Blind detection of interleaver parameters

Guillaume Sicot, Sébastien Houcke, Johann Barbier

Blind detection of interleaver parameters

Guillaume Sicot [a,c], Sebastien Houcke [a,c,*], Johann Barbier [b]

[a] Institut TELECOM, TELECOM Bretagne, UMR CNRS 3192 Lab-STICC, Technopole Brest-Iroise, CS 83818 29285 Brest Cedex 3, France
[b] Cryptology Department, CELAR and the Virology and Cryptology Lab, ESAT, France
[c] Université Européenne de Bretagne, France

A B S T R A C T

Interleaving is a key component of many digital communication systems involving error correction schemes. It provides a kind of time diversity to protect the transmitted data against bursts of errors. Recently, interleavers have become an even more integral part of the code design itself, if we consider for example turbo and turbo-like codes. In a non-cooperative context, a passive adversary has to solve the problem of estimating the interleaver parameters. In this paper, we propose an algorithm that is able to estimate the size, the starting position (frame synchronization) of the interleaver, and some information about the interleaver function. This is accomplished blindly at the output of a binary symmetric channel (BSC). Moreover, an improvement of the proposed method is introduced when a soft information on the decided bits is available.

A theoretical analysis of the proposed technique is given. This allows us to express the optimal detection threshold and the theoretical probability of detection. This analysis gives us insight on the behavior of our method and allows us to improve our algorithm to get better performance. Some experimental results are run to validate the probability of success of our algorithm.

Keywords:
Passive listening
Communication interception
Interleaver
Blind detection

1. Introduction and notations

In a noisy communication system, the use of an error correcting code is mandatory. Most of them are efficient when the errors are randomly distributed but generally offer lower performance when the errors occur in bursts. For bursts of errors, interleavers are commonly used [1] for uniformly dispatching the error along the coded sequence. In such a scheme, the receiver has to demodulate the signal, synchronize the frame and deinterleave it before decoding the sequence and correct some transmission errors.

In a non-cooperative context, a passive adversary needs to have access to the information exchanged between legal users. In such a context, the adversary has no *a priori* knowledge about the parameters used for the communication. Therefore, he has to blindly estimate these parameters knowing only the intercepted signal. We make the hypothesis that the adversary is able to retrieve the interleaved coded sequence, i.e. he has already found the parameters of the demodulation and the parameters of the scrambler if one is used. Then, he has access to the noisy interleaved binary stream at the input of the scheme composed of the interleaver and the decoder. Without loss of generality, we consider that the channel is a binary symmetric channel (BSC). This can be justified by the fact that the interleaver tends to uniformly distribute the errors at the input of the decoder. In this paper, we take the place of the adversary and try to answer the following question: "given this binary stream, how to retrieve the parameters of the interleaver?". We present algorithms to

* Corresponding author at: Institut TELECOM, TELECOM Bretagne, UMR CNRS 3192 Lab-STICC, Technopole Brest-Iroise, CS 83818 29285 Brest Cedex 3, France. Tel.: +33 2 29 00 15 36; fax:+33 2 29 00 10 12.

E-mail addresses: guillaume.sicot@telecom-bretagne.eu (G. Sicot), sebastien.houcke@telecom-bretagne.eu (S. Houcke), johann.barbier@dga.defense.gouv.fr (J. Barbier).

find the parameters of an error correcting block code and of the interleaver. We also explain how to locate the codewords into an interleaved block.

The state-of-the-art techniques [2–5] consist in recovering in the same time the parameters of the encoder and of the interleaver. The classic way (detailed in Section 1.2) to achieve this is to look for a basis of parity checks of the code. Two strategies can be applied. The first one is to adapt algorithms to find codewords of small Hamming weight in random codes, such as [6–8]. This strategy was introduced by Planquette [9] and improved by Valembois and recently by Cluzeau [10]. Another strategy for recovering the encoder and the interleaver parameters is to directly apply a Gauss elimination process. It was first introduced by Burel and Gautier [11] for noiseless channels, generalized by Sicot and Houcke [2,3] for noisy channels and analyzed by Barbier et al. [4,12]. Both strategies are based on the *rank criteria* introduced by Valembois [13] and consist in looking for the parameters which minimize the rank of the *interception matrix*. The rows of this particular matrix are the noisy intercepted codewords. This matrix is defined in the next subsection.

Having received a block coded and interleaved binary sequence corrupted by a high bit error rate (BER), we design in this paper an algorithm based on the same concept as in [11] that blindly estimates

- the size of the interleaver,
- the position of the interleaver (frame synchronization),
- the code rate,
- the position of codewords in the interleaved sequence (this allows to get a precise idea of the type of the interleaver used).

Compared to a previous work [2], we compute the theoretical probability of detection of the interleaver parameters based on a rank criteria and derive the expression of the optimal detection threshold. This study allows us to clearly understand the reason for which the probability of detection decreases with the size of the interleaver. Furthermore, we develop a new version of the algorithm that takes advantage of the reliability of the estimated bits if available (if an AWGN channel is considered instead of a BSC). The paper is organized as follows. Section 2 presents the principle of the blind estimation of interleaver parameters based on the rank property of a specific matrix built from the intercepted binary sequence. We adapt the method initially introduced for error free sequence to a high noisy BSC. In Section 3, we develop the theoretical study of this kind of algorithms and give an analytical expression of the optimal threshold. Unfortunately, this expression depends on unknown parameters at the receiver side. However, taking advantage of the iterative procedure of our proposed method, we propose a practical way to adaptively set up the threshold. Section 4 presents a way to blindly estimate the interleaver function used. This is accomplished by locating codewords within the interleaved block. In Section 5, we show that the performance of estimation of the interleaver size depends on our capability to synchronize on the interleaver: we do not have the same performance if we start with the end or with the beginning of an interleaved block. This comprehension makes us change our algorithm to get even better identification performance. Finally, we detail in Section 6 the experimental results which validates our theoretical analysis and also illustrate the efficiency of our method even for a channel with a high BER. Finally, Section 7 concludes the work.

1.1. Notations

A block encoder is defined by a full-rank generator matrix G that transforms each block of k_c information bits into n_c encoded bits ($k_c < n_c$). Let vectors b_i and y_i denote the ith information block and the ith encoded block, respectively. $y_i = Gb_i$ is called a *codeword* and the ratio $r = k_c/n_c$ is called the *code rate*. The interleaver can be modeled by a permutation matrix Π of size $S \times S$ where S is the interleaver size. The interleaver performs a permutation within each block of S encoded bits. In almost all systems, the interleaver size is a multiple of the size of the codeword i.e. $S = N \times n_c$, where N is the number of codewords within the interleaved block. The transmitted sequence $\mathbf{X}$ is the concatenation of M interleaved blocks. Let us denote by $\mathbf{Z}$ the intercepted sequence of $\mathbf{X}$. As the adversary has no *a priori* knowledge about the transmission, he may miss the first t_0 bits. Then, $\mathbf{Z}$ can be considered as a delayed replica of $\mathbf{X}$ (by t_0 bits) that has been transmitted through a BSC of error probability P_e. Without loss of generality, we assume that the restitution delay t_0 is smaller than the size S of the interleaver.

In a non-cooperative context, the adversary has to know how many bits he missed and also the value of S. For this, he bets (n_a, d) on (S, t_0) and fills the *interception matrix* $H(n_a, d)$, of n_a columns, from the top-left corner to the bottom-right one using the intercepted bits $Z = Z_0 Z_1 \dots Z_{M \times S}$ and skipping the first $(n_a - d)$ ones. Without loss of generality, we consider that $H(n_a, d)$ has always M rows. If he found the right parameters, then the first bit of the interception matrix is the first bit of an interleaved block. Moreover, for noisy channels, the interception matrix can be written as $H(n_a, d) = \tilde{H}(n_a, d) + E(n_a, d)$, where $\tilde{H}(n_a, d)$ is the *noiseless interception matrix* that is built in the same way as $H(n_a, d)$ but with $X = X_0 X_1 \dots X_{M \times S}$ instead of $Z = Z_0 Z_1 \dots Z_{M \times S}$. If we denote E_i the error introduced by the channel, i.e. $Z_i = X_i + E_i \ \forall i$, then $E(n_a, d)$ represents the *error matrix* which is built in the same way as $H(n_a, d)$ but with $E = E_0 E_1 \dots E_{M \times S}$ instead of $Z = Z_0 Z_1 \dots Z_{M \times S}$. Moreover, its density is exactly P_e.

1.2. Error correcting code reconstruction problem

The problem the observer has to solve is the following: "Given the interception matrix $H(n_a, d)$, how to retrieve the parameters (S, t_0), G *and* Π?". The strategy we adopted consists in reconstructing the code $\mathscr{C}$ generated by G, then obtain (S, t_0) and finally, partially retrieve Π. Unfortunately, recovering the initial coder G and then decode is

equivalent to the problem of decoding a random code which is NP-complete [14]. The starting point of our technique is to reconstruct the dual code $\mathscr{C}^{\perp}$, which is equivalent to reconstruct $\mathscr{C}$. Indeed, using the definition of the dual code, for all codewords $y_i \in \mathscr{C}$ and for all $h \in \mathscr{C}^{\perp}$

$$\langle y_i, h\rangle = \langle h, y_i\rangle = 0, \tag{1}$$

where $\langle u, v\rangle$ denotes the scalar product of u and v, that implies

$$\tilde{H}(S, t_0)h = 0, \tag{2}$$

i.e. h belongs to the kernel of $\tilde{H}(S, t_0) : h \in \mathrm{Ker}(\tilde{H}(S, t_0))$. Moreover, if we assume that among the M codewords that compose the rows of $\tilde{H}(S, t_0)$, $N(n_c - k_c) = S(1 - r)$ are linearly independent, then $\mathscr{C}^{\perp} = \mathrm{Ker}(\tilde{H}(S, t_0))$. This assumption is reasonable as information words are usually generated by a compression and a ciphering process and sometimes by a scrambler.

In a non-cooperative context, we have no access to $\tilde{H}(S, t_0)$ because of the noise introduced by the channel and therefore, we are not able to directly compute $\mathscr{C}^{\perp}$. However, two important remarks give us the keys to retrieve $\mathscr{C}^{\perp}$. First of all, if we consider the code $\mathscr{C}'$, generated by the columns of $H(S, t_0)$, then Cluzeau [10] showed that it has a particular distribution of its vectors of small weights. It appears that $\mathscr{C}'$ has a huge amount of small weight codewords and that this amount decreases with the BER. In addition, Barbier [12] proved that a vector h, such that $H(S, t_0)h$ has a small Hamming weight, is in $\mathrm{Ker}(H(S, t_0))$ with a probability that tends towards 1 when M increases. Finally, reconstructing $\mathscr{C}^{\perp}$ and then $\mathscr{C}$ is an equivalent problem of finding small Hamming weight codewords in the code generated by the columns of the interception matrix $H(S, t_0)$. In this paper, we propose a technique adapted to this particular distribution of small Hamming weight codewords to estimate the right values for (n_a, d), then build $H(S, t_0)$ and finally reconstruct $\mathscr{C}^{\perp}$.

2. Identification of the size of the interleaver based on the rank criteria

2.1. Estimation in noise-free channels: Burel's method

In this section, we assume that no error is introduced by the channel (i.e. $P_e = 0$). In order to reconstruct the code $\mathscr{C}$, we search for a basis of $\mathscr{C}^{\perp}$. From (2), we need to compute $\mathrm{Ker}(H(S, t_0))$. For each vector $h = (h_1, \ldots, h_S)$, the xor of the columns of index i in $H(S, t_0)$, such that $h_i = 1$, is equal to 0. This property is illustrated in Fig. 1. A redundant bit is represented by the shaded box in this figure. This bit is a linear combination of other bits located in the same block. If $n_a = \alpha S$ and $d = t_0$, this linear relation is also satisfied for the next row and thus for the whole column. If $n_a \neq \alpha S$, we make the hypothesis that the interception matrix behaves like a random binary matrix. This hypothesis is practically observed. Moreover, if $n_a = \alpha S$ and $d \neq t_0$, then some bits of the linear combination may have moved to the next line, and so the xor of the considered columns is not equal to 0 anymore. In short: if $n_a \neq \alpha S$ then $\mathrm{rank}(H(n_a, d)) = n_a$ with a high probability, if $n_a = \alpha S$ and $d = t_0$ then $\mathrm{rank}(H(n_a, d))$ is minimal. To conclude, this gives us a criterion, also called the *rank criteria*, to estimate the right parameters. For instance, Burel et al. [11] examined the behavior of the ratio $\rho(n_a, d)$ defined by

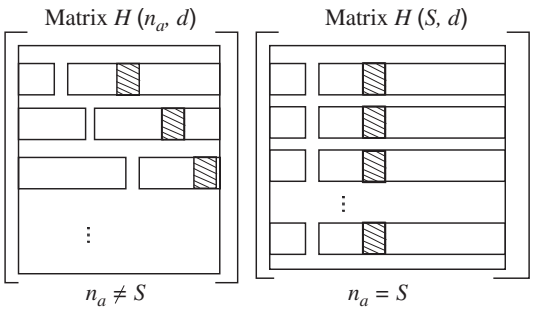

Fig. 1. Illustration of the rank deficiency of matrix $H(n_a, d)$ when $n_a = S$.

$$\rho(n_a, d) = \frac{\mathrm{rank}(H(n_a, d))}{n_a}, \tag{3}$$

for different values of n_a and d.

This rank deficiency property of $H(S, d)$ allows the authors to estimate S by doing an exhaustive search over the parameter n_a. Once the size of the interleaver is estimated, the minimum of $\rho(S, d)$ with respect to d allows them to estimate t_0. Practically, the rank criteria is applied to the upper $(n_a \times n_a)$-square matrix $H_1(n_a, d)$ extracted from $H(n_a, d)$.

As shown in [11], this approach is well adapted to estimate the parameters of the interleaver in an error free sequence. However, in a passive listening context, the intercepted sequence may be highly corrupted, which transforms the $H(S, t_0)$ matrix into a full-rank matrix and thus, the previous algorithm cannot be directly used. Indeed, we show in Appendix A.1 that the probability to have a rank deficient matrix if $n_a = S$ for a BSC of BER P_e is upper bounded by

$$\begin{aligned}\mathscr{P}^1_{\mathrm{det}} &= (n-k)\left(\sum_{i=0}^{\lfloor w_h/2\rfloor}\binom{w_h}{2i}P_e^{2i}(1-P_e)^{w_h-2i}\right)^S \\ &= (n-k)\left(\frac{1+(1-2P_e)^{w_h}}{2}\right)^S,\end{aligned} \tag{4}$$

where w_h is the smallest Hamming weight in $\mathscr{C}^{\perp}$. This probability tends toward zero when S increases. On the other hand, the probability to have a rank deficiency if $n_a \neq S$ is

$$\mathscr{P}^1_{fa} = 1 - \prod_{i=0}^{n_a-1}(1 - 2^{i-n_a}). \tag{5}$$

This probability is a straightforward application of the classical result [15] which gives the probability that a random matrix of size $k \times l$ is of full rank, i.e.

$$\prod_{i=0}^{l-1}(1 - 2^{i-k}).$$

If a sequence of size $M \times S$ is intercepted, we are able to construct $\binom{M}{S}$ matrices. The probability to have at least

one of those matrices with a deficient rank is

$$P_{\text{det}} = \binom{M}{S} \mathscr{P}_{\text{det}}^{1} \tag{6}$$

and for each tested size n_a, the false alarm probability is given by

$$P_{fa} = \binom{M}{n_a} \mathscr{P}_{fa}^{1}.$$

Eq. (6) gives us a lower bound of the detection performance of the interleaver parameters if we apply directly the method proposed by Burel et al. As soon as S and/or P_e are high, this method has no chance to estimate the interleaver parameters or need a huge intercepted sequence. In the next subsection, we propose an algorithm based on the same concept: a Gauss–Jordan elimination algorithm is adapted in order to identify "almost rank-deficient matrices". Indeed, if $n_a = S$, few erroneous bits due to propagation errors may destroy a rank-deficient matrix $H(n_a, d)$. Applying our proposed method allows us to accurately identify those cases.

2.2. Estimation in a noisy channel: our proposed method

The basic idea of our approach is to find "almost dependent columns" of $H(n_a, d)$. To achieve this, we adapt the well-known Gauss–Jordan elimination through pivoting (GJETP) algorithm [16]. We first briefly recall the GJETP for the binary field. This algorithm converts $H(n_a, d)$ into a lower triangular matrix noted $L(n_a, d)$ such that the number of its zeroed columns is exactly the dimension of its kernel. If this number is not zero, then $H(n_a, d)$ is rank-deficient. To describe the GJETP, we denote I_j the identity matrix of size j and N^i the ith column of a given matrix N.

Initialize $L(n_a, d)$ with $H(n_a, d)$, A_1 with I_M and A_2 with I_{n_a}. For $i = 1$ to $i = n_a$ do

(1) If the ith element of $L^i(n_a, d)$ is equal to zero, exchange $L^i(n_a, d)$ with the first $L^{i'}(n_a, d)$ $(i' > i)$ that has a one on its ith element. Exchange A_2^i and $A_2^{i'}$.
(2) If the ith element of $L^i(n_a, d)$ is equal to zero, exchange the ith row of $L(n_a, d)$ with its first row i' $(i' > i)$ that has a one on its ith element. Exchange the ith row of A_1 with its i'th.
(3) If the ith element of $L^i(n_a, d)$ is equal to one, xor $L^i(n_a, d)$ to any $L^{i'}(n_a, d)$ $(i' > i)$ that has a one on its ith row and xor A_2^i to $A_2^{i'}$.

End for output $L(n_a, d)$, A_1 and A_2.

It can be clearly seen that $L(n_a, d)$, A_1 and A_2 verify

$$A_1 H(n_a, d) A_2 = L(n_a, d). \tag{7}$$

If $L^i(n_a, d)$ is a zeroed column, then the vector A_2^i is in $\text{Ker}(H(n_a, d))$ and we call $L^i(n_a, d)$ a "dependent column". Without any modification, this algorithm is able to detect elements of $\text{Ker}(H(n_a, d))$ with a probability P_{det} and therefore has necessarily the same performance as the rank-based algorithm. As explained in the Introduction, if we find a vector h such that $H(S, t_0)h$ has a low Hamming weight, then $h \in \mathscr{C}^{\perp}$ with a high probability [12]. We consider now columns $L^i(n_a, d)$ such that their Hamming weights are small. As A_1 consists in row exchanges, $H(n_a, d)A_2^i$ has exactly the same Hamming weight as $L^i(n_a, d)$. We conclude that if such a column $L^i(n_a, d)$ exists, then A_2^i is in $\text{Ker}(\tilde{H}(n_a, d))$ with a high probability. This column is said to be an "almost dependent column". We call a "independent column", a column that is not almost independent. The notion of "almost rank-deficient matrix" is then naturally defined by a matrix with at least one "almost dependent column". We explain now, how to determine (S, t_0) using this notion. First of all, let B_k be the Hamming weight of $L^k(n_a, d)$. Note that the xor of independent columns gives an independent column. Thus, for an independent column, B_k is Binomial distributed and its mean is $m_B = M/2$. Let us define $\phi(k)$ by

$$\phi(k) = \frac{B_k}{m_B}. \tag{8}$$

For $n_a = \alpha S$, $\alpha \in \mathbb{N}$, the number of parity checks that are still detectable is less or equal to $2^{\alpha(n-k)}$, as explained in Section 2.1. For each parity check h_j, a set of column positions of $\tilde{H}(\alpha S, d)$: $\mathscr{I}_j^{(\alpha S,d)} = \{i_1^{(j)}, \ldots, i_{w_h}^{(j)}\}$ exists, such that

$$\tilde{H}(\alpha S, d) h_j = \tilde{H}^{i_1^{(j)}}(\alpha S, d) + \cdots + \tilde{H}^{i_{w_h}^{(j)}}(\alpha S, d) = 0. \tag{9}$$

Let us also define $\mathscr{D}_{\alpha S,d} = \{\mathscr{I}_1^{(\alpha S,d)}, \ldots, \mathscr{I}_{Q(\alpha S,d)}^{(\alpha S,d)}\}$ a basis of all sets $\mathscr{I}_j^{(\alpha S,d)}$, such that the set $\{h_1, \ldots, h_{Q(\alpha S,d)}\}$ is a basis of $\text{Ker}(\tilde{H}(\alpha S, d)$. Its cardinal $Q(\alpha S, d)$, which is less or equal to $\alpha(n-k)$, is non-zero and its maximum is reached when $d = t_0$. Assuming that there is no error on and over the main diagonal of $A_1 H(\alpha S, d)$, there is, for each element $\mathscr{I}_j^{(\alpha S,d)}$ of $\mathscr{D}_{\alpha S,d}$, one column of $L(\alpha S, d)$ at position $k_j \in \mathscr{I}_j^{(\alpha S,d)}$ such that B_{k_j} follows a Binomial law of parameters (M, P) and we have

$$\lim_{M\to\infty} \phi(k_j) \xrightarrow{\mathscr{P}} 2P, \tag{10}$$

with

$$\begin{aligned} P &= 1 - \sum_{l=0}^{\lfloor w_h/2 \rfloor} \binom{w_h}{2l} P_e^{2l} (1 - P_e)^{w_h - 2l} \\ &= \frac{1 - (1 - 2P_e)^{w_h}}{2}, \end{aligned} \tag{11}$$

where w_h is the cardinal of $\mathscr{I}_j^{(\alpha S,d)}$, i.e. the weight of the associated vector h_j. For the other columns $k \in \mathscr{I}_j^{(\alpha S,d)}, k \neq k_j$,

$$\lim_{M\to\infty} \phi(k) \xrightarrow{\mathscr{P}} 1.$$

Fig. 2 shows that even for a finite value of M, the gap between the two behaviors of $\phi_{kS}(.)$ is significant as long as w_h and P_e are not too large. This figure has been obtained for a $(7, 4)$ Hamming code, a pseudo-random interleaver of size 56, a BER $p_e = 0.08$ and an intercepted sequence of $10,000$ bits. Note that for this particular code, w_h is equal to 4, $\forall i$.

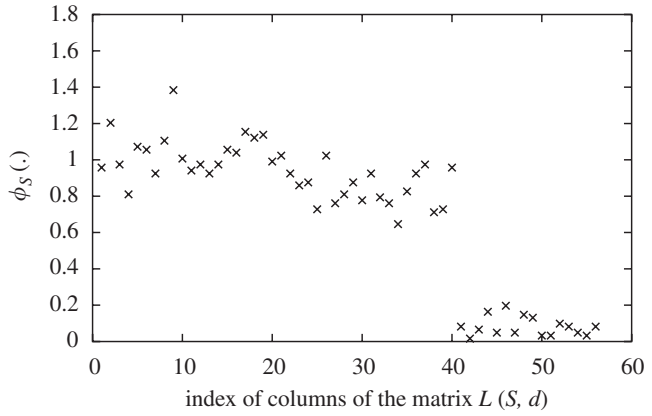

Fig. 2. Examples of value taken by $\phi_S(i)$: the gap between the independent columns and the dependent one is sufficient even for a finite size of the intercepted sequence.

The existence of this gap allows us to estimate $Q(\alpha S, d)$ by

$$\hat{Q}(\alpha S, d) = \mathrm{Card}(\{k \in \{1, \ldots, \alpha S\} / \phi(k) < \beta\}),$$

where Card$\{A\}$ is the cardinal of the set A and β a well-defined threshold (see Section 3.1 for its optimal expression). Moreover, using the distribution of small Hamming weight vectors pointed out by Cluzeau [10] and the analysis of Barbier [12], we justify that $\hat{Q}(\alpha S, d)$ is non-zero with a high probability and is maximal when $d = t_0$.

For $n_a \neq \alpha S$, the interception matrix behaves like a random binary matrix. Thus, the columns of $\tilde{H}(n_a, d)$ are all independent, with a probability of $(1 - \mathscr{P}_{fa}^1)$, then the cardinal $Q(n_a, d)$ of $\mathscr{D}_{n_a,d}$ is zero. This can be justified by easily showing that

$$\forall k \in \{1, \ldots, n_a\}, \quad \lim_{M \to \infty} \phi(k) \xrightarrow{\mathscr{P}} 1,$$

with $\xrightarrow{\mathscr{P}}$ meaning the convergence in probability.

Let us now discuss the effect of transmission errors on the GJETP. Transmission errors may have two different effects whether an erroneous bit "is used" by the GJETP algorithm to triangulate $H(n_a, d)$ (denoted by case (a)) or not (denoted later on by case (b)). Case (a) occurs when errors are located in the upper part of $H(n_a, d)$. An erroneous bit may lead the GJETP algorithm to add a column to other columns when it should not or not to add it when it should. This leads of course to a loss of dependent columns and those errors affect dramatically the efficiency of our algorithm. In case (b), rather than finding a zeroed column (that represents a linearly dependent column), we find a low Hamming weight column, with ones corresponding to the error positions. This case is of course not so problematic as case (a). One way to avoid effects of case (a) is accomplished by a randomized iterative procedure. Indeed the GJETP algorithm uses exclusively the upper part of $H(n_a, d)$ and the lower part is used to detect the dependent columns (deals with the case (b) errors). We propose for each iteration to choose a virtual new realization of $H(n_a, d)$ by randomly mixing the rows of $H(n_a, d)$. For each iteration, different almost dependent columns may be detected and the basis $\mathscr{D}_{n_a,d}$ may be completed. The row permutation of $H(n_a, d)$ can be seen as a virtual new realization of $H(n_a, d)$. The proposed algorithm is summarized below:

Blind Detection of Interleaver Parameters Algorithm.
Inputs: Z the intercepted sequence,
$S_{\min}$ the lowest interleaver length tested,
$S_{\max}$ the highest interleaver length tested,
$\beta \in [0, 1]$ the threshold,
nb the number of iterations.
Outputs: $\hat{\mathscr{C}}$ the estimation of $\mathscr{C}^\perp$ the dual code of $\mathscr{C}$ or $\emptyset$,
r the rate of the code or 0,
n the dimension of the code or 0.

```
Ĉ ← ∅
n ← 0
r ← 1
For n_a from S_min to S_max
    For d from 0 to n_a − 1
        H ← ∅
        fill H(n_a, d) using Z according to Section 1.1
        For i from 1 to nb
            fill H_(i)(n_a, d) by randomly mixing the rows of H(n_a, d)
            (A_1, A_2, L_(i)(n_a, d)) ← GJETP(H_(i)(n_a, d))
            For j from 1 to n_a
                If w_H(L^j_(i)(n_a, d))/n_a ⩽ β then
                    H ← H ∪ {A^j_2}
                End if
            End for
        End for
        H ← Span(H)
        If r > 1 − Dim(H)/n_a then
            r ← 1 − Dim(H)/n_a
            n ← n_a
            Ĉ ← H
        End if
    End for
End for
Return (Ĉ, r, n)
```

3. Theoretical analysis of the proposed method

3.1. Expression of the optimal detection threshold β

The choice of the threshold is important for our algorithm: if the threshold is too high, we may miss the detection of correlated rows. On the other hand, if it is too small we may consider independent columns as dependent ones (false alarm). Therefore, we take the optimal threshold as the one that minimizes the probability of miss-detection P_{md} of a theoretically dependent column (i.e. a column in the set $\mathscr{I}$). P_{md} corresponds to the sum of two probabilities: the probability to have $\phi_S(i) > \beta$ and the column i is a dependent column, with the probability to have $\phi_S(i) < \beta$ and the column i is an independent column (see shadowed part of Fig. 3). Fig. 3 presents an example of the probability density function[1] of $\phi_S(i)$, the left one corresponds to a dependent column and the other one to an independent column. We have also represented a possible threshold β and the associated probability of miss-detection P_{md}.

Knowing the probability law of ϕ, we are able to compute the optimal threshold. For this purpose, we have

[1] Those densities are discrete. Nevertheless, for seek of clarity, they are represented as continuous ones.

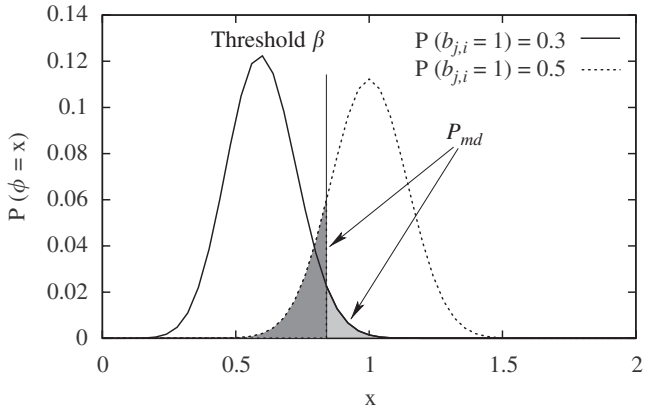

Fig. 3. Definition of the probability of miss-detection P_{md} of a theoretically dependent column.

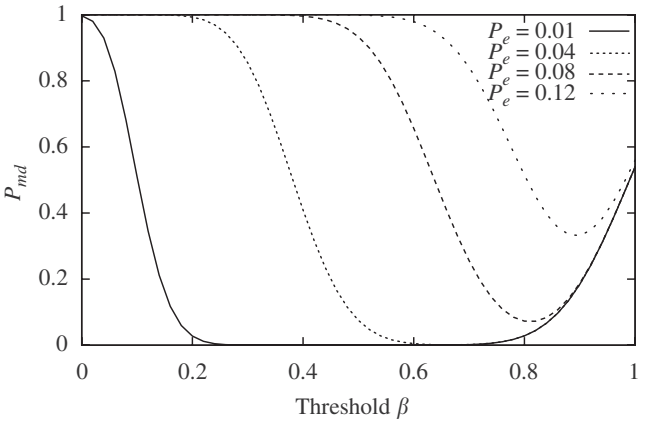

Fig. 4. P_{md} versus the threshold β for different value of P_{e}.

to solve this optimization problem:

$$\beta^* = arg\min_{\beta}(P_{\mathrm{md}}(m_B, P, \beta)), \tag{12}$$

with

$$P_{\mathrm{md}}(m_B, P, \beta) = \sum_{i=0}^{\lfloor m_B.\beta \rfloor} \binom{i}{2m_B} (0.5)^{2m_B} + \sum_{i=\lfloor m_B.\beta \rfloor + 1}^{2m_B} \binom{i}{2m_B} P^i (1-P)^{2m_B - i}.$$

Unfortunately, it is not possible to obtain an analytic expression for β^*. However, in order to study the influence of P, m_B and β on P_{md}, we compute $P_{\mathrm{md}}(m_B, P, \beta)$ over a fine fixed grid of β by computer simulation.

First of all, let us study the influence of β over P_{md} for different values of P_{e}. In Fig. 4, we plotted P_{md} versus β where m_B is fixed to 50 and P is computed for $w_{\mathrm{h}} = 6$ and for different P_{e} (see (11)). This figure shows that for channels having a low BER, the threshold has low influence on the probability of good detection. For $P_{\mathrm{e}} = 0.01$, if the threshold $\beta \in [0.2, 0.8]$, the value of P_{md} is close to zero.

Fig. 5 illustrates the dependence of the optimal threshold β^* on the variable w_{h}. We notice in this figure that the bigger w_{h} is, the bigger β^* is. This means that the bigger w_{h} is, the more difficult the detection is.

At last, let us study the influence of m_B on the value β^*. When m_B increases, the variance of both Binomial random variables decreases but the position of the optimal threshold stays almost constant. Therefore, we notice that m_B does not have much influence on the value β^* but impacts only the value of P_{md}. This is shown in Fig. 6, where we plotted $P_{\mathrm{md}}(m_B, P, \beta^*)$ versus P_{e}, for different values of m_B. This figure shows that P_{md} decreases significantly with m_B: for $P_{\mathrm{e}} = 0.1$, P_{md} is equal to 0.35 when $m_B = 25$, and $P_{\mathrm{md}} = 0.06$ when $m_B = 100$.

In order to obtain an analytical expression of the threshold, we approximate the Binomial laws of parameters (M, P) (see Eq. (11) for the expression of P) by a

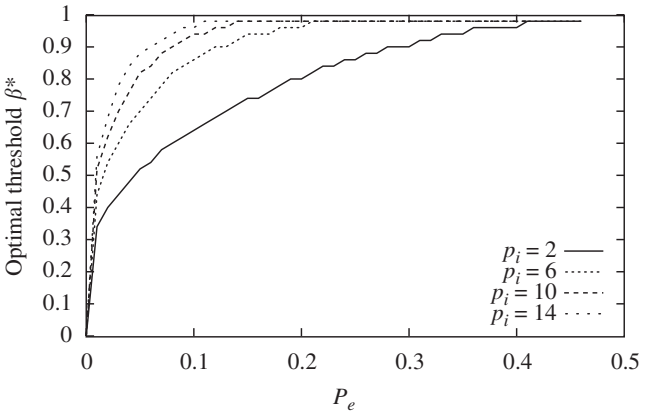

Fig. 5. Optimal threshold β^* versus w_{h}.

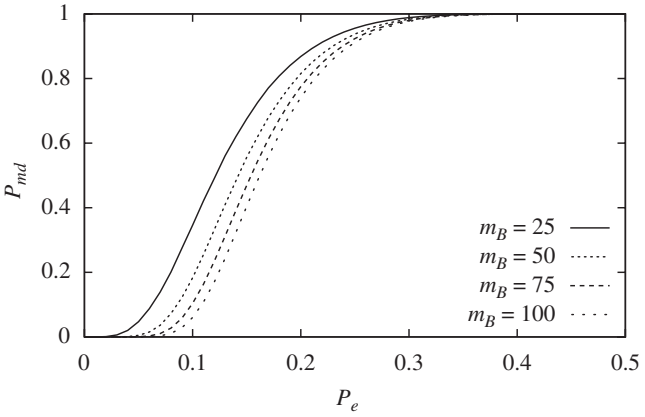

Fig. 6. P_{md} versus P_{e} at $\beta = \beta^*$.

Normal law [17]. With this approximation, we find the following expression for the optimal threshold:

$$\beta^* = \frac{-b - \sqrt{b^2 - ac}}{a},$$

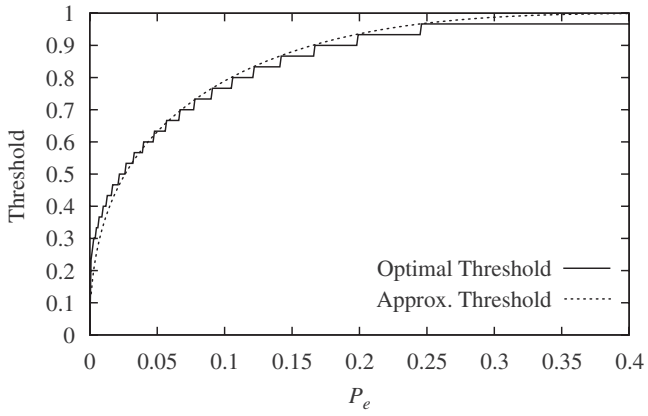

Fig. 7. Optimal and approximated threshold for $m_B = 30$ and $w_h = 4$.

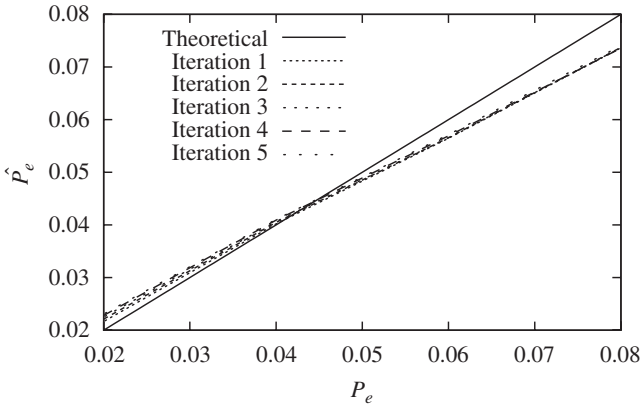

Fig. 8. Estimation of P_e to set the optimal threshold.

where

$$a = -m_B(1-2P)^2, \tag{14}$$
$$b = -2m_B P(1-2P), \tag{15}$$
$$c = 4m_B P(1-2P) - 2P(1-P)\ln(4P(1-P)). \tag{16}$$

The proof of the above equations is given in Appendix A.2. Note again that m_b does not influence much the value of β^* in Eq. (13). Indeed, as $2P(1-P)\ln(4P(1-P)) \in [\sqrt{1-e^{-1}}(\sqrt{1-e^{-1}}-1), 0]$, $\forall P \in [0; \frac{1}{2}]$, the second term in (16) can be neglected when m_B is large enough. In this case, the threshold β^* becomes independent of m_B. In Fig. 7, the optimal threshold and the approximated threshold given by (13) are presented. Simulation parameters are $m_B = 30$ and $w_h = 4$. The last value is used to calculate P according to (11).

Fig. 7 shows that the Gaussian approximation to compute the threshold is accurate.

The optimal threshold β^* depends on w_h, P_e and m_B. Whereas m_B is known by the observer, w_h and P_e are not. Thus, we are *a priori* not able to set the optimal threshold at the beginning of the algorithm. However, we present in the next subsection a way to iteratively adjust the threshold in order to improve the detection.

3.2. Practical choice of the threshold β

In order to set up the value of the threshold, it is possible to take advantage of the iterative nature of the algorithm. At the end of an iteration, we may have a value of n_a for which we detect some dependent columns. This value of n_a corresponds to the estimated size of the interleaver. Let us denote by $\mathscr{K}$ the set of dependent columns found. Using the matrix A_2 (see (7)), we can find the value w_h for those columns (w_h represents the number of columns involved to obtain column i in $L(S,d)$). Using the value of $\phi(i)$ for $i \in \mathscr{K}$, it is possible to estimate P by

$$\hat{P} = \frac{1}{\mathrm{Card}(\mathscr{K})}\sum_{i\in\mathscr{K}}\frac{\phi(i)}{2}. \tag{17}$$

Using (11), an estimation of P_e is easily obtained. Finally, using (13), an estimation of the optimal threshold is found. The next iteration is run with this new value of the threshold and the probability to detect additional dependent columns increases. By iterating this procedure, the estimation of P and thus the estimation of the optimal threshold is improved. Fig. 8 presents the estimated value of P_e versus P_e. After the first iteration, an accurate estimation of P_e is obtained.

A self-adapting optimal threshold is a powerful feature of our algorithm and it allows to significantly improve the performance. In the next section, we propose another feature if soft data (reliability information on the intercepted bits) are available.

3.3. GJETP algorithm and soft information

Let us denote by $H_{(i)}(n_a, d)$ the matrix $H(n_a, d)$ used at iteration i. If we do not have any soft information of the intercepted bits, the only possibility we have to obtain a new matrix $H_{(i)}(n_a, d)$ is to simply randomly permute the rows of $H(n_a, d)$. However, if we have any information about the reliability of the bits (for example the soft value of the bits before hard decision), we may choose matrix $H_{(i)}(n_a, d)$ that has reliable values in its upper part. In the following, we assume that we have such an information. Instead of a BSC, we now consider an AWGN channel with the following "bit to symbol mapping": $a_k = 2\alpha_k - 1$, where a_k is the symbol corresponding to the bit α_k. The absolute value of a received symbol is called its *reliability*. In an AWGN channel, the probability that a symbol leads to an erroneous decision is $P_{be} = \frac{1}{2}\mathrm{erfc}(\sqrt{E_b/N_0})$ where $\mathrm{erfc}(x)$ is the complementary error function defined by

$$\mathrm{erfc}(x) = \frac{2}{\pi}\int_x^\infty e^{-u^2}\,du.$$

Realizations of the Gaussian noise being independent, the probability to have k errors per row is equal to

$$P_{bel}(k) = \binom{n_a}{k} P_{be}^k (1-P_{be})^{n_a-k}. \tag{18}$$

An estimation of the number of rows in the matrix $H(n_a, d)$ with k errors is given by $M \times P_{bel}(k)$. Ideally, we would like to order rows of $H(n_a, d)$ according to the number of errors per row. In order to be as close as possible to this ordering, we use the function $F_\alpha(j)$ which gives an estimation of the

number of reliable bits in each row:

$$F_\alpha(j) = \mathrm{card}(\{m; |h_{j,m}| > \alpha\}_{m=1,\ldots,n_a}), \quad (19)$$

where $h_{j,m}$ is the element of $H(n_a, d)$ at position (j, m).

Once the ordering is achieved, our detection algorithm is performed using the matrix obtained after the hard decision taken from $H_{(i)}(n_a, d)$. Sorting the rows of $H(n_a, d)$ clearly improves the performance of the first iterations of the algorithm. However, when the number of iterations grows, the difference of performance between the two approaches (with and without ordering) decreases. To summarize, the main point of this ordering is to construct the upper part of the matrix $H(n_a, d)$ with the more reliable blocks. Nevertheless, as the data sequence is intercepted before being deinterleaved, it may be corrupted by burst of errors. In other words, the stream is composed of long error free sequences followed by burst of errors. This particularity makes the proposed algorithm well adapted to practical transmission channels.

4. Identification of the interleaver function

In order to estimate the size of the interleaver, we just need to detect at least one parity check in the matrix $H(S, d)$. However, to get a precise idea of the interleaver function used (or an estimation of the rate of the code), the larger the number of detected parity checks the more reliable the estimation. Therefore, in the next section, we express the probability of detection of a parity check and then present our procedure to estimate the position of codewords in the interleaved block.

4.1. Probability of detection of a parity check

Let us consider a parity check h_j defined by the set of column indexes $\mathscr{I}_j^{(S,d)} = \{i_1^{(j)}, \ldots, i_{w_h}^{(j)}\}$ as defined in Section 2.2. We introduce the parameter $n_j^{(S,d)} = \max_i \{i \in \mathscr{I}_j^{(S,d)}\}$. In order to retrieve h_j, the following conditions should be satisfied:

- The GJETP algorithm "performs well" for h_j (i.e. it adds the columns belonging to the set $\mathscr{I}_j^{(S,d)}$ together and so, h_j is a columns of A_2). The probability of such a case is lower bounded by $\mathscr{P}_{\mathrm{det}}^1(j)$.
- $\phi(j) < \beta$. This case occurs with probability $\mathscr{P}_{\mathrm{det}}^2(j)$.

First, the GJETP algorithm is performed on $H(S, d)$. As this algorithm only depends on the upper square matrix, for obvious complexity reasons, we restrict the GJETP algorithm to $H_1(S, d)$ where

$$H(S,d) = \begin{bmatrix} H_1(S,d) \\ \cdots\cdots \\ H_2(S,d) \end{bmatrix} \text{ and } H_1(S,d) \text{ is an } S \times S \text{ matrix.}$$

Two cases may lead h_j to be a column of A_2. First of all, for each row in the first $(n_j^{(S,d)} - 1)$ rows of $E(S, d)$, the number of errors that appear at indexes $\mathscr{I}_j^{(S,d)}$ is even. This case occurs with probability $\mathscr{P}_{\mathrm{det}}^1(j)$ where

$$\mathscr{P}_{\mathrm{det}}^1(j) = \left(\sum_{i=0}^{\lfloor w_h/2 \rfloor} P_e^{2i} (1 - P_e)^{w_h - 2i} \right)^{n_j^{(S,d)} - 1} = \left(\frac{1 + (1 - 2P_e)^{w_h}}{2} \right)^{n_j^{(S,d)} - 1}. \quad (20)$$

On the other hand, the fact that h_j is in A_2 despite the first case is not verified. For instance, if an erroneous bit stays below the diagonal during all the pivoting, it does not affect the result. Practically, this case occurs frequently and unfortunately, evaluating its probability is still an open problem. Nevertheless, $\mathscr{P}_{\mathrm{det}}^1(j)$ is a lower bound for the probability that h_j is in A_2. Moreover, compared to the algorithm based on the rank criteria, a much higher probability of detection is obtained here since $n_j^{(S,d)}$ may be much smaller than S.

After observing h_j as a column of A_2, we aim to decide between the two hypothesis:

$$\mathscr{H}_0 : h_j \in \mathrm{Ker}(\tilde{H}(S,d)), \quad (21)$$

$$\mathscr{H}_1 : h_j \notin \mathrm{Ker}(\tilde{H}(S,d)). \quad (22)$$

The induced decision rule $\mathscr{R}_\beta$ is the following one. One decides $\mathscr{H}_0$ if $\phi(j) \leqslant \beta$ and $\mathscr{H}_1$ otherwise. The probability $\mathscr{P}_{\mathrm{det}}^2(j)$ can be easily obtained and is given by

$$\mathscr{P}_{\mathrm{det}}^2(j) = P(B_k \leqslant 2m_B\beta | \mathscr{H}_1).$$

Under the assumption that correlations in codewords are independent (i.e. $\{i_1^{(j)}, \ldots, i_{w_h}^{(j)}\}$ and $\{i_1^{(k)}, \ldots, i_{w_h}^{(k)}\}$ are independent), the probability of detection of the size of the interleaver is

$$\mathscr{P}_{\mathrm{det}} = \sum_{\text{all } j} \mathscr{P}_{\mathrm{det}}^1(j) \mathscr{P}_{\mathrm{det}}^2(j). \quad (23)$$

For real codes, the correlations are surely not independent. Therefore, we estimate $\mathscr{P}_{\mathrm{det}}$ by

$$\hat{\mathscr{P}}_{\mathrm{det}} = \min\left(1, \sum_j \mathscr{P}_{\mathrm{det}}^1(j) \mathscr{P}_{\mathrm{det}}^2(j)\right). \quad (24)$$

The probability of false alarm to detect the parity check h_j is

$$\mathscr{P}_{\mathrm{fa}}(j) = P(B_k \leqslant m_B\beta | \mathscr{H}_2) \quad (25)$$

$$= 2^{-2m_B} \sum_{i=0}^{\lfloor m_B\beta \rfloor} \binom{2m_B}{i}. \quad (26)$$

Compared to a straightforward rank approach, we notice that our false alarm probability goes to zero when the size of the intercepted sequence grows (i.e. m_B). The proof of this assertion can be found in Appendix A.3.

4.2. Position of codewords within the interleaved sequence

We assume in this section that we have correctly estimated the size and the start of the interleaved frame. In order to have an idea of the structure of the interleaver (i.e. the interleaver function used), we need to locate the bits belonging to the same codeword in the interleaved block. In other words, we need to estimate $\mathscr{D}_{S,t_0}$, the basis

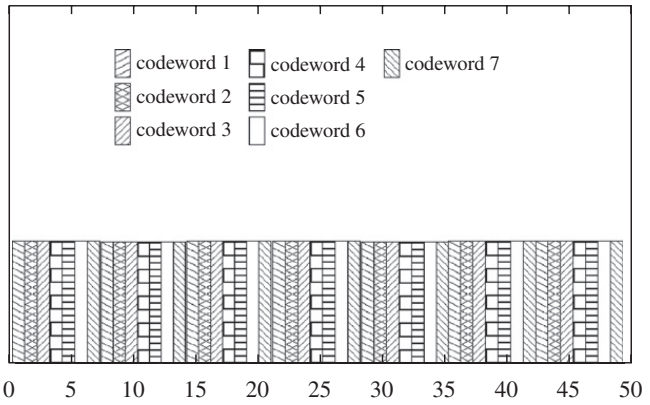

Fig. 9. Estimation of the position of codewords in the interleaved block.

of sets of indexes $\{\mathscr{I}_{j=1,\ldots,Q(S,t_0)}^{(S,t_0)}\}$ representing the parity checks within the interleaved block. Once this basis is estimated, we are able to find the position of the bits belonging to the same codeword. Indeed, the sets $\mathscr{I}_j(S,t_0)$ necessarily indicate positions of bits of the same codeword. If $p \in \mathscr{I}_j(S,t_0)$ and $p \in \mathscr{I}_i(S,t_0)$, then bits at positions $\mathscr{I}_j(S,t_0)$ and $\mathscr{I}_i(S,t_0)$ belong to the same codeword. For each column k of matrix $L(S,t_0)$ satisfying $\phi(k)<\beta$, $\mathscr{I}_k^{(S,t_0)}$ is obtained using A_2^k. These columns are identified by

$$H^{i_1^{(k)}}(S,t_0)+\cdots+H^{i_{w_h}^{(k)}}(S,t_0)=(A^{-1}L(S,t_0))^k.$$

In other words, $\hat{\mathscr{I}}_j^{(S,t_0)}=\{i_1^{(k)},\ldots,i_{w_h}^{(k)}\}$ is an estimator of one element of $\mathscr{D}_{S,t_0}$.

Note that if the synchronization was not correctly achieved, we would not be able to find the codeword to which the first or last bit of the block belongs. This algorithm may also be used to perform the blind frame synchronization. Remark also that the efficiency of our interleaver reconstruction procedure depends directly on the number of dependent columns found in $L(n_a,d)$. Therefore, the choice of the optimal threshold is not so crucial for the detection of the size of the interleaver (because the detection of only one parity check is sufficient to estimate the interleaver size) but becomes a real necessity in order to estimate the interleaver function.

Let us illustrate this algorithm with the following example. We intercepted a binary stream coming from a (7,4) Hamming encoder followed by an interleaver of size 49 bits (an interleaved block contains seven codewords). In this example, we consider that the size of the intercepted sequence is 10,000 bits. Five iterations are run and the probability of an error is set to 0.05. We are able to estimate all parity checks in 99.8% of the cases and Fig. 9 shows the position of codewords in the interleaved frame. In that particular case, the interleaver can be clearly identified: it is a row/column type interleaver.

5. Blind frame synchronization: identification of t_0

Our algorithm performance is directly linked to the probability $\mathscr{P}_{\text{det}}^1$ that the GJETP algorithm performs well. If we take a close look at the expression of $\mathscr{P}_{\text{det}}^1$ in (20), we

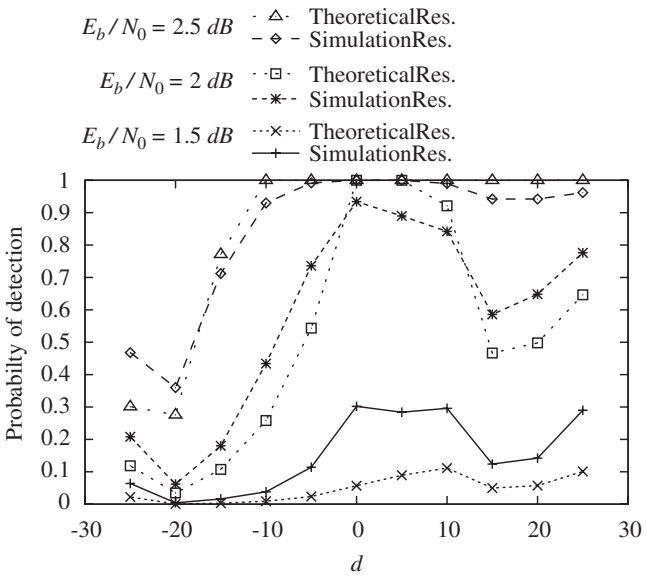

Fig. 10. Comparison between results obtained by simulations and with the theoretical expression in (20) for different delay d, and different E_b/N_0 with a row–column interleaver.

notice that when we are not yet synchronized (i.e. when $d\neq t_0$), the parameter $n_j^{(S,d)}$ has a major impact on this probability. Thus, two different cases should be considered: $t_0-S/2\leqslant d<t_0$ and $t_0\leqslant d\leqslant t_0+S/2$.

5.1. Case 1: $t_0-S/2\leqslant d<t_0$

In this case, the first (t_0-d) columns of $H(S,d)$ correspond to the last bits of the interleaved blocks. The probability to get a complete parity check in these first (t_0-d) columns decreases when d becomes closer to t_0 and the probability to get complete parity checks in the $(S-(t_0-S))$ remaining columns increases. However, the probability to detect those parity checks is penalized since we have necessarily $n_j^{(S,d)}=n_j^{(S,t_0)}+(t_0-d)$.

5.2. Case 2: $0\leqslant d-t_0\leqslant S/2$

In this case, we do not have the drawback previously explained since the present parity checks have lower values of $n_j^{(S,d)}$ than in the previous case. This gives a higher probability of correct detection. However, as $d\neq t_0$ we still have the possibility to loose parity checks (the ones involving the first $(d-t_0)$ bits of the interleaving blocks. Therefore, the probability of detection of our algorithm for a given offset d is not symmetric around $d=t_0$. This result is verified in Fig. 10 of the next section. In order to avoid this drawback, we perform our algorithm twice, once on the matrix $H(S,d)$ built as explained in Section 1.1 and a second time on a matrix $\bar{H}(S,d)$ constructed by a symmetric permutation of the columns of $H(S,d)$. The ith column of $\bar{H}(S,d)$, $\bar{H}^i(S,d)$ is exactly $H^{S-i}(S,d)$.

Simulations allows us to verify this behavior.

6. Simulation results

In this section, we illustrate the theoretical expressions we pointed out and enlighten the efficiency of our algorithm. In all simulations, a $(7,4)$ Hamming block code is used and the interleaver has a length of 56 bits. Without loss of generality, we assume that $t_0 = 0$ and that the number of intercepted bits is set to 50,000 bits. Moreover, we use the optimal threshold found in Section 4.1.

6.1. Detection of a parity check

In this first simulation, we verify the probability of correct parity check detection given by (20). The interleaver is a row–column interleaver. As we use a $(7,4)$ Hamming code, each block of the interleaver contains eight codewords. Fig. 10 presents the probability of correct detection of at least one parity check versus d. Notice that when $d = 0$ we are synchronized with the interleaver and we get the best probability of detection. Five hundred Monte-Carlo trials have been run where the noise and information bits were randomly chosen at each trial.

The difference observed between the theoretical and estimated performance in Fig. 10 can be explained by the fact that the correlations introduced by the code are not independent. Nevertheless, the curves behavior is similar and the bound we pointed out appears to be accurate. As explained in Section 5, we observe that the probability is not symmetric with respect to the delay d.

In the remaining of this paper we use the improved algorithm where the detection is performed twice, on $H(n_a, d)$ and on its symmetric version $\bar{H}(n_a, d)$ as explained in Section 5.

6.2. Detection of the interleaver and parity checks found

Let us now illustrate the improvement obtained with our proposed iterative procedure (see Section 5). Fig. 11 presents the probability of correct detection of the interleaver size versus the signal-to-noise ratio (SNR) for different iterations. Five thousand Monte-Carlo trials are run and d is set to zero. This iterative procedure improves significantly the probability of detection. Indeed, for $E_b/N_0 = 1$ dB at iteration 1, we have a probability of detection equal to 0.29. After five iterations, we obtained a probability of detection equal to 0.78.

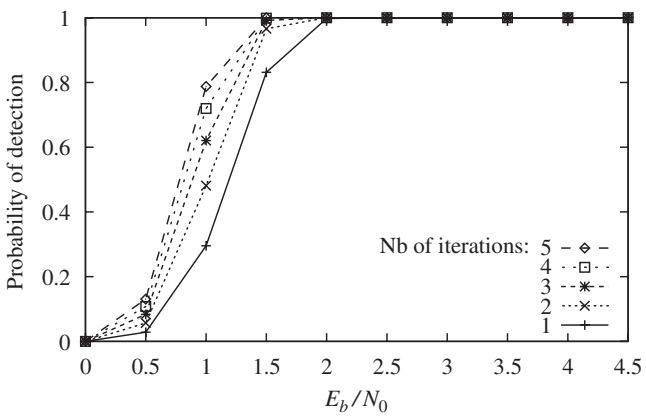

Fig. 11. Probability of detection for different numbers of iterations.

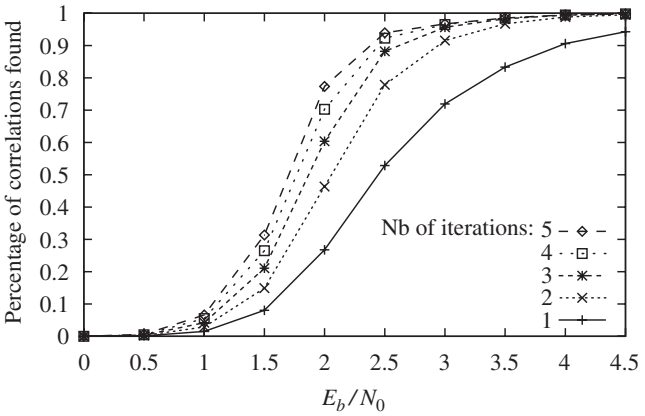

Fig. 12. Percentage of parity checks found for different numbers of iterations.

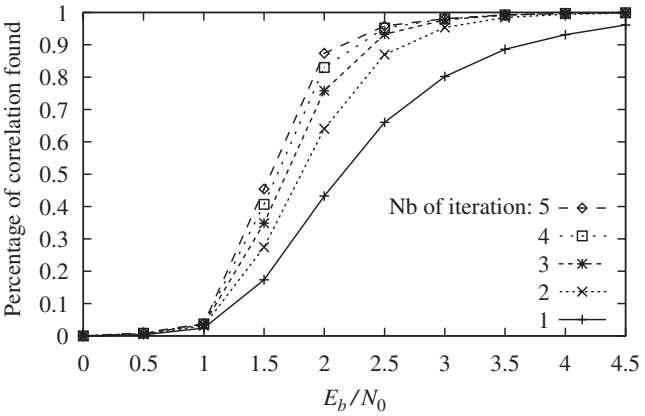

Fig. 13. Number of parity checks found with soft input data for different numbers of iterations.

Fig. 12 shows the percentage of the basis of parity checks found versus the SNR. A single parity check detection is enough to be able to identify the size of the interleaver. However, the more parity checks are identified, the more reliable the identification of the interleaver structure is. The iterative procedure improves significantly our capability to identify the interleaver. Indeed, the number of parity checks found increases with the number of iterations. Note that in our simulation, 24 parity checks are available in an interleaver block.

6.3. Performance improvement using soft data

In this simulation, we use the soft data to order the most reliable rows in the upper part of $H(n_a, d)$. The row ordering is achieved using the function given in (19) with

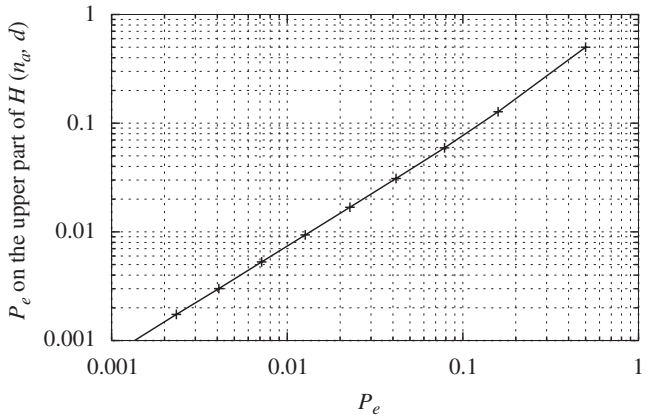

Fig. 14. Error probability on the upper part of $H(n_a, d)$ against P_e.

$\alpha = 1$. In order to illustrate the improvement obtained using rows ordering, we present in Fig. 13 the percentage of parity checks found versus the SNR and for different iterations. Those results should be compared to the ones of Fig. 12 obtained without ordering. The ordering allows us to get better performance with less iterations.

The row ordering has a direct impact on the probability $\mathscr{P}^1_{\text{det}}(j)$ (see (20)) that the GJETP algorithm performs well. The row ordering decreases the probability to have erroneous bits in the upper part of $H(n_a, d)$. This is illustrated in Fig. 14 where we plot the BER estimated in the upper part of the matrix $H(n_a, d)$ versus the error probability of the channel.

7. Conclusion

We have presented in this paper an algorithm based on linear algebra properties which, from a delayed and corrupted interleaved sequence of coded bits, allows us to blindly estimate the interleaver size, to synchronize the interleaver blocks and to estimate the dual of the code and its rate. We have also given a fine analysis of the probabilities of success and of false alarm. Moreover, we have proposed some improvements and detailed an algorithm to locate codewords inside an interleaved block, in order to obtain a more precise idea of the kind of the interleaver used. Unfortunately, retrieving the initial decoder and then decoding is equivalent to the problem of decoding a random code which is NP-complete [14]. The complexity of the proposed algorithm is clearly exponential in the BER but also in the size of the interleaver. One approach we have proposed is to use the soft information to decrease the BER in the upper square matrix of $H(n_a, d)$ to greatly improve the probability of detection of parity checks when applying the GJETP algorithm. Finally, we have shown some experimental comparisons between the theoretical bounds and our algorithm. This experiments illustrate that our theoretical analysis is correct and that our method works for high probability of errors of the BSC and that for instance, at a BER of 8%, we are able to correctly estimate the interleaver size in 76% of cases. Such results lead us to think that this technique may be efficiently applied in most of the communication channels using linear block codes. As the BER is a very limiting factor, we have investigated some techniques to artificially decrease it, at least inside the upper square matrix extracted from the intercepted matrix. In this way, we have improved the detection step of our algorithm.

Appendix A

A.1. Proof of Eq. (4)

Proof. For this proof we use notations introduced in Section 2. Let h be a parity check, i.e. $h \in \text{Ker}(\tilde{H}(n_a, d))$ then,

$$h \in \text{Ker}(H(n_a, d)) \text{ if and only if } \sum_{j=1}^{n_a} [E(n_a, d)]_{ij} h_j \equiv 0 \bmod 2 \quad \forall j = 1 \dots n_a.$$

This result is straightforward and does not cope with any difficulty. □

Let $h \in \text{Ker}(\tilde{H}(n_a, d))$ and $\mathscr{C}_{\text{h}} = \{\tilde{H}^i(n_a, d)$, the ith column of $\tilde{H}(n_a, d)$, such that $h_i = 1\}$. The previous result claims that $h \in \text{Ker}(\tilde{H}(n_a, d))$ if and only if the number of erroneous bits in each row in the columns of $\mathscr{C}_{\text{h}}$ is even. We can deduce the following theorem.

Theorem 1. *Let $h \in \text{Ker}(\tilde{H}(n_a, d))$, of Hamming weight w_{h}, then the probability that h is in $\text{Ker}(H(n_a, d))$ is*

$$P(h \in \text{Ker}(H(n_a, d)) | h \in \text{Ker}(\tilde{H}(n_a, d))) = \left(\frac{1 + (1 - 2P_{\text{e}})^{w_{\text{h}}}}{2} \right)^{n_a}.$$

Proof. Let us define $\mathscr{P} = P(h \in \text{Ker}(H(n_a, d)) | h \in \text{Ker}(\tilde{H}(n_a, d)))$, then

$$\mathscr{P} = \sum_{i=0}^{\lfloor w_{\text{h}}/2 \rfloor} \binom{w_{\text{h}}}{2i} P_{\text{e}}^{2i} (1 - P_{\text{e}})^{w_{\text{h}} - 2i}^{n_a}$$

and

$$\sum_{i=0}^{w_{\text{h}}} \binom{w_{\text{h}}}{i} (-1)^i P_{\text{e}}^i (1 - P_{\text{e}})^{w_{\text{h}} - i} = (1 - 2P_{\text{e}})^{w_n}, \tag{A.1}$$

$$\sum_{i=0}^{w_{\text{h}}} \binom{w_{\text{h}}}{i} P_{\text{e}}^i (1 - P_{\text{e}})^{w_{\text{h}} - i} = ((1 - P_{\text{e}}) + P_{\text{e}})^{w_{\text{h}}} = 1. \tag{A.2}$$

Using (A.1) and (A.2), we have

$$\sum_{i=0}^{\lfloor w_{\text{h}}/2 \rfloor} \binom{w_{\text{h}}}{2i} P_{\text{e}}^{2i} (1 - P_{\text{e}})^{w_{\text{h}} - 2i}^{n_a} = \left(\frac{1 + (1 - 2P_{\text{e}})^{w_{\text{h}}}}{2} \right)^{n_a}.$$

This last equality ends the proof of Theorem 1. This probability increases when w_{h} decreases. Moreover, the probability to observe a rank deficiency is the probability to detect at least one vector of $\mathscr{C}^{\perp}$ in the kernel of the interception matrix. If we consider a basis of $\mathscr{C}^{\perp}$ that contains the vector h of smallest Hamming weight w_{h}, then the probability to observe a rank deficiency is upper bounded by $(n - k)\mathscr{P}$, where $(n - k)$ is the number of independent vectors of $\mathscr{C}^{\perp}$. □

A.2. Proof of the approximation of the optimal threshold (Section 3.1)

Let X be a random variable following a Binomial law, i.e. $X \sim \mathscr{B}(N,p)$. According to [17] the Normal distribution $\mathscr{N}(Np, Np(1-p))$ is a good approximation of the Binomial law $\mathscr{B}(N,p)$. Using this approximation, we are able to compute easily the optimal threshold β^*. Using a $\tilde{\cdot}$ over the variable name indicates that we are using the Normal model to obtain it. For instance $\tilde{\beta}^*$ is the optimal threshold obtained with the Normal approximation.

The minimization problem (12) can then be rewritten as follows:

$$\tilde{\beta}^* = arg\min_{\beta}(\tilde{P}_{\mathrm{md}}(m_B,P,\beta)). \tag{A.3}$$

In this proof, we consider that $P \in]0,\frac{1}{2}[$, which is actually true in our context since $P_e \in]0,\frac{1}{2}[$ (see (11)). With the Normal approximation, we have

$$\begin{aligned}\tilde{P}_{\mathrm{md}}(m_B,P,\beta) = &\int_{-\infty}^{m_B\beta} \frac{1}{\sqrt{\pi m_B}}\exp\left(-\frac{(u-m_B)^2}{m_B}\right)\mathrm{d}u\\ &+\int_{m_B\beta}^{+\infty}\frac{1}{\sqrt{4\pi m_B P(1-P)}}\\ &\times\exp\left(-\frac{(v-2m_BP)^2}{4m_BP(1-P)}\right)\mathrm{d}v.\end{aligned}$$

In order to minimize $\tilde{P}_{\mathrm{md}}(m_B,P,\beta)$ with respect to β, we compute $\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)/\partial\beta$, and obtain

$$\begin{aligned}\frac{\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)}{\partial\beta} &= \frac{1}{\sqrt{\pi m_B}}\exp\left(-\frac{(m_B\beta-m_B)^2}{m_B}\right)\\ &\quad-\frac{1}{\sqrt{4\pi m_BP(1-P)}}\exp\left(-\frac{(m_B\beta-2m_BP)^2}{4m_BP(1-P)}\right)\\ &= \frac{1}{\sqrt{\pi m_B}}\exp(-m_B(\beta-1)^2)\\ &\quad-\frac{1}{\sqrt{4\pi m_BP(1-P)}}\exp\left(-\frac{m_B(\beta-2P)^2}{4P(1-P)}\right).\end{aligned}$$

Let us now find the value of β such that $\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)/\partial\beta = 0$.

$$\begin{aligned}\frac{\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)}{\partial\beta} = 0 &\Longleftrightarrow \frac{1}{\sqrt{\pi m_B}}\exp(-m_B(\beta-1)^2)\\ &\quad= \frac{1}{\sqrt{4\pi m_BP(1-P)}}\exp\left(-\frac{m_B(\beta-2P)^2}{4P(1-P)}\right)\\ &\Longleftrightarrow \sqrt{4P(1-P)}\\ &\quad= \exp\left(m_B(\beta-1)^2-\frac{m_B(\beta-2P)^2}{4P(1-P)}\right)\\ &\Longleftrightarrow 2P(1-P)\ln(4P(1-P))\\ &\quad= 4P(1-P)m_B(\beta-1)^2-m_B(\beta-2P)^2.\end{aligned}$$

To summarize, we obtain

$$\frac{\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)}{\partial\beta} = 0 \Longleftrightarrow a\beta^2+2b\beta+c=0,$$

with

$$\begin{aligned}a &= -m_B(1-2P)^2,\\ b &= -2m_BP(1-2P),\\ c &= 4m_BP(1-2P)-2P(1-P)\ln(4P(1-P)).\end{aligned}$$

We get a solution if $b^2-ac \geqslant 0$.

$$\begin{aligned}b^2-ac &= 4m_B^2P^2(1-2P)^2+4m_B^2P(1-2P)^3\\ &\quad-2m_BP(1-P)(1-2P)^2\ln(4P(1-P))\\ &= 4m_BP(1-2P)^2(m_BP+m_B(1-2P)\\ &\quad-(1-P)\ln(\sqrt{4P(1-P)}))\\ &= 4m_BP(1-2P)^2(m_B(1-P)-(1-P)\ln(\sqrt{4P(1-P)}))\\ &= 4m_BP(1-2P)^2(1-P)(m_B-\ln(\sqrt{4P(1-P)})).\end{aligned}$$

As $\forall x \in]0,\frac{1}{2}[$, $\sqrt{4P(1-P)} \in]0,1[$, we conclude that $b^2-ac \geqslant 0$, $\forall P \in]0,\frac{1}{2}[$. We obtain the following result:

$$\begin{aligned}&\frac{\partial\tilde{P}_{\mathrm{md}}(m_B,P,\beta)}{\partial\beta} = 0 \Longleftrightarrow \beta = \frac{-b-\sqrt{b^2-ac}}{a}\\ &\text{or } \beta = \frac{-b+\sqrt{b^2-ac}}{a}.\end{aligned}$$

As $b<0$, the second solution is negative because $a<0$. This solution is not acceptable for our problem. Therefore the only relevant solution for $\tilde{\beta}^*$ is

$$\tilde{\beta}^* = \frac{-b-\sqrt{b^2-ac}}{a}. \tag{A.4}$$

We can easily show that this solution is effectively a minimum. We just have to study the variations of the function $\tilde{P}_{\mathrm{md}}(m_B,P,\beta)$ around $\tilde{\beta}^*$.

A.3. Limit of $\mathscr{P}_{fa}$ when $m_b \to +\infty$ (Section 4.1)

This result is an application of Bienaymé–Tchebychev inequality. As B_k is Binomial distributed, $\mathscr{P}_{fa}(j) = P(B_k \leqslant m_B\beta|\mathscr{H}_2)$ and the mean of B_k under the assumption $\mathscr{H}_2$ is m_B and its variance is $m_B/2$. As the threshold β is chosen such that $\beta<1$, we have

$$P(B_k \leqslant m_B\beta|\mathscr{H}_2) \leqslant P(|B_k-m_B| \geqslant (1-\beta)m_B|\mathscr{H}_2).$$

With the Bienaymé–Tchebychev inequality, we obtain the following result:

$$\begin{aligned}P(|B_k-m_B| \geqslant (1-\beta)m_B|\mathscr{H}_2) &\leqslant \frac{var(B_k)}{(1-\beta)^2m_B^2},\\ &\leqslant \frac{1}{2(1-\beta)^2m_B}.\end{aligned}$$

Then we have $\lim_{m_B\to\infty} P(B_k \leqslant m_B\beta|\mathscr{H}_2) = 0$, which means that $\lim_{m_B\to\infty}\mathscr{P}_{fa}(j) = 0$.

References

[1] J. Proakis, Digital Communications, McGraw-Hill, New York, 1968.

[2] G. Sicot, S. Houcke, Blind detection of interleaver parameters, in: Proceedings of the ICASSP 2005, Philadelphia, USA, 2005.

[3] G. Sicot, S. Houcke, Theoretical study of the performance of a blind interleaver estimator, in: Proceedings of the ISIVC 2006, Hammamet, Tunisia, 2006.

[4] J. Barbier, G. Sicot, S. Houcke, Algebraic approach for the reconstruction of linear and convolutional error correcting codes, in: C. Ardil (Ed.), Proceedings of the 3rd International Conference on Computer Science and Engineering CISE 2006, vol. 16, World Enformatika Society, Venice, Italy, 2006, pp. 66–71, ISBN: 975-00803-6-X.
[5] M. Cluzeau, Block code reconstruction using iterative decoding techniques, in: Proceedings of the 2006 IEEE International Symposium on Information Theory, ISIT06, Seattle, USA, 2006.
[6] A. Canteaut, F. Chabaud, A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511, IEEE Trans. Inform. Theory IT-44 (1) (1998) 367–378.
[7] J. Stern, A method for finding codewords of small weight, in: G. Cohen, J. Wolfmann (Eds.), Proceedings of the Coding Theory and Applications, Lecture Notes in Computer Science, vol. 388, Springer, Toulon, France, 1989, pp. 106–113.
[8] J. Leon, A probabilistic algorithm for computing the minimum weight of large error-correcting codes, IEEE Trans. Inform. Theory IT-34 (5) (1988) 1354–1359.
[9] G. Planquette, Identification de trains binaires codés, Ph.D. Thesis, Université de Rennes I, France, December 1996.

É

[12] J. Barbier, Analyse de canaux de communication dans un contexte non coopératif. application à la stéganographie et aux codes correcteurs d'erreurs, Ph.D. Thesis, École Polytechnique, Palaiseau, France, November 2007.
[13] A. Valembois, Detection and recognition of a binary linear code, Discrete Appl. Math. 111 (2001) 199–218.
[14] E. Berlekamp, R. McEliece, H. van Tilborg, On the inherent intractability of certain coding problems, IEEE Trans. Inform. Theory IT-24(3) (1978).
[15] R. Lidl, H. Niederreiter, Finite Fields, Cambridge University Press, Cambridge, 1983.
[16] G. Golub, C.V. Loan, Matrix Computations, The Johns Hopkins University Press, Baltimore, MD, 1989.
[17] D.B. Peizer, J.W. Pratt, A normal approximation for Binomial, F, beta, and other common, related tail probabilities, J. Amer. Statist. Assoc. 63 (324) (1968) 1457–1483.