on the low-rate code, that is, mother code and of both the number and positions of the puncturing symbols. The pattern of punctured symbols is called the puncturing pattern of the punctured code, and it is described in matrix form called the puncturing matrix and denoted as P.

A. General notations of convolutional encoders

The mathematical notions of convolutional encoders are briefly recalled. A convolutional encoder, denoted $C(n,k,K)$, is described by three parameters: n is the number of outputs, k is the number of inputs, and K is the constraint length. The $(k \times n)$ polynomial generator matrix of the convolutional encoder, denoted $G(D)$, is defined by

$$G(D) = \begin{bmatrix} g_{1,1}(D) & \cdots & g_{1,n}(D) \\ \vdots & \cdots & \vdots \\ g_{k,1}(D) & \cdots & g_{k,n}(D) \end{bmatrix} \quad (1)$$

where $g_{i,j}(D)$, $\forall i = 1, \cdots, k$, $\forall j = 1, \cdots, n$, are generator polynomials and D represents the delay operator. The encoding process can be described by

$$c(D) = m(D).G(D) \quad (2)$$

where $m(D)$ is the input sequence and $c(D)$ is the output sequence. For more details of the convolutional encoder, see [14], [15], [16].

B. Puncturing principle

The principle of a convolutional encoder is to produce n output bits for k input bits. On condition that both transmit M k-bits information words and receive M n-bits codewords at the output one would pass from a $C(n,k,K)$ code, called the mother code, to the Mth blocking code of C, denoted as $C^{[M]}(M.n, M.k, K)$, which is equivalent to C. The puncturing process consists of deleting a few bits from the codewords through use of the puncturing matrix (P), which is an $(n \times M)$ binary matrix with a total of N ones and $(n.M - N)$ zeros where ones correspond to the transmitted bits and zeros to the deleted ones. Application of this puncturing pattern to the $C^{[M]}(M.n, M.k, K)$ code leads to the $C_p(n_p, k_p, K_p)$ code, called the *equivalent punctured code*, where $n_p = N$ and $k_p = k.M$. This is exemplified hereafter.

Example *1:* Let us consider the encoder for the $C(2,1,K)$ convolutional encoder. The coding and puncturing processes can be represented as follows

$$\begin{pmatrix} c_1^0 & c_1^1 & c_1^2 & c_1^3 & \cdots \\ c_2^0 & c_2^1 & c_2^2 & c_2^3 & \cdots \end{pmatrix} \Rightarrow \begin{pmatrix} c_1^0 & c_1^1 & c_1^2 & c_1^3 & \cdots \\ c_2^0 & & c_2^2 & & \cdots \end{pmatrix} \quad (3)$$

where c_j^t is the bit of the output, j, encoded at the time, t. Using the puncturing pattern given in (4)

$$P = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \quad (4)$$

leads to a new encoder of rate $r_p = k_p/n_p$ where $k_p = k.M = 2$ and $n_p = N = 3$.

■

C. Equivalent punctured code

As shown in [17], the equivalent punctured convolutional encoder can be described by a simple convolutional encoder, $C_p(n_p, k_p, K_p)$; this equivalent punctured code is defined by a generator matrix, $G_p(D)$, of size $(k_p \times n_p)$. This code is equivalent to the mother code and the puncturing pattern, $C(n,k,K)$ and P respectively. A high-rate equivalent punctured code can be built from an original rate convolutional code through the simple process detailed hereafter. But prior to obtaining this equivalent punctured encoder, $C_p(n_p, k_p, K_p)$ code, it is worth recalling some definitions necessary to understand the constructing of this punctured code; moreover, additional information is available in [15], [18].

***Definition** 1: If $a(D) = a_0 + a_1.D + a_2.D^2 + \cdots$ is a polynomial in the indeterminate D, then for any integer $M > 1$, the Mth polyphase decomposition of $a(D)$ is the list of (i,M)th polyphase components ($\forall i = 0, \cdots, M-1$). Let us denote by $q_i(D)$ the (i,M)th polyphase component of $a(D)$ such that*

$$q_i(D) = D^{-i/M} a_{[i]_M}(D^{1/M}), \quad (5)$$

where $[i]_M$ (with i and M are integers) is the congruence class of i (modulo M), that is the set of integers of the from $i + l.M$ for $l = -\infty, \cdots, -1, 0, 1, \cdots, +\infty$. Finally, let $a_{[i]_M}(D^r)$ be the polynomial issued from $a(D)$ by selecting only the $[i]_M$ degree terms and then substituting D^r for D.

Example *2:* For illustration, let us consider the generator polynomial $a(D)$

$$a(D) = 1 + D^2 + D^3 + D^4 + D^6 \quad (6)$$

For $M = 3$, the congruence class of i (modulo M) is

i	l	$[i]_3$
0	$0, 1, 2$	$0, 3, 6$
1	$0, 1$	$1, 4$
2	$0, 1$	$2, 5$

The polynomial $a(D^{1/M})$ is defined by

$$a(D^{1/M}) = a_0 + a_2(D^{2/3}) + a_3(D^{3/3}) + a_4(D^{4/3}) + a_6(D^{6/3}) \quad (7)$$

and the polyphase component of $a(D)$ is

$$\begin{aligned} q_0(D) &= D^{-0/3}.a_{[0]_3}(D^{1/3}) \\ &= a_0 + a_3(D^{3/3}) + a_6(D^{6/3}) = 1 + D + D^2 \\ q_1(D) &= D^{-1/3}.a_{[1]_3}(D^{1/3}) \\ &= D^{-1/3}.\left(a_1(D^{1/3}) + a_4(D^{4/3})\right) = D \\ q_2(D) &= D^{-2/3} a_{[2]_3}(D^{1/3}) \\ &= D^{-2/3}\left(a_2(D^{2/3}) + a_5(D^{5/3})\right) = 1 \end{aligned} \quad (8)$$

■

***Definition** 2: Let $a(D) = a_0 + a_1.D + a_2.D^2 + \cdots$ be a polynomial in the indeterminate D and $(q_0(D), q_1(D), \cdots, q_{M-1}(D))$ be the Mth polyphase decomposition of $a(D)$. The Mth polycyclic pseudocirculant matrix (or PCPC for short) associated with $a(D)$ is then the*

$(M \times M)$ *polynomial matrix,* $Q^{[M]}(D)$*, such that*

$$Q^{[M]}(D) = \begin{bmatrix} q_0(D) & q_1(D) & \cdots & q_{M-1}(D) \\ D.q_{M-1}(D) & q_0(D) & \cdots & q_{M-2}(D) \\ \vdots & \vdots & \ddots & \vdots \\ D.q_1(D) & D.q_2(D) & \cdots & q_0(D) \end{bmatrix} \quad (9)$$

Example *3:* With the previous example 2, the third PCPC associated with $a(D)$ is such that

$$Q^{[3]}(D) = \begin{bmatrix} 1+D+D^2 & D & 1 \\ D & 1+D+D^2 & D \\ D^2 & D & 1+D+D^2 \end{bmatrix} \quad (10)$$

■

***Theorem** 1: If C is an (n,k) convolutional code, then the Mth blocking code of C, denoted by $C^{[M]}$, is an $(n.M, k.M)$ convolutional code. If $G(D)$ (1) is a $(k \times n)$ polynomial generator matrix for the original code C, then a generator matrix for $C^{[M]}$, say $G^{[M]}(D)$, can be obtained from $G(D)$ by substituting the corresponding Mth PCPC for each entry $g_{i,j}(D)$ from $G(D)$, and then interleaving the columns and lines at depth M.*

Example *4:* To illustrate the principle of the interleaving process, let us consider the matrix A

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix} \quad (11)$$

After interleaving the columns at depth 2, the matrix obtained is such that

$$A = \begin{pmatrix} a_{1,1} & a_{1,3} & a_{1,2} & a_{1,4} \\ a_{2,1} & a_{2,3} & a_{2,2} & a_{2,4} \\ a_{3,1} & a_{3,3} & a_{3,2} & a_{3,4} \\ a_{4,1} & a_{4,3} & a_{4,2} & a_{4,4} \end{pmatrix} \quad (12)$$

and after interleaving the lines at depth 2, the matrix is

$$A = \begin{pmatrix} a_{1,1} & a_{1,3} & a_{1,2} & a_{1,4} \\ a_{3,1} & a_{3,3} & a_{3,2} & a_{3,4} \\ a_{2,1} & a_{2,3} & a_{2,2} & a_{2,4} \\ a_{4,1} & a_{4,3} & a_{4,2} & a_{4,4} \end{pmatrix} \quad (13)$$

■

From the previous definitions, one can build the $C^{[M]}$ code, which has the same properties as the mother code. Its generator matrix, $G^{[M]}(D)$, is composed of $(k.M \times n.M)$ generator polynomials.

Example *5:* As illustration, let us consider the $C(2,1,3)$ mother code. The generator matrix of this code is: $G(D) = \begin{bmatrix} 1+D^2 & 1+D+D^2 \end{bmatrix}$. According to Definition 1, the Mth polyphase components of $g_{1,1}(D)$ and $g_{1,2}(D)$ for $M = 2$ are

$$\begin{cases} g_{1,1}(D) => (q_0(D), q_1(D)) = (1+D, 0) \\ g_{1,2}(D) => (q_0(D), q_1(D)) = (1+D, 1) \end{cases} \quad (14)$$

Let us denote by $Q^{[2]}_{1,1}$, the 2nd PCPC associated with $g_{1,1}(D)$ and by $Q^{[2]}_{1,2}$, the 2th PCPC associated with $g_{1,2}(D)$. According to Definition 2, these matrices are written as follows

$$Q^{[2]}_{1,1} = \begin{bmatrix} 1+D & 0 \\ 0 & 1+D \end{bmatrix} \quad \text{and} \quad Q^{[2]}_{1,2} = \begin{bmatrix} 1+D & 1 \\ D & 1+D \end{bmatrix} \quad (15)$$

So, the generator matrix of the $C^{[2]}$ code (see Theorem 1) is such that

$$G^{[2]}(D) = \begin{bmatrix} 1+D & 1+D & 0 & 1 \\ 0 & D & 1+D & 1+D \end{bmatrix} \quad (16)$$

■

***Definition** 3: On condition that $G(D)$ is a $(k \times n)$ polynomial matrix and that P is an $(n \times M)$ binary matrix, then the $n.M$ columns of the matrix, $G^{[M]}(D)$, are in natural one-to-one correspondence with the $n.M$ entries of P, and the matrix, $G_p(D)$, is the matrix issued from $G^{[M]}(D)$ after deletion of the columns corresponding to P entries. The code defined by the generator matrix, $G_p(D)$, is called the P-punctured version of C.*

Let ϕ be a bijection such that

$$(i,j) \rightarrow \phi(i,j) = i + n.(j-1) \quad (17)$$

To associate the $P(i,j)$ coefficient with the $\phi(i,j)$ column of $G^{[M]}(D)$ let us delete $G^{[M]}(D)$ columns according to P coefficients; it leads to the equivalent punctured convolutional code matrix, $G_p(D)$.

Example *6:* Further to the calculation of the matrix, $G^{[2]}(D)$, in example 5, let us assume that

$$P = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad (18)$$

The coefficient, $P(2,2)$, is equal to zero and corresponds to the third column of $G^{[2]}(D)$. Deleting this column leads to the generator matrix of the equivalent punctured code:

$$G_p(D) = \begin{pmatrix} 1+D & 1+D & 1 \\ 0 & D & 1+D \end{pmatrix} \quad (19)$$

The equivalent punctured code rate is $r_p = 2/3$ and the constraint length is $K_p = 2$.

■

III. Blind recovery of a punctured convolutional code

This part deals with the blind identification of the punctured code in a noisy environment. For that, the puncturing principle is summarized in Fig. 1.

Fig. 1. Puncturing principle

In this paper, the method dedicated to the blind identification of punctured convolutional encoder consists of identifying both the mother code and the puncturing pattern from the only knowledge of the equivalent punctured encoder.

In literature, few methods deal with the identification of punctured convolutional encoders and most of them [19], [20] allow the identification of the equivalent punctured code

$C_p(n_p, k_p, K_p)$ and not both the mother code and the puncturing pattern. But the decoding of the received sequence through use of this equivalent punctured code is not optimal. Indeed, the cost of the Viterbi decoding algorithm is proportional to the code rate. The objective is now to identify both the mother code and the puncturing pattern simply from the knowledge of the equivalent punctured encoder.

A first approach to recover a mother code and puncturing pattern was reported in [10], [11]. In these papers, among the numerous hypotheses made, the authors assumed that the mother code rate was only equal to $1/2$, and the puncturing pattern was such that

$$P = \begin{pmatrix} 1 & \cdots & 1 & 1 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \tag{20}$$

The approach reported in [12] dealt with the case of a rate $1/n$ mother code. The method proposed here is aimed at identifying both the mother code and the puncturing pattern in any case on the one condition that the equivalent punctured convolutional code, $C_p(n_p, k_p, K_p)$, is known. As this equivalent encoder can be described by a simple convolutional encoder, it can be estimated by the methods dedicated to the blind identification of a convolutional encoder. Thereafter, we will assume that the equivalent encoder has been identified. In the literature, some methods are available for blind identification in a noisy environment. For example, in the case of a rate $1/2$ convolutional encoder, a method was proposed in [21]. At nearly the same time, an algorithm was developed in [22] to identify a rate $1/n$ convolutional encoder. In [6], we proposed an iterative method of blind recovery for a convolutional encoder of rate $(n-1)/n$.

A. Blind identification of punctured convolutional code: principle

Assuming that the equivalent punctured convolutional encoder is identified, the aim is now to get the mother code and the puncturing matrix from the knowledge of the equivalent punctured convolutional encoder matrix, $G_p(D)$, alone.

It was shown in [17], how to get the generator matrix from the generator of an equivalent punctured encoder. Let us write the matrix, β, so that

$$\beta = \begin{pmatrix} M & M+1 & \cdots & M+(M-1) \\ M-1 & M & \cdots & M+(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \cdots & M \end{pmatrix} \tag{21}$$

From (9) let us rewrite the Mth PCPC of the (i,j)th generator polynomial so that

$$Q_{i,j}^{[M]} = \begin{bmatrix} q_{i,j_{(1,1)}}(D) & q_{i,j_{(1,2)}}(D) & \cdots & q_{i,j_{(1,n)}}(D) \\ \vdots & \vdots & \cdots \vdots & \\ q_{i,j_{(k,1)}}(D) & q_{i,j_{(k,2)}}(D) & \cdots & q_{i,j_{(k,n)}}(D) \end{bmatrix} \tag{22}$$

It follows that the (i,j)th generator polynomial, $g_{i,j}(D)$, is expressed by

$$g_{i,j}(D) = \sum_{m=1}^{M} D^{\beta(m,l)-M}.q_{i,j_{(m,l)}}(D^M) \quad \forall l = 1, \cdots, M \tag{23}$$

Example *7:* Let us consider again the example of the $C(2,1,3)$ code for $M = 2$. The Mth PCPC was given in (15). The matrix, β, for $M = 2$ is such that

$$\beta = \begin{pmatrix} 2 & 3 \\ 1 & 2 \end{pmatrix} \tag{24}$$

According to (23), for $l = 1$ and $l = 2$, the (i,j)th generator polynomial is given by

$$\begin{aligned} g_{i,j}(D) &= D^{\beta(1,1)-2}.q_{i,j_{(1,1)}}(D^2) + D^{\beta(2,1)-2}.q_{i,j_{(2,1)}}(D^2) \\ g_{i,j}(D) &= D^{\beta(1,2)-2}.q_{i,j_{(1,2)}}(D^2) + D^{\beta(2,2)-2}.q_{i,j_{(2,2)}}(D^2) \end{aligned} \tag{25}$$

Therefore, the generator polynomial $g_{1,1}(D)$, for $l = 1$ and $l = 2$ is

- $l = 1$:

$$g_{1,1}(D) = D^0.(1+D^2) + D^{-1}.(0) = 1 + D^2 \tag{26}$$

- $l = 2$:

$$g_{1,1}(D) = D^1.(0) + D^0.(1+D^2) = 1 + D^2 \tag{27}$$

and $g_{1,2}(D)$ is:

- $l = 1$:

$$g_{1,2}(D) = D^0.(1+D^2) + D^{-1}.(D^2) = 1 + D + D^2 \tag{28}$$

- $l = 2$:

$$g_{1,2}(D) = D^1.(D^2) + D^0.(1+D^2) = 1 + D + D^2 \tag{29}$$

■

Thus, from (23) one can identify the generator polynomial from a single column of its Mth PCPC.

B. Blind identification of a punctured convolutional code: procedure

In this part, the procedure to estimate the mother code and the puncturing pattern from the knowledge of the punctured code alone is explained. For that, a brief recall of the link between the generator matrix of the mother code $G(D)$ and its equivalent punctured code $G_p(D)$ is presented.

Let us denote by $G'(D)$ a matrix obtained from $G(D)$ by replacing each entry $g_{i,j}(D)$ from $G(D)$ with the corresponding Mth PCPC so that

$$G'(D) = \begin{bmatrix} Q_{1,1}^{[M]}(D) & Q_{1,2}^{[M]}(D) & \cdots & Q_{1,n}^{[M]}(D) \\ \vdots & \vdots & \cdots & \vdots \\ Q_{k,1}^{[M]}(D) & Q_{k,2}^{[M]}(D) & \cdots & Q_{k,n}^{[M]}(D) \end{bmatrix} \tag{30}$$

The matrix of the Mth blocking code of C, $G^{[M]}(D)$, is obtained from $G'(D)$ by interleaving the columns and rows at depth M. Finally, the matrix, $G_p(D)$ is obtained from

$G^{[M]}(D)$ by deleting the columns that correspond to P entries. This matrix is such that

$$G_p(D) = \begin{bmatrix} g_{p_{(1,1)}}(D) & \cdots & g_{p_{(1,n_p)}}(D) \\ \vdots & \dots & \vdots \\ g_{p_{(k_p,1)}}(D) & \cdots & g_{p_{(k_p,n_p)}}(D) \end{bmatrix} \quad (31)$$

where $g_{p_{(i,j)}}(D)$ is the (i,j)th generator polynomial of $G_p(D)$, $\forall i = 1, \cdots, k_p$ and $\forall j = 1, \cdots, n_p$.

This method is proposed for a rate $1/n$ mother code at first, and then for a rate k/n mother code.

1) Case of a rate $1/n$ mother code: In the case of a rate $1/n$ mother code, the high-rate code is k_p/n_p with $k_p = k.M = M$.

Let us, now, assume that the parameters of the equivalent punctured code, $C_p(n_p, k_p, K_p)$, and the generator matrix, $G_p(D)$, were all identified. Under this condition the matrix given in (30) is such that:

$$G'(D) = \begin{bmatrix} Q_{1,1}^{[M]}(D) & Q_{1,2}^{[M]}(D) & \cdots & Q_{1,n}^{[M]}(D) \end{bmatrix} \quad (32)$$

According to (31), each column of $G_p(D)$ is associated with only one column of a Mth PCPC. According to (23), we showed that the generator polynomial can be identified from only one column of its PCPC.

Our method of blind identification of a punctured code is summed up in Algo. 1.

In this algorithm, let us denote by $\hat{G}_p(D)$ the generator matrix of the equivalent punctured code so that: $\hat{G}_p(D) = G_p(D)$. Then let us take a first column of $\hat{G}_p(D)$ in order to build its generator polynomial (23), denoted by $q'(D)$ and such that

$$q'(D) = \sum_{m=1}^{M} D^{\beta(m,l)-M} \, . \, \hat{g}_{p_{(m,i)}}(D^M), \quad \forall i = 1, \cdots, n_p \quad (33)$$

where l lies within 1 and M. Let us, at first, apply (33) for $l = 1$ and see whether the result gives a generator polynomial. If it is not the case, the value of l ($\forall l = 1, \cdots, M$) is incremented to use (33) again. Once one generator polynomial, $q'(D)$, has been identified, its Mth PCPC, $Q'^{[M]}(D)$ is built. Then, for i between 1 and M, the ith column of $Q'^{[M]}(D)$ is compared with $\hat{G}_p(D)$ columns. Therefore, if the ith column of $Q'^{[M]}(D)$ corresponds to a column of $\hat{G}_p(D)$, this column is deleted, and the puncturing pattern is built by associating a 1 bit. In the reverse case, a 0 bit is associated to build the puncturing pattern. This algorithm given in Algo. 1 is iterated till there is no column in $\hat{G}_p(D)$.

Let us respectively denote by $\hat{g}_j(D)$ ($\forall j = 1, \cdots, n_p$) the generator polynomials, by $\hat{G}(D)$ the generator matrix of the mother code and by $\hat{P}$ the identified puncturing pattern. At the algorithm output, the $\hat{n}$ generator polynomials of the mother code and the puncturing pattern, $\hat{P}$, of size $(\hat{n} \times M)$ composed of n_p ones are identified.

Example *8:* As illustration of this blind recognition, let us consider the $C(3, 1, 7)$ mother code. The generator matrix of this code is

$$G = \begin{pmatrix} 171 & 165 & 133 \end{pmatrix}, \quad (34)$$

Algorithm 1: Blind identification of $\hat{G}(D)$ and $\hat{P}$

Input: The generator matrix $G_p(D)$ and M
Output: The generator matrix $\hat{G}(D)$ and $\hat{P}$

$\hat{G}_p(D) = G_p(D)$;
$\hat{P} = [\ ]$, $\hat{G}(D) = [\ ]$, $j = 0$;
while $\hat{G}_p(D) \neq \varnothing$ **do**
  $P' = [\ ]$;
  $l = 1$;
  while $l < M + 1$ **do**
    $q'(D) = \sum_{m=1}^{M} D^{\beta(m,l)-M} . \hat{g}_{p_{(m,1)}}(D^M)$;
    if $q'(D)$ *is a polynomial generator* **then**
      Build the Mth PCPC associated of $q'(D) => Q'^{[M]}(D)$;
      for $i = 1$ *to* M **do**
        if *the ith column of* $Q'^{[M]}(D) \in \hat{G}_p(D)$ **then**
          $P' = [P' \quad 1]$;
          This column is deleted from $\hat{G}_p(D)$;
        else
          $P' = [P' \quad 0]$
        end
      end
      $l = M + 1$;
    else
      $l = l + 1$
    end
  end
  $j = j + 1$;
  $\hat{g}_j(D) = q'(D)$;
  $\hat{P} = [\hat{P}; P']$;
end
$\hat{n} = j$;
$\hat{G}(D) = \{\hat{g}_j(D)\}_{\forall j = 1, \cdots, \hat{n}}$;

and the puncturing pattern for $M = 3$ is such that

$$P = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (35)$$

According to the previous definitions, the generator matrix of the $C_p(4, 3, 3)$ equivalent punctured code is

$$G_p = \begin{pmatrix} 7 & 6 & 0 & 4 \\ 2 & 5 & 7 & 4 \\ 2 & 2 & 3 & 7 \end{pmatrix} \quad (36)$$

The aim is now to identify the mother code and the puncturing pattern with this equivalent punctured code through use of our algorithm (Algo. 1). For $M = 3$, the matrix, β, is such that

$$\beta = \begin{pmatrix} 3 & 4 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 3 \end{pmatrix} \quad (37)$$

Let us set $\hat{G}_p(D) = G_p(D)$ and take into account the first

column of $\hat{G}_p(D)$. Application of (33) for $i = 1$ leads to

$$q'(D) = \sum_{m=1}^{M} D^{\beta(m,l)-M}.\hat{g}_{p_{(m,1)}}(D^M), \quad \forall l \in \{1, \cdots, M\} \tag{38}$$

- First step:

By using the first column of $\hat{G}_p(D)$ (36) and (38), one gets for $l = 1$

$$\begin{aligned} q'(D) &= D^0.(1 + D^3 + D^6) + D^{-1}.(D^3) + D^{-2}.(D^3) \\ &= 1 + D + D^2 + D^3 + D^6 \end{aligned} \tag{39}$$

The PCPC associated to $q'(D)$ is such that

$$Q'^{[3]}(D) = \begin{bmatrix} 1 + D + D^2 & 1 & 1 \\ D & 1 + D + D^2 & 1 \\ D & D & 1 + D + D^2 \end{bmatrix} \tag{40}$$

As columns 1 and 3 of $Q'^{[3]}(D)$ correspond to columns 1 and 4 of $\hat{G}_p(D)$, the puncturing pattern and the new $\hat{G}_p(D)$ matrix obtained after deletion of columns 1 and 4 are, thus, such that

$$P' = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \tag{41}$$

$$\hat{G}_p(D) = \begin{bmatrix} 1 + D & 0 \\ 1 + D^2 & 1 + D + D^2 \\ D & D + D^2 \end{bmatrix} \tag{42}$$

It follows that the first generator polynomial and the puncturing pattern are such that

$$\begin{cases} \hat{g}_1(D) = 1 + D + D^2 + D^3 + D^6 \\ \hat{g}_1 = 171 \\ \hat{P} = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} \end{cases} \tag{43}$$

- Second step:

For $l = 1$, using the new $\hat{G}_p(D)$ matrix expressed in (42) together with (38) leads to

$$\begin{aligned} q'(D) &= D^0.(1 + D^3) + D^{-1}.(1 + D^6) + D^{-2}.(D^3) \\ &= D^{-1} + 1 + D + D^3 + D^5 \end{aligned} \tag{44}$$

In this case, $q'(D)$ is not a polynomial generator.

So, on condition that $l = 2$ is taken, one gets

$$\begin{aligned} q'(D) &= D^1.(1 + D^3) + D^0.(1 + D^6) + D^{-1}.(D^3) \\ &= 1 + D + D^2 + D^4 + D^6 \end{aligned} \tag{45}$$

which is a polynomial generator. The PCPC associated to $q'(D)$ is such that

$$Q'^{[3]}(D) = \begin{bmatrix} 1 + D^2 & 1 + D & 1 \\ D & 1 + D^2 & 1 + D \\ D + D^2 & D & 1 + D^2 \end{bmatrix} \tag{46}$$

Column 2 of $Q'^{[3]}(D)$ corresponds to column 1 of $\hat{G}_p(D)$. Thus, the puncturing pattern and the new $\hat{G}_p(D)$ matrix are such that

$$P' = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \tag{47}$$

$$\hat{G}_p(D) = \begin{bmatrix} 0 \\ 1 + D + D^2 \\ D + D^2 \end{bmatrix} \tag{48}$$

Thus, the second generator polynomial and the puncturing pattern are such that

$$\begin{cases} \hat{g}_2(D) = 1 + D + D^2 + D^4 + D^6 \\ \hat{g}_2 = 165 \\ \hat{P} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \end{cases} \tag{49}$$

- Third step:

Using the new $\hat{G}_p(D)$ expressed in (48) together with (38) leads, for $l = 1$, to

$$\begin{aligned} q'(D) &= D^0.(0) + D^{-1}.(1 + D^3 + D^6) + D^{-2}.(D^3 + D^6) \\ &= D^{-1} + 1 + D + D^2 + D^4 + D^5 \end{aligned} \tag{50}$$

Once again, $q'(D)$ is not a polynomial generator.

By taking $l = 2$, one gets

$$\begin{aligned} q'(D) &= D^1.(0) + D^0.(1 + D^3 + D^6) + D^{-1}.(D^3 + D^6) \\ &= 1 + D^2 + D^3 + D^5 + D^6 \end{aligned} \tag{51}$$

which is a polynomial generator. The PCPC associated to $q'(D)$ is such that

$$Q'^{[3]}(D) = \begin{bmatrix} 1 + D + D^2 & 0 & 1 + D \\ D + D^2 & 1 + D + D^2 & 0 \\ 0 & D + D^2 & 1 + D + D^2 \end{bmatrix} \tag{52}$$

As column 2 of $Q'^{[3]}(D)$ corresponds to column 1 of $\hat{G}_p(D)$, the resulting puncturing pattern and $\hat{G}_p(D)$ matrix are as follows

$$P' = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \tag{53}$$

$$\hat{G}_p(D) = \varnothing \tag{54}$$

The third generator polynomial and the puncturing pattern are, therefore, such that

$$\begin{cases} \hat{g}_3(D) = 1 + D^2 + D^3 + D^5 + D^6 \\ \hat{g}_3 = 133 \\ \hat{P} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{cases} \tag{55}$$

At the end of the three steps, one obtains

$$\hat{n} = 3, \tag{56}$$

$$\hat{G} = \begin{pmatrix} 171 & 165 & 133 \end{pmatrix} \tag{57}$$

and

$$\hat{P} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \tag{58}$$

It is worth noting that the estimated parameters correspond to the parameters of the mother code (34) and puncturing pattern (35).

■

2) Case of a rate k/n mother code: In the case of a rate k/n mother code, the first step of the blind identification is the finding of both the number of inputs, $\hat{k}$, and $\hat{M}$ value. As

$$k_p = \hat{k}.\hat{M} \tag{59}$$

k_p is known. All possible $(\hat{k}, \hat{M})$ pairs have, thus, to be tested. But as $\hat{k}$ and $\hat{M}$ are integers, the number of possible pairs is small. Moreover, the number of inputs of a convolutional encoder is very small, in practice $k < 5$. Thus, it is possible to limit the number of possible pairs by fixing a maximum value of k (e.g. $k_{max} = 5$). Then, the algorithm given in Algo. 1 is applied to each $(\hat{k}, \hat{M})$ pair.

In the case of a rate $1/n$ mother code, each column of $G_p(D)$ was associated with only one generator polynomial. But, in the general case, each column of $G_p(D)$ is associated with k generator polynomials and, thus, the line at depth, M, of $G_p(D)$ matrix has to be deinterleaved. Let us denote by $G'_p(D)$ the matrix issued from $G_p(D)$ by interleaving the rows at M depth

$$G'_p(D) = \begin{bmatrix} g'_{p_{(1,1)}}(D) & \cdots & g'_{p_{(1,n_p)}}(D) \\ \vdots & \cdots & \vdots \\ g'_{p_{(k_p,1)}}(D) & \cdots & g'_{p_{(k_p,n_p)}}(D) \end{bmatrix} \tag{60}$$

and by $G^j_p(D)$, k sub-matrices of size $(M \times n_p)$ $(\forall j = 1, \cdots, k)$ such that

$$G^j_p(D) = \begin{bmatrix} g'_{p_{((j-1).M+1,1)}}(D) & \cdots & g'_{p_{((j-1).M+1,n_p)}}(D) \\ \vdots & \cdots & \vdots \\ g'_{p_{(j.M,1)}}(D) & \cdots & g'_{p_{(j.M,n_p)}}(D) \end{bmatrix} \tag{61}$$

Application of our algorithm given in Algo. 1 to $G^j_p(D)$ leads to $\hat{n}$ generator polynomials, for each sub-matrix, and to an $\left(\hat{M} \times \hat{n}\right)$ puncturing pattern. Once the $\left(\hat{k} \times \hat{n}\right)$ generator polynomials are obtained, one needs to check that the identified $\hat{k}$ puncturing patterns are identical. At the algorithm output, one gets both the mother code generator matrix and the puncturing pattern.

Example *9:* For illustration, let us consider the $C(3, 2, 3)$ mother code. Puncturing at depth $M = 2$ so that

$$P = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \tag{62}$$

leads to the following equivalent punctured code

$$G_p = \begin{pmatrix} 3 & 2 & 1 & 0 & 0 \\ 0 & 3 & 3 & 0 & 2 \\ 1 & 0 & 0 & 2 & 1 \\ 1 & 0 & 1 & 3 & 3 \end{pmatrix} \tag{63}$$

Its rate is $r_p = 4/5$.

As the first step of our blind identification technique is the estimation of every possible pair $(\hat{k}, \hat{M})$, let us consider the case where $k_p = 4$. It leads to

$$\begin{cases} (\hat{k}, \hat{M}) \Rightarrow (1, 4) \\ (\hat{k}, \hat{M}) \Rightarrow (2, 2) \end{cases} \tag{64}$$

- $\hat{k} = 1$ and $\hat{M} = 4$

Running the algorithm given in Algo. 1 gives the following $C(3, 1, 6)$ mother code for the generator matrix

$$\hat{G} = \begin{pmatrix} 72 & 62 & 53 \end{pmatrix} \tag{65}$$

and the puncturing pattern is

$$\hat{P} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{66}$$

But according to the properties of an optimal convolutional encoder [14], the convolutional encoder described by the estimated generator matrix is not optimal.

Let us now consider the case where:

- $\hat{k} = 2$ and $\hat{M} = 2$

and denote by G'_p the matrix issued from G_p (63) once the rows at depth 2 have been deinterleaved

$$G'_p = \begin{pmatrix} 3 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 1 \\ 0 & 3 & 3 & 0 & 2 \\ 1 & 0 & 1 & 3 & 3 \end{pmatrix} \tag{67}$$

According to (61), the two matrices, G^1_p and G^2_p, are as follows

$$\begin{cases} G^1_p = \begin{pmatrix} 3 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 2 & 1 \end{pmatrix} \\ G^2_p = \begin{pmatrix} 0 & 3 & 3 & 0 & 2 \\ 1 & 0 & 1 & 3 & 3 \end{pmatrix} \end{cases} \tag{68}$$

Running the algorithm given in Algo. 1 for G^1_p leads to 3 polynomial generators, denoted by $\hat{g}_{1,j}$ $(\forall j = 1, \cdots, 3)$ and to the following puncturing pattern

$$\begin{cases} (\hat{g}_{1,1}, \hat{g}_{1,2}, \hat{g}_{1,3}) = (7, 4, 1) \\ \\ \hat{P} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \end{cases} \tag{69}$$

and for G^2_p to the 3 polynomial generators, $\hat{g}_{2,j}$ $(\forall j = 1, \cdots, 3)$, and to the puncturing pattern

$$\begin{cases} (\hat{g}_{2,1}, \hat{g}_{2,2}, \hat{g}_{2,3}) = (2, 5, 7) \\ \\ \hat{P} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \end{cases} \tag{70}$$

It is worth noting that the estimated puncturing patterns are a like. The generator matrix of the $C(3, 2, 3)$ mother code is built through use of the identified generator polynomials

$$\hat{G} = \begin{pmatrix} 7 & 4 & 1 \\ 2 & 5 & 7 \end{pmatrix} \tag{71}$$

According to the properties of a convolutional encoder, this generator matrix describes an optimal convolutional encoder. Moreover, the identified parameters correspond to the parameters of the mother code and puncturing pattern.

■

IV. Analysis and performances: punctured code

On condition that the equivalent punctured convolutional code is known, the method proposed in this paper is dedicated to the blind identification of both the mother code and the puncturing pattern for any case (rate k_p/n_p). In this section, in order to analyse our blind identification method (which includes the identification of the equivalent punctured code, the mother code and the puncturing pattern), the rate of the equivalent punctured code is assumed to be equal to $(n_p-1)/n_p$. Since, the blind identification of the convolutional encoder of rate (k/n) is still under study. However, it is very important to note that the method dedicated to the blind identification of the punctured code works in any case (rate k_p/n_p) on condition that the equivalent code has been identified. Therefore, the iterative method proposed in [6] can be used to identify the equivalent punctured code of rate $(n_p-1)/n_p$. This analysis of the performances is proposed for two punctured convolutional encoders given in Table I.

TABLE I
Punctured convolutional encoders

Mother code	Punctured code		
$C(n,k,K)$	M	P	$C_p(n_p,k_p,K_p)$
$C(2,1,7)$	2	$\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$	$C_p(3,2,4)$
$C(3,2,3)$	2	$\begin{pmatrix}1 & 0\\ 1 & 1\\ 1 & 1\end{pmatrix}$	$C_p(5,4,2)$

The method of blind identification of a punctured encoder is divided into two parts:

1. *Identification of the equivalent punctured code:* algorithm presented in [6] for a rate $(n_p-1)/n_p$ of convolutional encoder;
2. *Identification of the mother code and puncturing pattern:* algorithm presented in Algo. 1.

For more details of the simulation parameters see [6]. Moreover, this paper [6] shows that 20 000 received bits are a relevant and sufficient number to achieve a high probability of detecting the true encoder.

For each simulation, 1000 Monte Carlos were run. To analyze the performances of the method dedicated to the blind identification of punctured codes, here, focus is on: i) the impact of the number of iterations upon the probability of detection and ii) the global performances in terms of probability of detection. One should note that, here, the probability of detection includes the complete identification of the punctured encoders, that is, the mother codes and the puncturing pattern.

A. Detection gain produced by the iterative process

In [6], an iterative process was proposed to increase the detection performances in the case of a rate $(n-1)/n$ convolutional encoder. Here, to evaluate the impact of the same iterative process in the case of the punctured code, let us denote by $\lambda_{x\to y}$ the gain (expressed in percent) between the xth iteration and the yth such that

$$\lambda_{x\to y}=\frac{P_{det}(y)-P_{det}(x)}{P_{det}(x)} \tag{72}$$

where $P_{det}(i)$ is the probability of detecting the true encoder at the ith iteration. The probability of detecting the true encoder, P_{det}, is called probability of detection.

Figs. 2 and 3 depict P_{det} against P_e, for 1, 10, 40 and 50 iterations for $C_p(5,4,2)$ and $C_p(3,2,4)$ equivalent punctured convolutional encoders, respectively. With both equivalent punctured encoders, 40 iterations permitted us to identify the true mother code and the puncturing pattern. Table II shows that the gain between the 40th and the 50th iterations is nearly null. It is worth noting that, to identify the true mother code and puncturing pattern, the punctured convolutional encoders require more iterations than the convolutional encoder (40 iterations against 10).

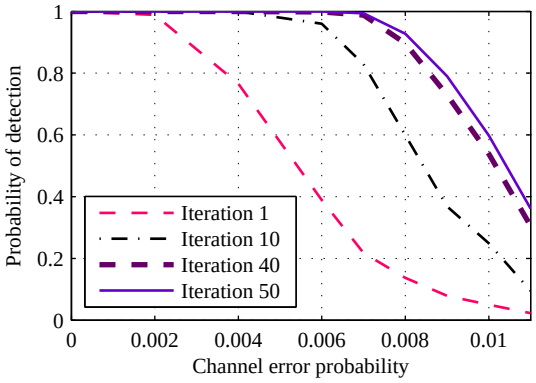

Fig. 2. $C_p(5,4,2)$ probability of detection against P_e

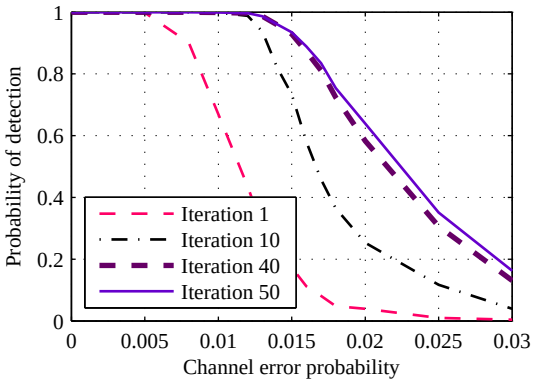

Fig. 3. $C_p(3,2,4)$ probability of detection against P_e

TABLE II
$C_p(5,4,2)$ and $C_p(3,2,4)$ detection gain

P_e	0.004	0.007	0.009
$C_p(5,4,2)$: $\lambda_{1\to10}$	30%	280%	365%
$C_p(5,4,2)$: $\lambda_{1\to50}$	30%	356%	900%
P_e	0.01	0.0015	0.02
$C_p(3,2,4)$: $\lambda_{1\to10}$	49%	324%	548%
$C_p(3,2,4)$: $\lambda_{1\to50}$	49%	436%	1394%

The performances obtained for the punctured codes are similar to those obtained for the $(n-1)/n$ convolutional codes presented in [6]. Indeed, it is clear that the algorithm performances are enhanced by iterations. Moreover, the number

of iterations needed to obtain the best performance is code-dependent: indeed, a punctured code requires more iterations than a convolutional code. In fact, the steps in the algorithm of identification of punctured codes are also more numerous. An equivalent punctured code is identified first, prior to the identification of the mother code and puncturing pattern.

B. Probability of detection

Three probabilities were defined in order to analyze the algorithm performances

1. The probability of identifying the true encoder denoted P_{det} (Probability of detection).
2. The probability of identifying an optimal encoder but not the true one denoted P_{fa} (Probability of false-alarm).
3. The probability of identifying no optimal encoder denoted P_m (Probability of miss).

In such a context, the true encoder represents the true mother code and the true puncturing pattern.

To evaluate the result of the blind identification method, a comparison between the detection probabilities and the code correction capability was proposed in [6]. Let us denote by BER_r the theoretical residual bit error rate obtained after decoding of the corrupted data stream with a hard decision, [14]. Here, the BER_r is considered as acceptable if it is close to 10^{-5}. Indeed, after this limit of BER_r, the decoded message is not correctly corrected for a civil application. In practice, this encoder will not be used for a channel error probability corresponding to this post-decoding bit error rate. Consequently, it is not necessary to identify the encoder when $BER_r > 10^{-5}$.

Figs. 4 and 5 present the different probabilities against P_e after 40 iterations, as well as the limit of the 10^{-5} acceptable BER_r for $C_p(5,4,2)$ and $C_p(3,2,4)$ equivalent punctured convolutional encoders, respectively. As previously, in the case of convolutional encoders, the probability of identifying the true encoder is close to 1 for any P_e with a post-decoding BER_r value less than 10^{-5}. With both punctured encoders, the algorithm also gives excellent results: P_{det} close to 1 with a P_e corresponding to $BER_r < 6 \times 10^{-4}$ ($P_e < 0.008$) for $C_p(5,4,2)$ punctured convolutional encoder and $BER_r < 1 \times 10^{-4}$ ($P_e < 0.015$) for the $C_p(3,2,4)$ one. Moreover, for a punctured code the probability of detecting an optimal encoder but not the true one is very small. Indeed, for $C_p(5,4,2)$ punctured convolutional encoder, this probability, P_{fa}, is zero.

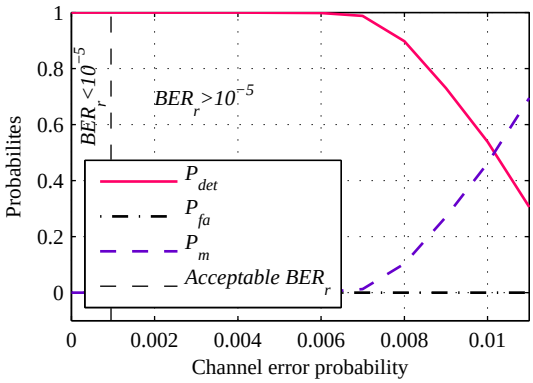

Fig. 4. $C_p(5,4,2)$: Probabilities

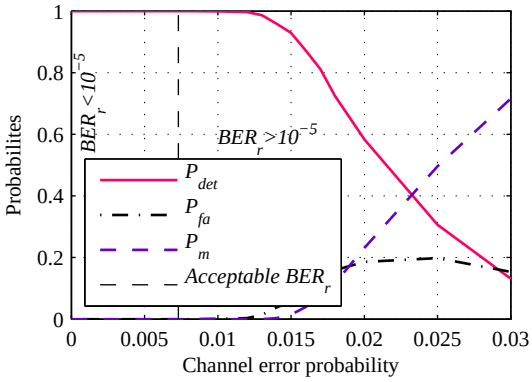

Fig. 5. $C_p(3,2,4)$: Probabilities

V. Conclusion

To conclude, this paper presented a new algorithm dedicated to the reconstruction of punctured convolutional code from received noisy data streams. This method allows one to estimate both the mother code and the puncturing pattern simply from the knowledge of the equivalent puncturing code (in the general rate case k_p/n_p). In a noisy environment, the equivalent punctured encoder can be estimated by an iterative method described in [6] in the case of rate $(n_p-1)/n_p$. In Sect. IV, an analysis of the blind identification method is proposed. For a rate $(n_p-1)/n_p$ of an equivalent punctured encoder, the performances of the method proved to be very good. Moreover, the probability to detect the true mother code and true puncturing pattern proved to be close to 1 for a channel error probability that generates a post-decoding BER_r less than 10^{-5}. In a cooperative context, the residual bit error rate obtained after decoding can be considered as acceptable if it is close to 10^{-5}.

Our future work will be to extend the method described in [6], dedicated to the blind identification of convolutional encoders of rate $(n-1)/n$, to the case of rate k/n. Thus, it will be possible to raise the hypothesis made in the analysis of the detection performances (see Sect. IV). In this case, the analysis of the global performances of our blind identification of a punctured code method will be proposed for any case rate of equivalent punctured codes.

Acknowledgements

This study was supported by the Brittany Region (France).

References

[1] Filiol E.: 'bibliothèque PERSEUS: protèger des communications avec du bruit'. GNU/Linux Magazine, 2011, 135, pp. 64-71.
[2] Barbier J., Sicot G. and Houcke S.: 'Algebraic approach for the reconstruction of linear and convolutional error correcting codes', Int. journal of applied mathematics and computer sciences, 2006, 2, (3), pp. 113-118.
[3] Cluzeau M. : 'Block code reconstruction using iterative decoding techniques'. Proc. IEEE Int. Symposium on Information Theory, Seattle, July USA, 2006, pp. 2269-2273.
[4] Cluzeau M. and Tillich J-P.: 'On the code reverse engineering problem'. Proc. IEEE Int. Symposium on Information Theory, Toronto, Canada, July 2008, pp. 634-638.

[5] Cluzeau M. : ‘Reconstruction of a linear scramble’. IEEE Trans. on Computers, 2007, 56, (8), pp. 1-8.

[6] Marazin M., Gautier R. and Burel G.: ‘Dual code method for blind identification of convolutional encoder for cognitive radio receiver design’, Proc. IEEE Broadband Wireless Access Workshop, IEEE GLOBECOM, Honolulu, Hawaii, USA, November 2009.

[7] Cluzeau M., Finiasz M. and Tillich J-P: ‘Methods for the reconstruction of parallel turbo codes’. Proc. IEEE Int. Symposium on Information Theory, Austin, Texas, USA, June 2010, pp. 2008-2012.

[8] Cote M. and Sendrier N.: ‘Reconstruction of a turbo-code interleaver from noisy observation’. Proc. IEEE Int. Symposium on Information Theory, Austin, Texas, USA, June 2010, pp. 2003-2007.

[9] Chabot C.: ‘Recognition of a code in a noisy environment’. Proc. IEEE Int. Symposium on Information Theory, Nice, France, June 2007, pp. 2211-2215.

[10] Li S., Lu P., Luo X. and Zou Y.: ‘Blind recognition of punctured convolutional codes’. Proc. IEEE Int. Symposium on Information Theory, Chicago, USA, July 2004, pp. 457.

[11] Lu P., Li S., Zou Y. and Luo X.: ‘Blind recognition of punctured convolutional codes’, Science in China Series F-Information Sciences, 2005, 48, (3), pp. 484-498.

[12] Cluzeau M. and Finiasz M.: ‘Reconstruction of punctured convolutional codes’. Proc. IEEE Information Theory Workshop, Seoul, Korea, June 2009, pp. 546-550.

[13] Cain J.B, Clark G.C. and Geist J.M.: ‘Punctured convolutional codes of rate (n-1)/n and simplified maximum likehood decoding’, IEEE Transactions on Information Theory, 1979, IT-25, (1), pp. 97-100.

[14] Johannesson R., and Zigangirov K.Sh.: ‘Fundamentals of Convolutional Coding’ (IEEE Press, 1999).

[15] McEliece R.J.: ‘The algebraic theory of convolutional codes’, in Handbook of coding theory, S. (Ed.): ‘V.S. Pless and W.C. Huffman’ (Elsevier, 1998), pp. 1065-1138

[16] Forney G. D.: ‘Convolutional codes I: Algebraic structure’, IEEE Transactions on Information Theory, 1970, 16, (6), pp. 720-738.

[17] Hole K.J.: ‘Punctured Convolutional Codes for the 1-D Partial-Reponse Channel’, IEEE Transactions on Information Theory, 1991, 37, (3), pp. 808-817.

[18] Hole K.J.: ‘Rate $k/(k+1)$ punctured convolutional encoders’, IEEE Transactions on Information Theory, 1991, 37, (3), pp. 653-655.

[19] Filiol E.: ‘Reconstruction of Punctured Convolutional Encoders’. Proc. Int. Symposium on Information Theory and Application, Hawaii, USA, November 2000, pp. 4-7.

[20] Barbier J.: ‘Analyse de canaux de communication dans un contexte non coopératif’. PhD thesis, Ecole Polytechnique, France, 2007.

[21] Wang F., Huang Z. and Zhou Y.: ‘A method for blind recognition of convolution code based on euclidean algorithm’. Proc. Int. Conf. Wireless Communications, Networking and Mobile Computing, Shanghai, China, Sept. 2007, pp. 1414-1417.

[22] Dingel J. and Hagenauer J.: ‘Parameter estimation of a convolutional encoder from noisy observation’. Proc. IEEE Int. Symposium on Information Theory, Nice, France, June 2007, pp. 1776-1780.