Computing the minimum distance of linear codes by the error impulse method.

Claude Berrou, Sandrine Vaton, Michel Jezequel, Catherine Douillard

Computing the Minimum Distance of Linear Codes by the Error Impulse Method

Claude Berrou, Sandrine Vaton, Michel Jézéquel and Catherine Douillard
Ecole Nationale Supérieure des Télécommunications
BP 832, 29285 Brest Cédex, France

Abstract - A new method for computing the minimum distances of linear error-correcting codes is proposed and justified. Unlike classical techniques that rely on exhaustive or partial enumeration of codewords, this new method is based on the ability of the Soft-In decoder to overcome Error Impulse input patterns. It is shown that the maximum magnitude of the Error Impulse that can be corrected by the decoder is directly related to the minimum distance. This leads to a very fast algorithm to obtain minimum distances of any linear code whatever the block size and the code rate considered. In particular, the method can be advantageously worked out for turbo-like concatenated codes.

Keywords: minimum distance, error floor, linear code, convolutional, concatenated, turbo code.

1 Introduction

The asymptotic performance of a linear block error correcting code at very low error rates is completely determined by its minimum distance $d_{\min}$ and by the multiplicity $n(d_{\min})$ if one considers the Frame Error Ratio (FER).

Consider a $C(n,k)$ linear block code (k is the number of information bits, n is the size of the codewords). Denote by $d_{\min}$ its minimum distance; for any distance $d \geq d_{\min}$, denote by $n(d)$ its multiplicity (number of codewords with weight d). On the Gaussian channel with maximum-likelihood (ML) decoding the Frame Error Ratio (FER) can be upperbounded by the union bound:

$$\text{FER} \leq \frac{1}{2} \sum_{d \geq d_{\min}} n(d) \text{erfc}((dR\frac{Eb}{N0})^{1/2}) \qquad (1)$$

where $\text{erfc}(x)$ is the complementary error function: $\text{erfc}(x) = 2/\sqrt{\pi} \int_x^\infty \exp(-t^2)\, dt$.

At low error rates, this upper bound is very tight; moreover, as the coefficients $\text{erfc}(x^{1/2})$ decrease exponentially with x, the FER can be approximated, at low error rates, by the first term of the union bound:

$$\text{FER} \simeq \frac{1}{2} n(d_{\min}) \text{erfc}((d_{\min} R\frac{Eb}{N0})^{1/2}) \qquad (2)$$

When designing a code that operates at very low error rates, it is very difficult to determine its performance by Monte Carlo simulations. It is highly desirable to propose some techniques to determine the minimum distance $d_{\min}$ and its multiplicity $n(d_{\min})$.

For algebraic codes such as BCH codes or Reed Solomon codes, $d_{\min}$ is a parameter that is specified prior to the design of the code. For elementary convolutional codes, the Viterbi algorithm [4] can be practically used to estimate the minimum distance, but this algorithm cannot be used to estimate the minimum distance of concatenated codes and in particular of Turbo Codes (TCs).

It is then important to develop some fast algorithms to compute the minimum distance $d_{\min}$ of concatenated codes and, in particular, of TCs. This will make it possible to design efficient TCs without intensive computations. This will for example make it possible to design good component codes and good interleavers in record time.

Some authors have suggested a simplified approach to determine the minimum distance $d_{\min}$ of TCs: looking for the minimum weight $d_{\min}^{(2)}$ of the codewords generated by weight 2 information words [8]. With statistical interleaving, when the interleaver size tends to infinity, the codewords of weight $d_{\min}$ are generated by information words of weight 2 with a high probability. But for non statistical interleavers, this method only provides an upper bound on the minimum distance: $d_{\min} \leq d_{\min}^{(2)}$, and in general this bound is not tight.

Another method has been proposed recently to compute the minimum distance $d_{\min}$, its information bit multiplicity $w(d_{\min})$ and its multiplicity $n(d_{\min})$ for parallel and serially concatenated convolutional codes [7]. This method does not make any assumption about the weight of the input sequence. It is based on the notion of constrained subcode i.e. a subset of a code defined via constraints on the edges of its trellis. It consists of an iterative construction of the information words that generate codewords of

This work will be presented at ISIT'02, Lausanne, Switzerland, July 2002.

minimum weight. This method keeps track of the different "candidate" information words that are under construction in a stack. For large interleavers, the memory requirements of the method can be very large.

The method that we propose in this paper is based on the notion of the Error Impulse Response of the decoder. The Error Impulse Response of the iterative Soft-In/ Soft-Out (SISO) decoding algorithm was introduced in [2]. It has also been used in [5]. The proof of our method is based on a reasoning in the Euclidean space $\mathbb{R}^n$. For the proof, we suppose that the decoder is a maximum likelihood (ML) decoder on the Gaussian channel and that the modulation is BPSK or QPSK. In practice, the turbo decoder on the Gaussian channel is used.

2 Principle of the method

We use the following notations:

- $x = (-1, -1, \cdots, -1)$ is the word associated with the "all zero" codeword by the modulation.
- $y = (-1, -1, \cdots, -1, -1 + A_i, -1, \cdots, -1)$ is the input to the decoder. A_i is a positive real number called the error impulse. i is the position of error.

We identify any codeword with a sequence of +1s and −1s. The Hamming weight of a codeword is the number of +1's in this codeword.

y is decoded according to the ML criterion on the Gaussian channel. The decoded codeword $\hat{x}$ is such that:

$$< \hat{x}, y > \geq < z, y > \qquad \forall z \in \mathcal{C} \tag{3}$$

where $< \bullet, \bullet >$ is the scalar product and $\mathcal{C}$ is the code.

Theorem 1 *If there exists a position of error i and an error impulse A_i such that the decoded codeword is not the "all zero" codeword, then*

$$A_i \ \geq \ \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \tag{4}$$

where the minimum is taken over all codewords z such that $z_i = +1$ and where $w_H(z)$ is the Hamming weight of z.

Proof:
Assume that $\hat{x} \neq x$. It results from -Eq.- (3) that:

$$< \hat{x}, y > \geq < x, y > \tag{5}$$

$< x, y >= n - A_i$. Denote by w the Hamming weight of $\hat{x}$: $w = w_H(\hat{x})$. If $\hat{x}_i = -1$ then $< \hat{x}, y >= n - 2w - A_i$ and this is conflicting. Therefore $\hat{x}_i = +1$, $< \hat{x}, y >= n - 2w + A_i$ and it results from -Eq.- (5) that :

$$n - 2w + A_i \ \geq \ n - A_i \quad \Rightarrow \quad A_i \ \geq \ w \tag{6}$$

Since $w \geq \min_{z \in \mathcal{C}/z_i = +1} w_H(z)$ we obtain

$$\min_{z \in \mathcal{C}/z_i = +1} w_H(z) \ \leq w \ \leq \ A_i \tag{7}$$

Theorem 2 *If there exists a position of error i and a positive real number A_i such that the decoded codeword is the "all zero" codeword, then*

$$A_i \ \leq \ \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \tag{8}$$

Proof

Assume that $\hat{x} = x$. It results from -Eq.- (3) that:

$$n - A_i \ \geq < z, y > \qquad \forall z \in \mathcal{C} \tag{9}$$

Consider a particular z such that $z_i = +1$ and denote by $w = w_H(z)$ its Hamming weight. Then it results from -Eq.- (9) that :

$$n - A_i \ \geq \ n - 2w + A_i \quad \Rightarrow \quad A_i \ \leq \ w \tag{10}$$

Therefore,

$$A_i \ \leq \ \min_{z \in \mathcal{C}/z_i = +1} w_H(z) \tag{11}$$

Theorem 3 *for any error position i, there exists a positive error impulse A_i^* such that*

$$\begin{array}{rl} & A_i^* = \min\{A_i \,/\, \hat{x} \neq x\} = \max\{A_i \,/\, \hat{x} = x\} \\ = & \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \end{array} \tag{12}$$

The minimum distance of the code is :

$$d_{\min} = \min_i A_i^* \tag{13}$$

Proof

It results from Th. (1) and (2) that

$$\max\{A_i \,/\, \hat{x} = x\} \ \leq \ \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \ \leq \ \min\{A_i \,/\, \hat{x} \neq x\} \tag{14}$$

Suppose that $\max\{A_i \,/\, \hat{x} = x\} \neq \min\{A_i \,/\, \hat{x} \neq x\}$. There exists A_i such that $\max\{A_i \,/\, \hat{x} = x\} < A_i < \min\{A_i \,/\, \hat{x} \neq x\}$. For this A_i, either $\hat{x} \neq x$ which is contradictory to $A_i < \min\{A_i \,/\, \hat{x} \neq x\}$, or $\hat{x} = x$ which is contradictory to $A_i > \max\{A_i \,/\, \hat{x} = x\}$. Therefore, the inequalities in -Eq.- (14) are equalities and there exists A_i^* such that:

$$\begin{array}{rl} & A_i^* = \min\{A_i \,/\, \hat{x} \neq x\} = \max\{A_i \,/\, \hat{x} = x\} \\ = & \min_{z \in \mathcal{C}, z_i = +1} w_H(z) \end{array} \tag{15}$$

Any codeword $z \neq x$ has at least one position i such that $z_i = +1$. Therefore $d_{\min}$ is equal to

$$d_{\min} = \min_i \min_{z \in \mathcal{C}, z_i = +1} w_H(z) = \min_i A_i^* \tag{16}$$

3 Algorithm

3.1 Estimation of the minimum distance as the minimum error impulse

We assume that $d_{\min}$ is in the range $[d_0, d_1]$ where d_0 and d_1 are two integers. Then $d_{\min}$ can be determined as follows.

- set $A_{\min} = d_1 + 0.5$.
- for i=1 to n do
  - $A = d_0 - 0.5$;
  - set $[(\hat{x} = x) = \text{TRUE}]$;
  - while $[(\hat{x} = x) = \text{TRUE}]$ and $(A \leq A_{\min} - 1.0)$ do
    - $A = A + 1.0$;
    - $y = (-1, \cdots, -1, -1 + A, -1, \cdots, -1)$ where $-1 + A$ is in position i;
    - ML decoding of y $\Rightarrow \hat{x}$;
    - if $(\hat{x} \neq x)$ then $[(\hat{x} = x) = \text{FALSE}]$;

    end while
  - $A_{\min} = A$

  end for
- $d_{\min}$ is the integer part of $A_{\min}$.

3.2 Practical issues

In most cases, the code is cyclic, which means that if a circular shift is applied to a codeword it is still a codeword. In this case, it is sufficient to test one position of error i to obtain the minimum distance. This is, for example, the case for convolutional codes if one neglects possible side effects due to termination.

For concatenated codes, when iterative SISO decoding is not available, the method cannot be applied because one-step decoding is largely suboptimal.

For turbo like (serially or parallel) concatenated codes the method can be applied with an iterative SISO decoder, although it has not been proved that turbo decoding is ML. This pattern of noise (no noise at all except in position i where A_i is large) is very improbable on the Gaussian channel. It is not sure that the turbo decoder maximizes $< y, z >$ since the quasi optimality of turbo decoding has been proven only by simulation with realistic patterns of noise.

Nevertheless the experiment shows good agreement between the minimum distances of TCs obtained with this algorithm and the asymptote obtained by simulation. The appropriate SISO decoder for each component code is the max-log-MAP algorithm [6]. This algorithm does not require the knowledge of the channel parameters. The extrinsic information should be passed from one component decoder to the other without any alteration (no attenuation, no saturation). The number of iterations needed for convergence depends on the expected value of $d_{\min}$. It may vary from a few tens to several hundreds for very large minimum distances. For example, in some cases, when $d_{\min}$ is around 20, the estimated $d_{\min}$ with 256 iterations is one point up on the value obtained with 128 iterations.

For the parallel concatenation of recursive systematic codes with interleaving, some restrictions should be made to the algorithm given in Section (3.1). In this case the error impulse A_i should be applied only on the systematic bits, but not on the parity bits.

When examining a turbo like concatenated code, the number of symbols to be tested is related to the periodic properties of the code. In the case of a classical turbo code, this number, denoted T, is the lowest common multiple of the period of the interleaver and of the period of the puncturing pattern, if any. The former depends on the permutation model that was adopted in the design of the interleaver. When no periodicity is observable (for instance when the permutation function was obtained by some pragmatic computer-based research), T must be maximum (i.e. $T = k$).

Multiplicity of $d_{\min}$

Because the method does not explicitly provide the multiplicity of codewords of weight $d_{\min}$ and higher, we must introduce some further hypotheses to use the asymptote on the FER given in -Eq.- (2). These assumptions seem to be in good agreement with the performance obtained by simulation, although they are irrelevant for some other codes, such as product codes.

Hyp. 1: there is only one codeword z with weight A_i^* and such that $z_i = +1$.

Hyp. 2: all the distances A_i^* obtained for the whole set of positions i $(1 \leq i \leq k)$ concern distinct codewords (there is no overlapping).

The former hypothesis is optimistic, while the latter is pessimistic. Both together, as confirmed in the examples given in the next Section, provide a good estimate of the FER, that can thus be calculated as:

$$\text{FER} \simeq \frac{1}{2} \sum_{i=1,k} \text{erfc}(\sqrt{RA_i^* \frac{Eb}{N0}}) \qquad (17)$$

Note that if the spectral thinning is pronounced the error floor will be completely determined by the minimum distance $d_{\min} = \min_i A_i^*$ and its multiplicity $n(d_{\min})$ which

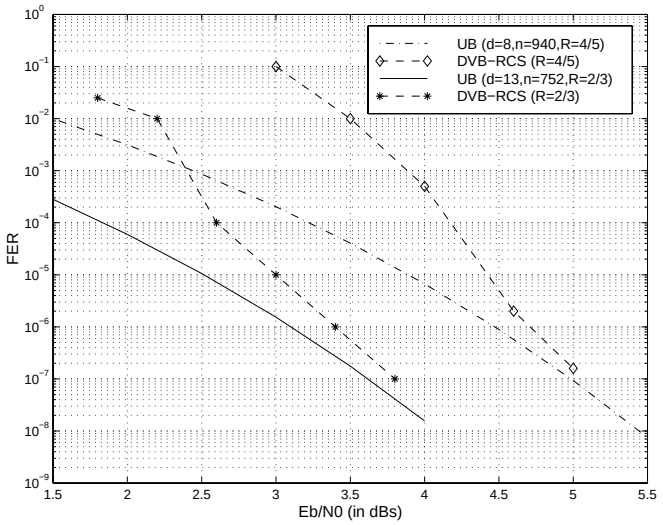

Figure 1: Frame Error Rate of the DVB-RCS turbo code for $k = 1504$ and rates $2/3$ and $4/5$. Both simulated FER and estimated FER (UB, eq. (2))) are given.

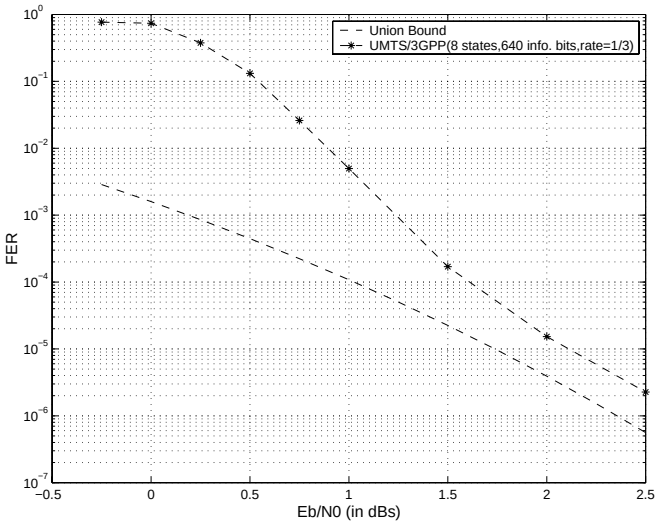

Figure 2: Frame Error Rate of the UMTS/3GPP turbo code for $k = 640$ and rate $1/3$. Both simulated FER and estimated FER (UB, eq. (17)) are given.

is estimated as the number of positions i for which $A_i^* = d_{\min}$.

4 Simulation results

4.1 DVB-RCS standard

This standard [3] uses duo-binary 8-state turbo codes. It was defined for various block sizes (12 to 206 bytes) and code rates ($R = 1/3$ to $6/7$). Figure 4.1 shows the performance in FER measured on FPGA hardware, for MPEG packets (188 bytes) and for rates $2/3$ and $4/5$. The asymptotic performance given by -Eq.- (2) is also displayed. The parameters obtained by the error impulse method are:

- $d_{\min} = 13$ and $n(d_{\min}) = 752$ for $R = 2/3$
- $d_{\min} = 8$ and $n(d_{\min}) = 940$ for $R = 4/5$

Measured and estimated curves are in good agreement.

4.2 UMTS/3GPP standard

A TC has been specified in the UMTS/3GPP [1] standard for personal communications. This standard uses binary 8-state RSC codes as component codes. The error impulse method has been tested on this turbo code for a relatively small interleaver: $k = 640$ bits. In this case the spectral thinning is not extremely pronounced because the interleaver size is moderate. Therefore, we have taken into account not only the codewords of weight $d_{\min}$ but also codewords with weights $d \geq d_{\min}$. Figure 4.2 shows the performance obtained by simulations and the asymptotic performance derived from -Eq.- (17). The "spectrum" provided by the error impulse method is the following:

$d_{\min} = 26$

$n(d), d = 26, 27, \cdots$									
						1	1	31	91	30	22
214	36	24	31	27	31	34	27	22	9
6	3	0	0	0	0	$\cdots$	$\cdots$		

where $n(d)$ denotes this time the number of positions i where the minimum error impulse is $A_i^* = d$.

There is good agreement between the performance simulated, and the performance estimated by the Error Impulse method. A loss of 0.2–0.3 dB can be observed; this loss has been observed by many contributors [2][9] and may be due to the suboptimality of the decoder.

5 Conclusion

A powerful tool for the design of linear codes, and in particular of turbo codes, has been introduced and justified. Whatever block sizes and coding rates, it gives the possibility to forecast performance at very low error rates in seconds or minutes. It may be used for instance as a fundamental algorithm in the search for good permutations in the construction of turbo codes. Nevertheless, further investigation has to be conducted to understand better the behavior of turbo decoders facing Error Impulse sequences.

References

[1] 3rd Generation Partnership Project. Multiplexing and Channel Coding (FDD), June 1999. 3G TS 25.212.

[2] C.Berrou. Some clinical aspects of turbo codes. In *Proc. of the Intern. Symp. on Turbo Codes and Rel. Topics*, pages 26–31, Brest, France, Sep. 1997.

[3] ETSI. Digital Video Broadcasting (DVB). Interaction Channel for Satellite Distribution Systems, Feb. 2000. DVB-RCS001.

[4] G.D.Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61:268–278, march 1973.

[5] J.B.Anderson. Transient and convergence properties of the BCJR decoder via a linear system model. In *Proc. of the Information Theory Workshop*, pages 135–137, Cairns, Australia, Sep. 2001.

[6] P.Robertson, P.Hoeher, and E.Villebrun. Optimal and suboptimal maximum a posteriori algorithms suitable for turbo decoding. *Europ. Trans. on Telecom.*, 8:119–125, March-Apr 1997.

[7] R.Garello, P.Pierleoni, and S.Benedetto. Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications. *IEEE Jour. on Select. Areas in Comm.*, 19(5), May 2001.

[8] S.Benedetto and G.Montorsi. Design of parallel concatenated convolutional codes. *IEEE Trans. on Comm.*, 44:591–600, May 1996.

[9] S.Benedetto, L.Gaggero, R.Garello, and G.Montorsi. On the design of binary serially concatenated convolutional codes. In *Proc. of the VIII Communication Theory Mini-Conf, CTMC*, pages 32–36, Vancouver,BC,Canada, June 1999.