

On Short-Length Error-Correcting codes for 5G-NR

Johannes Van Wonterghem^{a,*}, Amira Alloum^b, Joseph Jean Boutros^c, Marc Moeneclaey^a

^a*Dept. of Telecommunications and Information Processing, Ghent University, 9000 Ghent, Belgium,
{johannes.vanwonderghem, marc.moeneclaey}@ugent.be*

^b*Nokia Bell Labs, 91620 Nozay, France, amira.alloum@nokia-bell-labs.com*

^c*Dept. of Electrical and Computer Engineering, Texas A&M University, 23874 Doha, Qatar, boutros@tamu.edu*

Abstract

We compare the performance of a selection of short-length and very short-length linear binary error-correcting codes on the binary-input Gaussian noise channel, and on the fast and quasi-static flat Rayleigh fading channel. We use the probabilistic Ordered Statistics Decoder, that is universal to any code construction. As such we compare codes and not decoders. The word error rate versus the signal-to-noise ratio is found for LDPC, Reed-Muller, Polar, Turbo, Golay, random, and BCH codes at length 20, 32 and 256 bits. BCH and random codes outperform other codes in absence of a cyclic redundancy check concatenation. Under joint decoding, the concatenation of a cyclic redundancy check makes all codes perform very close to optimal lower bounds. Optimizations of the Ordered Statistics Decoder are discussed and revealed to bring near-ML performance with a notable complexity reduction, making the decoding complexity at very short length affordable.

Keywords: 5G, Error-correcting codes, Soft-Decision decoding, Complexity

1. Introduction

Information Theory [1] predicts the existence of good error-correcting codes that are capable of achieving channel capacity [2]. In the past half century coding theorists built many families of error-correcting codes [3],[4], to achieve the asymptotic fundamental limits predicted by Shannon.

Paradoxically, even in the finite block-length regime considered for practical communication systems, the channel codes used to be evaluated with respect to the channel capacity, until the work of Polyansky, Poor and Verdu [5] characterized with tight bounds how the non asymptotic lengths impose a severe penalty on the maximum achievable rate. Moreover, Shannon capacity is a poor benchmark in the context of the coming fifth generation of mobile networks (5G), where a plethora of services such as internet of things and augmented reality are constrained to support real-time transmissions for short packets within few milliseconds. Accordingly, the design and analysis of error-correcting codes in the short block-length regime ignited a spark of interest in the coding theory community in both academia and industry.

Our paper is dedicated to error-correcting codes of short length (typically 256 bits) and very short length (less than 32 bits). We compare the performance of the codes under equal-complexity identical decoding conditions based

on a universal decoder. We consider the binary input additive white Gaussian noise (BI-AWGN) channel, and fast and quasi-static flat Rayleigh fading channels. Over the years, many different decoding strategies have been developed [4],[6]. These strategies are often specific to one family of error-correcting codes, or are sub-optimal, favoring decoding speed over performance. For our comparison, we use the Ordered Statistics Decoder (OSD) which is a universal decoder that can decode any linear binary block code and is also near-optimal. As a result we compare codes and not decoders. The authors of [7] also compared the performance of different short-length codes (128 bits) on the BI-AWGN channel, but did not use the same decoding strategy for all codes. Some codes were decoded using a near-optimal decoder, whereas for other codes a sub-optimal decoder was used. As a result it is not always clear if one error-correcting scheme performs better than the other because of the choice of its error correcting code, or because of its decoder.

Complexity is not the main issue of this paper, but is treated for very short-length error-correcting codes for the 5G use-case where the OSD is revealed to be a viable option for practical applications.

The paper is structured as follows. System model and notations are described in Section 2. Section 3 explains OSD decoding. The discussion on the performance of short-length error-correcting codes is provided in Section 4. Very short-length codes are treated in Section 5. We conclude in the final section.

*Corresponding author

This paper is an extension of the work published in [8] and [9]. In addition to the BI-AWGN channel, we extend the study in the current paper to the fast and quasi-static flat Rayleigh fading channels. Furthermore, the section on very short-length codes is extended by new simulation results and more extensive analysis of the results.

2. System Model and Notations

At the transmitter we consider a length- k binary information message $\mathbf{b} = (b_1, \dots, b_k)$, with $P(b_i=0)=P(b_i=1)=1/2$ independent and identically distributed (i.i.d.). The information message is encoded into a binary coded message $\mathbf{c} = (c_1, \dots, c_n)$ of length n using a linear binary block code C , completely specified by its $k \times n$ generator matrix G [3]. In systematic form we have $G = [I_k|P]$, where I_k is the $k \times k$ identity matrix and P is a parity matrix defining the code. The encoding operation can be written as $\mathbf{c} = \mathbf{b}G = [\mathbf{b}|\mathbf{p}]$ with \mathbf{p} the parity bits corresponding to \mathbf{b} .

We consider transmission over the BI-AWGN channel. The coded message is first mapped to a BPSK symbol sequence $\mathbf{s} \in \{-1, +1\}^n$ using the rule $s_i = 2c_i - 1$ and then transmitted over the AWGN channel characterized by its single sided noise spectral density N_0 . At the output of the channel, we receive $\mathbf{r} = \mathbf{s} + \mathbf{w}$ where $\mathbf{w} = (w_1, \dots, w_n)$ is a set of i.i.d. real Gaussian random variables with zero mean and variance $\sigma^2 = N_0/2$. The symbols s_i are normalized to unit energy such that the energy transmitted per information bit equals $E_b = \frac{n}{k} = \frac{1}{R}$.

At the receiver, soft-decision decoding is performed to construct an estimate $\hat{\mathbf{b}}$ of the originally transmitted information message \mathbf{b} . For this estimate, the decoder makes use of two vectors corresponding to the sign and magnitude of the received signal \mathbf{r} :

The hard-decision $\mathbf{y} = [\mathbf{b}_{\text{HD}}|\mathbf{p}_{\text{HD}}]$ where

$$y_i = \begin{cases} 0 & \text{for } r_i < 0 \\ 1 & \text{for } r_i \geq 0 \end{cases}, \quad (1)$$

and the confidence values

$$\alpha_i = |r_i|, \quad i = 1 \dots n. \quad (2)$$

To understand that α_i is indeed a measure for the confidence of the received r_i , it suffices to see that the log-likelihood ratio is $\Lambda_i = \log \frac{P(c_i=0|r_i)}{P(c_i=1|r_i)} = \frac{2r_i}{\sigma^2}$ for the BI-AWGN.

We also consider fast and quasi-static flat Rayleigh fading channels. At the output of the fading channel we receive $r_i = g_i s_i + w_i$, where s_i and w_i are defined as on the BI-AWGN channel, and g_i is the fading gain at time instant i . We consider complete knowledge of channel state information (CSI) at the receiver, such that g_i is known at the receiver.

Note that the the hard-decision rule does not change compared to the BI-AWGN channel model, but that the confidence values change to

$$\alpha_i = |g_i r_i|, \quad i = 1 \dots n, \quad (3)$$

because now $\Lambda_i = \log \frac{P(c_i=0|r_i, g_i)}{P(c_i=1|r_i, g_i)} = \frac{2g_i r_i}{\sigma^2}$.

For both the fast and quasi-static fading channels, the fading gains are i.i.d. Rayleigh distributed. On the fast fading channel, g_i changes for every symbol; on the quasi-static fading channel there are n_c independent fading gains per codeword, each affecting n/n_c symbols.

3. Soft-decision decoding using the OSD algorithm

Soft-decision decoding by the receiver is performed using the OSD algorithm, an efficient most reliable basis decoding algorithm proposed by Dorsch [10], further developed by Fang and Battail [11], and later analyzed and revived by Fossorier and Lin [12]. In the first step of the algorithm, the received vector \mathbf{r} is sorted in order of descending confidence and the corresponding permutation π_1 is applied to the generator matrix G , yielding G' . Gaussian elimination is now performed on G' to construct the systematic \tilde{G} , note that an additional permutation π_2 may be necessary. We write $\tilde{\mathbf{y}} = \pi_2(\pi_1(\mathbf{y})) = [\tilde{\mathbf{b}}_{\text{HD}}|\tilde{\mathbf{p}}_{\text{HD}}]$ and $\tilde{\boldsymbol{\alpha}} = \pi_2(\pi_1(\boldsymbol{\alpha}))$. The vector $\tilde{\mathbf{b}}_{\text{HD}}$ corresponds to the most-reliable independent positions of the received vector \mathbf{r} .

During the OSD algorithm, test-error patterns (TEPs) \mathbf{e}_i of increasing weight are generated and added to the hard-decision information bits $\tilde{\mathbf{b}}_{\text{HD}}$ on the MRB. The corresponding codeword $\tilde{\mathbf{c}}_i$ is obtained by re-encoding via the systematic generator matrix \tilde{G} . The trivial TEP $\mathbf{e}_0 = \mathbf{0}$ results in the order-0 OSD codeword $\tilde{\mathbf{c}}_0 = (\tilde{\mathbf{b}}_{\text{HD}} + \mathbf{e}_0) \cdot \tilde{G} = \tilde{\mathbf{b}}_{\text{HD}} \cdot \tilde{G}$. The TEP \mathbf{e}_j results in codeword $\tilde{\mathbf{c}}_j = \tilde{\mathbf{c}}_0 + \mathbf{e}_j \cdot \tilde{G}$. Undoing the permutations yields the estimate $\hat{\mathbf{c}}_j = \pi_1^{-1}(\pi_2^{-1}(\tilde{\mathbf{c}}_j))$ of the original codeword \mathbf{c} .

After every re-encoding operation, the Euclidean distance between the OSD codeword $\hat{\mathbf{c}}_j$ and the received vector \mathbf{r} is calculated. If this distance is lower than that of the current best OSD codeword, we select $\hat{\mathbf{c}}_j$ as the new best codeword estimate. For BPSK modulation, minimizing the Euclidean distance is equivalent to minimizing the weighted Hamming distance

$$\text{WHD}_j = \sum_{\substack{1 \leq i \leq n \\ \hat{c}_{j,i} \neq y_i}} \alpha_i. \quad (4)$$

The OSD algorithm is terminated after a predetermined number of re-encodings. For example, in OSD order 2, the

following patterns are generated:

weight 0	weight 1	weight 2
000...000	000...001	000...011
	000...010	000...101

	100...000	110...000

It follows that in OSD order 1, $1 + k$ patterns are generated, in OSD order 2 we generate $1 + k + \frac{1}{2}k(k - 1)$ patterns, etc. Hence the complexity of the algorithm is $O(k^{\text{OSD order}})$. In [12] it was shown that order- l reprocessing is asymptotically optimal (close to ML) for

$$l \geq \min \{ \lceil d_H/4 - 1 \rceil, k \}, \quad (5)$$

such that the complexity is determined by both k and the minimum Hamming weight of the code, d_H . Choosing the order lower than the optimal, allows a performance-complexity trade-off. In the subsequent sections, we compare codes at the best possible decoding performance, unless noted otherwise.

3.1. Fast OSD Improvements

In the literature, several improvements to the original OSD algorithm have been presented that aim to reduce the complexity of the optimal decoder and offer a finer performance-complexity trade-off [13]-[19].

In our implementation¹ of the OSD algorithm we used the probabilistic necessary condition from [17], the probabilistic sufficient condition [15], the reference re-encoding scheme [17], the preprocessing rules from [16], and the multiple biases diversity scheme from [18].

To determine if the decoder performs (near-)optimally for the chosen parameters of the algorithm, we make use of an ML lower bound calculated during computer simulation. Whenever the decoder outputs an erroneous estimate of the originally transmitted information word, the Euclidean distance between the original codeword \mathbf{c} and the received vector \mathbf{s} is evaluated. If this distance is larger than the distance between the decoder output $\hat{\mathbf{c}}$ and \mathbf{s} , then the ML decoder would also have made an erroneous decision.

Algorithm 1 gives an overview of the steps taken during the decoding process. The OSD order here is limited to 1, an extension to higher orders is trivial. Two optimizations to the OSD are shown: the probabilistic necessary and sufficient conditions. The vector $\mathbf{1}_j$ is defined as having a one at index j and a zero at all other indices; λ is the parameter of the probabilistic necessary condition and τ is the parameter of the probabilistic sufficient condition.

¹Our implementation can be found at <http://www.josephboutros.org/coding/shortcodes>. If you use it, please cite this paper.

Algorithm 1 OSD algorithm

Require: Parameters λ and τ , generator matrix G

- 1: Receive $\mathbf{r} = \{r_1, \dots, r_n\}$
 - 2: Derive $y_i = \begin{cases} 0 & \text{for } r_i < 0 \\ 1 & \text{for } r_i \geq 0 \end{cases}$ and $\alpha_i = |r_i|$
 - 3: Sort in order descending confidence α : $G, \mathbf{y} \xrightarrow{\pi_1} G', \mathbf{y}'$
 - 4: Gaussian elimination on G' : $G' \xrightarrow{\text{GE}} \tilde{G}, \mathbf{y}' \xrightarrow{\pi_2} \tilde{\mathbf{y}}$
 - 5: Calculate $\rho = \frac{\sum_{i=1}^k \tilde{\alpha}_i}{\sum_{i=1}^k \tilde{\alpha}_i + \lambda \sum_{i=k+1}^n \tilde{\alpha}_i}$
- Order 0 OSD**
- 6: Re-encode: $\tilde{\mathbf{c}} = \tilde{\mathbf{y}}_{[1,k]} \cdot \tilde{G}$
 - 7: Calculate $\hat{\mathbf{c}}_{\text{opt}} = \pi_1^{-1}(\pi_2^{-1}(\tilde{\mathbf{c}}))$, $\text{WHD}_{\text{opt}} = \sum_{\substack{1 \leq i \leq n \\ \hat{\mathbf{c}}_{\text{opt},i} \neq y_i}} \alpha_i$
 - 8: Calculate $\text{psc} = [\tilde{\mathbf{y}}_{[1,k]} | \tilde{\mathbf{y}}_{[k+1,n]}] \cdot \tilde{H}^T = \tilde{\mathbf{y}} \cdot \tilde{H}^T$
 - 9: **if** $\text{psc} < \tau$ **then return** $\hat{\mathbf{c}}_{\text{opt}}$
 - 10: Calculate $\text{pnc} = \rho \text{WHD}_{\text{opt}}$

Order 1 OSD

- 11: **for** $j=1:k$ **do**
 - 12: **if** $\tilde{\alpha}_j \leq \text{pnc}$ **then**
 - 13: Re-encode: $\tilde{\mathbf{c}} = (\tilde{\mathbf{y}}_{[1,k]} \oplus \mathbf{1}_j) \cdot \tilde{G}$
 - 14: Calculate $\hat{\mathbf{c}} = \pi_1^{-1}(\pi_2^{-1}(\tilde{\mathbf{c}}))$, $\text{WHD} = \sum_{\substack{1 \leq i \leq n \\ \hat{\mathbf{c}},i \neq y_i}} \alpha_i$
 - 15: **if** $\text{WHD} < \text{WHD}_{\text{opt}}$ **then**
 - 16: $\hat{\mathbf{c}}_{\text{opt}} = \hat{\mathbf{c}}$, $\text{WHD}_{\text{opt}} = \text{WHD}$
 - 17: Calculate $\text{psc} = [(\tilde{\mathbf{y}}_{[1,k]} \oplus \mathbf{1}_j) | \tilde{\mathbf{y}}_{[k+1,n]}] \cdot \tilde{H}^T$
 - 18: **if** $\text{psc} < \tau$ **then return** $\hat{\mathbf{c}}_{\text{opt}}$
 - 19: Calculate $\text{pnc} = \rho \text{WHD}_{\text{opt}}$
 - 20: **return** $\hat{\mathbf{c}}_{\text{opt}}$
-

The OSD algorithmic complexity is related to several terms:

- D_{fixed} denotes the computational complexity related to sorting and Gaussian elimination per codeword and to the calculation of ρ . The complexity of Gaussian elimination is $O(n \times (n - k)^2)$. Efficient sorting algorithms have an average time complexity of $O(n \log n)$.
- n_{skipped} TEPs are skipped and not re-encoded, thanks to the probabilistic necessary condition. D_{skipped} denotes the complexity involving a comparison and $(l - 1)$ sums in the real domain, l is the Hamming weight of the TEP.
- n_{accepted} TEPs are left to be re-encoded. D_{accepted} denotes the related complexity cost involving a re-encoding by $(n - k)$ binary XOR operations if intermediate results are saved, or $(k(n - k))$ for a trivial implementation. In addition, the weighted Hamming distance is calculated by approximately $(n - k)$ real sums and compared to the probabilistic necessary condition threshold.

- n_{newbest} TEPs are better than the current best pattern. When this happens, the probabilistic sufficient condition is checked at a complexity cost denoted D_{newbest} involving binary multiplication of a length- n vector with a $n \times (n - k)$ matrix and updating the probabilistic necessary condition by a multiplication in the real domain.

Hence, the total computational complexity becomes:

$$\begin{aligned} D_{\text{tot}} &= D_{\text{fixed}} + n_{\text{skipped}} D_{\text{skipped}} \\ &\quad + n_{\text{accepted}} D_{\text{accepted}} + n_{\text{newbest}} D_{\text{newbest}} \\ &\approx D_{\text{fixed}} + n_{\text{accepted}} D_{\text{accepted}} \end{aligned} \quad (6)$$

The approximation is motivated heuristically by observing that ($n_{\text{newbest}} \ll n_{\text{accepted}}$) and ($D_{\text{skipped}} \ll D_{\text{accepted}}$).

4. Performance Comparison of Short-Length Error-Correcting Codes

In this section the following codes are compared:

- Polar codes: The code length is $n = 2^\ell$. Take Arikan's kernel G_2 [22] and build its Kronecker product ℓ times, i.e. build $G_2^{\otimes \ell}$. Then select the k rows from $G_2^{\otimes \ell}$ corresponding to highest mutual information channels after ℓ splittings [22]. The generator matrix of the code is found by exact splitting and adapted to each value of the channel parameter. We used density evolution [6] to split the channel and construct the code [23]. Polar codes are not considered on the fading channels because for these channels the SNR, which is a parameter used in the design of polar codes, is not fixed.
- Reed-Muller codes: A (256,128) Reed-Muller subcode is constructed from the (256,163) Reed-Muller code using the greedy algorithm described in [20].
- BCH codes: Standard binary primitive (n, k, t) BCH codes are built from their generator polynomial [3], [4]. Extension by one parity bit is made to get an even length.
- LDPC codes: Regular (3,6) LDPC codes are built from a random bipartite Tanner graph [6]. Length-2 cycles are avoided, the number of length-4 cycles is reduced, no other constraint was applied to the graph construction.
- Parallel Turbo codes: we consider 16-state RSC $(23, 35)_8$ as constituent codes [21]. Both trellises are terminated and puncturing is applied to get a rate close to 1/2. We used a size-128 S-random interleaver [24] with parameter $S = 8$.
- Random codes: The generator matrix of the random code is filled by a white Bernoulli(1/2) process.

The use of a cyclic redundancy check (CRC) code to improve list decoding of polar codes was introduced by I. Tal and A. Vardy [25]. Here, given the universal nature of OSD decoding on the BI-AWGN channel, the CRC code was jointly decoded with all of the codes listed above to investigate its influence on the performance. The joint generator matrix is simply the product of the CRC matrix with the generator matrix of the original code C . Let G be the $k \times n$ generator matrix of C . Let G_{CRC} be the $(k - m) \times k$ generator matrix of the CRC code, where m is the degree of the CRC polynomial. Then, joint OSD decoding is based on the following generator matrix:

$$G_{CRC} \times G. \quad (7)$$

The serial concatenation has the CRC as outer code and the original error-correcting code C as inner code. This corresponds to selecting a subcode from the original error-correcting code. We consider $m = 16$ redundancy bits and the LTE CRC code

$$g(x) = x^{16} + x^{12} + x^5 + 1. \quad (8)$$

We ran computer simulations to obtain the performance of the binary codes listed in this section. Randomly generated data was transmitted using the system described in section 3. We considered the BI-AWGN channel, the fast Rayleigh fading channel, and the quasi-static flat Rayleigh fading channel with $n_c = 4$. At every considered value of E_b/N_0 , codewords were generated, transmitted, and decoded until 100 word errors occurred on the BI-AWGN and 200 word errors occurred on the fading channels. During the computer simulation, the ML lower bound was also recorded but we omit it from the figures to keep the graphs as clear as possible. The OSD parameters were chosen such that the performance is near-ML and the ML lower bound (almost) coincides with the actual simulated performance of the code. The normal approximation of the Polyanskiy-Poor-Verdú (PPV) bound [5] is shown for the BI-AWGN channel. The evaluation of this bound is described in [26]².

Figure 1 shows the word error rate (WER) versus the SNR for the considered codes, all under OSD decoding on the BI-AWGN channel. The performance of the LDPC code under iterative belief-propagation (BP) decoding is also included. No CRC is used for this performance comparison. Note that the PPV bound is plotted twice, once for the codes of rate $\approx 1/2$ and once for the Turbo code. The binary (256,131) BCH code and the (256,128) random code outperform all other codes. This observation is in line with [7], where the (128,64) BCH code was observed to perform very close to the normal approximation of the PPV bound.

Figure 2 shows the performance of the same codes concatenated with the 16-bit LTE CRC code. Over the considered

²Routines for the numerical evaluation of this bound can be found at <https://github.com/yp-mit/spectre>.

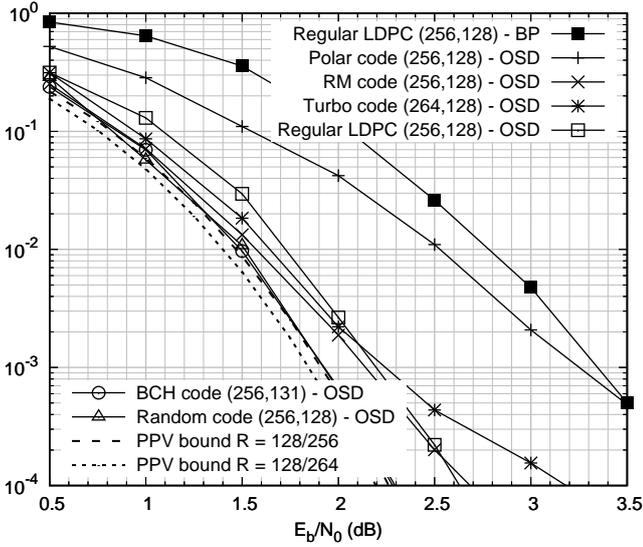


Figure 1: Word error rate versus signal-to-noise ratio (SNR). Performance comparison of codes with length 256 and rate 1/2 on the BI-AWGN. No CRC.

SNR range, all codes exhibit a performance within a small range of error rates approaching the PPV bound because the CRC selects a subcode from the original generator matrix that has a performance close to that of a random code.

For cyclic codes, e.g. BCH codes, the CRC concatenation corresponds to a multiplication of two generator polynomials. A suitable CRC generator polynomial can always be chosen such that the concatenation is BCH. Indeed, a judicious choice of $g_{\text{CRC}}(x)$ and

$$g_{\text{BCH}_2}(x) = g_{\text{BCH}_1}(x) \times g_{\text{CRC}}(x)$$

increases the error-correcting capability t of $g_{\text{BCH}_1}(x)$ to that of $g_{\text{BCH}_2}(x)$. A bad choice for the CRC polynomial may lead to a concatenation with weak performance. A CRC polynomial that has roots already contained in $g_{\text{BCH}_1}(x)$ does not improve the original BCH code.

For non-cyclic codes the above explanation cannot be used, but it's clear that different CRC choices will lead to differences in the performance of the concatenation. Indeed, the CRC will determine which codewords from the original code will be represented in the subcode and hence implies the Hamming weight distribution of this subcode.

The union bound on the binary erasure channel (BEC), based on the weight distribution, is a simplified method to determine the code performance and compare it with

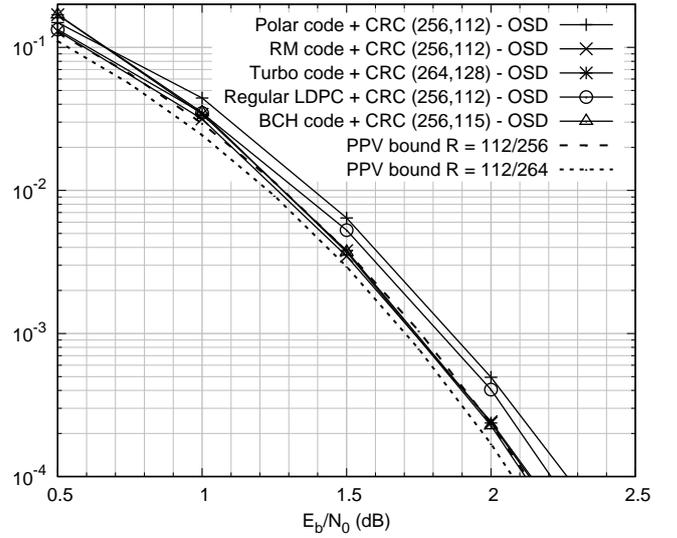


Figure 2: Word error rate versus SNR. Performance comparison of codes with length 256 and rate 1/2 on the BI-AWGN. A 16-bit CRC is concatenated with all codes.

other codes:

$$\begin{aligned} Pr[\text{word error}] &= Pr[\text{word error}|0 \text{ transmitted}] \\ &= Pr[\text{nonzero codeword covered}] \\ &= Pr\left[\bigcup_{c \in C \setminus 0} c \text{ covered}\right] \\ &\leq \sum_{c \in C \setminus 0} Pr[c \text{ covered}] \\ &= \sum_{w \geq d_{\min}} A_w \epsilon^w, \end{aligned} \quad (9)$$

where A_w is the number of codewords of weight w , d_{\min} is the minimum distance of the code, and ϵ is the BEC erasure probability. Thus, the union bound corresponds to evaluating the weight enumerator polynomial at the BEC erasure probability. For length-64 non-cyclic codes, we used (9) to exhaustively check that the 16-bit LTE CRC is among the best of all 16-bit CRC codes.

Figure 3 shows the WER versus the SNR on the fast fading channel with CSI at the receiver. Note that the introduction of fast fading has resulted in a loss of approximately 2 dB compared to the BI-AWGN channel. However the ordering of the considered codes' performances is the same.

The concatenation of the 16-bit LTE CRC code in Figure 4 has a similar result as on the BI-AWGN channel. All codes exhibit a performance within a small range of error rates and are close to the performance of the random code.

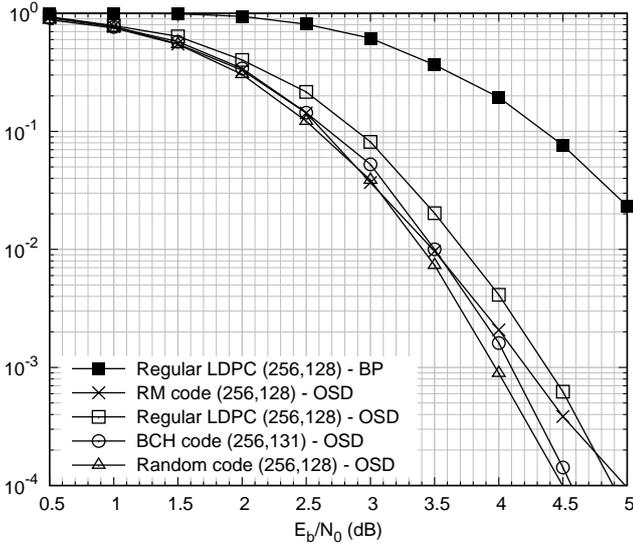


Figure 3: Word error rate versus SNR. Performance comparison of codes with length 256 and rate 1/2 for fast fading with CSI at the receiver. No CRC.

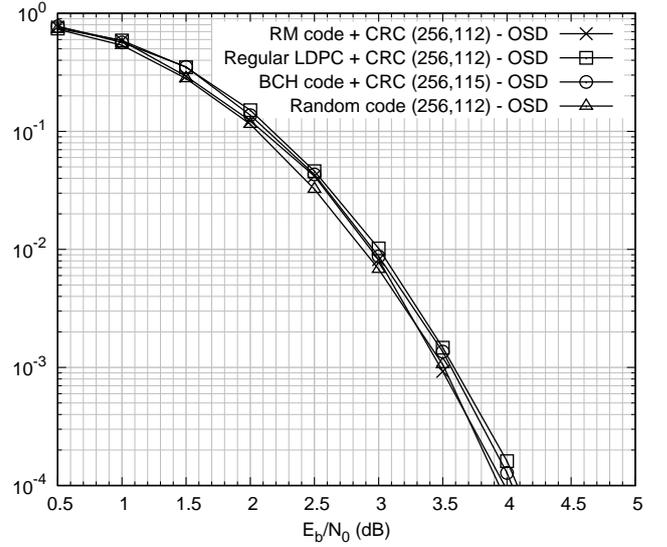


Figure 4: Word error rate versus SNR. Performance comparison of codes with length 256 and rate 1/2 for fast fading with CSI at the receiver. A 16-bit CRC is concatenated with the codes.

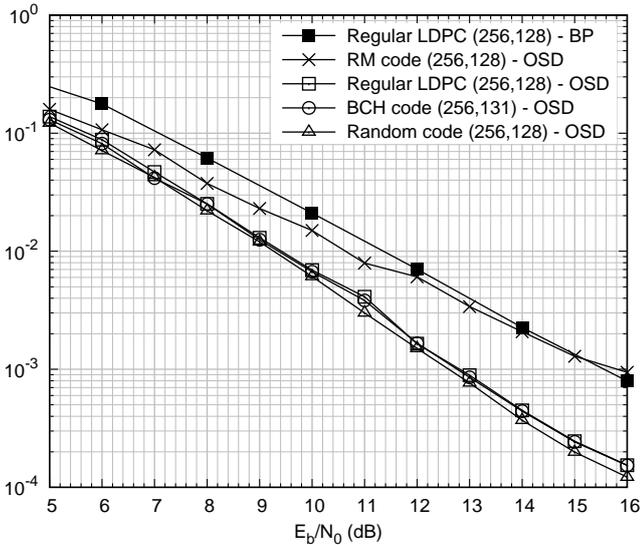


Figure 5: Word error rate versus SNR. Performance comparison of codes with length 256 and rate 1/2 for quasi-static flat fading with $n_c = 4$ and CSI at the receiver. No CRC.

Figure 5 shows the WER versus the SNR on the quasi-static flat fading channel with $n_c = 4$ and CSI at the receiver. Note that the slope of the word error rate curves has changed when compared with the results on the BI-AWGN and the fast fading channel. To understand this, we need to consider the diversity order attained by the

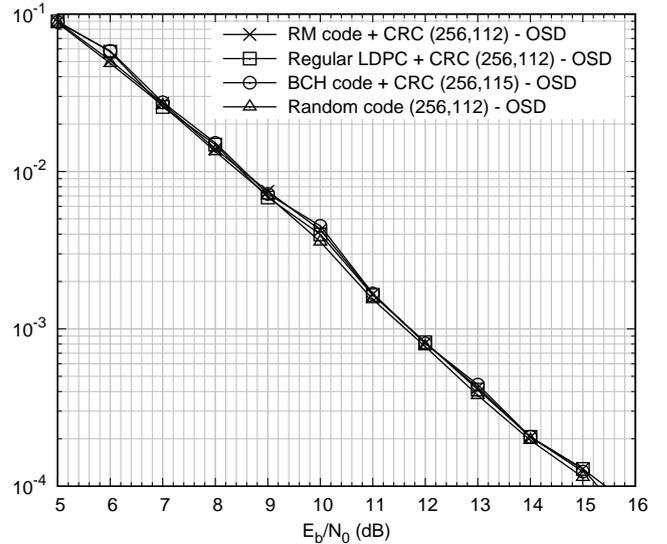


Figure 6: Word error rate versus SNR. Performance comparison of codes with length 256 and rate 1/2 for quasi-static flat fading with $n_c = 4$ and CSI at the receiver. A 16-bit CRC is concatenated with the codes.

codes, and defined as [27]:

$$\text{diversity} = - \lim_{\gamma \rightarrow +\infty} \frac{\log P_{ew}}{\log \gamma}, \quad (10)$$

where P_{ew} is the WER *after decoding* and γ is the average SNR per symbol: $\gamma = E_s/N_0 = RE_b/N_0$. I.e., diversity is the WER slope at high SNR, and it is decoder dependent.

The diversity order attained by a code of rate R on a chan-

nel with n_c independent fading gains per codeword can be upper bounded by the following Singleton-like bound [28]:

$$\text{diversity} \leq 1 + \lfloor n_c(1 - R) \rfloor. \quad (11)$$

For $n_c = 4$, we obtain that $\text{diversity} \leq 3$ for codes with $R = 1/2$. As shown in [29], the diversity order under ML decoding can be evaluated by calculating the rank of the parity check sub-matrices corresponding to the different fading states. By applying this rule, we found that the exact diversity order is 2 for all considered codes. Root-LDPC should be employed to guarantee maximal diversity. Luckily, at finite SNR, the BCH code, the random code, and the LDPC code under OSD decoding all exhibit an effective diversity order of 3 as plotted in Fig. 5.

Figure 6 shows one more time that the CRC concatenation brings the codes to the same error performance together with the random code. In this specific case, both simulation results and analytical calculations of the diversity order show that all codes attain a triple diversity order.

5. Performance Comparison of Very Short-Length Error-Correcting Codes: A 5G-NR Case Study

In this section we provide a comparison of very short-length error-correcting codes that competed for encoding the 5G-NR control channel below 22 information bits. In the LTE standard, the signalling encoding included Reed-Muller codes, convolutional codes and repetition codes [30]. In the current described context the proposed encoders and decoders are:

- LTE Reed-Muller codes: denoted RM and described in [31]. In practice binary Reed-Muller codes are decoded with a Fast Hadamard Transform (FHT) decoder [31].
- Reed-Muller-based codes: denoted RM-based and described in [32], are decoded with the FHT decoder.
- Golay-based codes: described in [33], and can be decoded by ML-decoding or OSD.
- Parity-check Polar codes: denoted PC-Polar, described in [34][35] as parity check and polar code concatenation; decoded with successive cancellation list decoding [25].

We consider information length $k \in [3, 11]$, and the code length n equals 20 or 32 bits. Independently of the decoders proposed above, we consider the OSD of order 2, using probabilistic necessary and sufficient conditions, as a common decoder. At the considered very short lengths, the complexity of the OSD algorithm is no longer prohibitive and the algorithm is shown to become a viable option for practical applications.

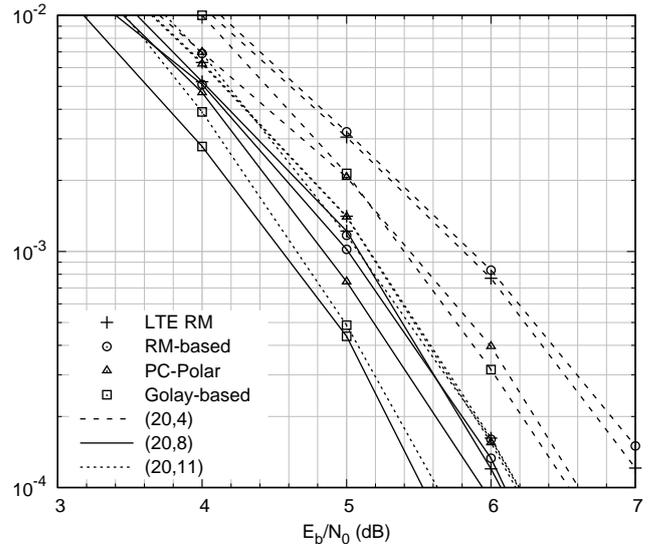


Figure 7: Word error rate versus SNR. Comparison of very short linear block codes of length 20 on the BI-AWGN.

Figure 7 shows a performance comparison in terms of WER for $n = 20$ with carefully selected parameters for the OSD enabling near ML performance. The results show that the Golay-based code outperforms other code families within the considered range, and that both RM codes have approximately the same performance. Both observations are consistent with the minimum distances provided in Table 1 together with optimal bounds [36]. Let us notice that the PC-Polar codes with $k = 4$ and 8 have the same minimum distance as the corresponding RM codes, but Table 2 shows that they have less codewords at the minimum distance, explaining the performance difference. At $k = 11$ the number of minimum weight codewords is approximately the same for both RM and PC-Polar.

Figures 8 and 9 show the SNR, expressed in terms of E_b/N_0 , at a WER target of 10^{-2} for very short RM-based and Golay-based codes of codeword length 20 and 32.

In the case of a code-length of 20 bits, it is shown that the Golay-based code is the best candidate at almost all considered values of k , except for 3 and 10. This is in line with the minimum distances in Table 1.

At the code-length of 32 bits, no common best code to all the considered dimensions is revealed. As for $k \in \{3, 4, 5\}$, each code outperforms the others once. When k varies between 6 and 11 all codes perform alike, again with each one outperforming the others an equal number of times. Looking at Table 1, let us notice that the performance in Figure 9 is not entirely in line with the minimum distance behaviour. This can be explained by noting that at low SNR the union upper bound on the WER is not dominated by the codewords at minimum distance. As a result the complete weight distribution of the code needs to be taken

Table 1: Comparison of the minimum distance of the considered codes.

N	K	3	4	5	6	7	8	9	10	11
20	LTE RM	8	8	8	8	6	6	6	6	4
	RM-based	11	8	8	8	6	6	6	6	4
	PC-Polar	11	8	8	8	6	6	6	6	4
	Golay-based	10	10	8	8	8	8	7	6	5
	Bound	11	10	9	8	8	8	7	6	5
32	LTE RM	16	16	16	16	12	12	12	12	10
	RM-based	18	16	16	16	12	12	12	12	12
	Golay-based	17	16	16	13	12	11	11	10	9
	Bound	18	16	16	16	14	13	12	12	12

Table 2: Weight distributions of some of the considered codes.

N	K	Name	Weight enumerator polynomial
20	4	LTE RM	$3x^8 + 8x^{10} + 3x^{12} + x^{20}$
		RM-based	$3x^8 + 8x^{10} + 4x^{12}$
		PC-Polar	$x^8 + 4x^{10} + 8x^{11} + 2x^{12}$
		Golay-Based	$11x^{10} + 3x^{12} + x^{14}$
20	11	LTE RM	$10x^4 + 170x^6 + 485x^8 + 716x^{10} + 485x^{12} + 170x^{14} + 10x^{16} + x^{20}$
		RM-based	$11x^4 + 164x^6 + 500x^8 + 696x^{10} + 500x^{12} + 164x^{14} + 11x^{16} + x^{20}$
		PC-Polar	$10x^4 + 180x^6 + 448x^8 + 760x^{10} + 478x^{12} + 148x^{14} + 23x^{16}$
		Golay-Based	$16x^5 + 120x^6 + 240x^7 + 130x^8 + 160x^9 + 528x^{10} + 480x^{11} + 120x^{12} + 80x^{13} + 120x^{14} + 48x^{15} + 5x^{16}$
32	4	LTE RM	$14x^{16} + x^{32}$
		RM-based	$15x^{16}$
		Golay-Based	$3x^{16} + 8x^{17} + 4x^{18}$
32	11	LTE RM	$64x^{10} + 240x^{12} + 448x^{14} + 542x^{16} + 448x^{18} + 240x^{20} + 64x^{22} + x^{32}$
		RM-based	$496x^{12} + 1054x^{16} + 496x^{20} + x^{32}$
		Golay-Based	$x^9 + 31x^{10} + 99x^{11} + 122x^{12} + 155x^{13} + 189x^{14} + 225x^{15} + 333x^{16} + 319x^{17} + 217x^{18} + 141x^{19} + 82x^{20} + 69x^{21} + 43x^{22} + 15x^{23} + 6x^{24}$

into account. In Table 1, we observe that at $k = 4$ all codes have the same minimum distance, though the number of codewords of minimum distance is different from one code to another. Whereas at $k = 11$, all codes show different minimum distances but perform alike at the target rate of 10^{-2} due to their different weight distributions.

For the complexity analysis of the OSD, we consider the LTE RM code with $(k, n) = (8, 32)$ where the ML decoder is known to consider 256 candidates whereas the FHT decoder considers 128 correlations [31]. In Fig. 10, the average number of accepted patterns $\bar{n}_{\text{accepted}}$ for different sets of OSD parameters and SNR is plotted. We notice that the number of checked patterns is not only dependent on the parameters λ and τ of the probabilistic necessary and sufficient conditions, but that it also varies with the SNR

as a result of these two optimizations.

In Fig. 11 we provide the WER obtained with the same choice of OSD parameters, together with the corresponding ML lower bound. Note that for a well chosen set of parameters, order 1 OSD approaches ML performance, while order 2 OSD reaches it.

The results show that the fast OSD improvements drastically reduce the number of checked patterns, especially at higher SNR values. The algorithm adapts itself to the current SNR by checking more patterns when the channel introduces many errors and checking less patterns when the channel introduces less errors. Carefully choosing the optimisation parameters λ and τ allows lowering the complexity significantly, without impacting the ML error performance reached at order 2.

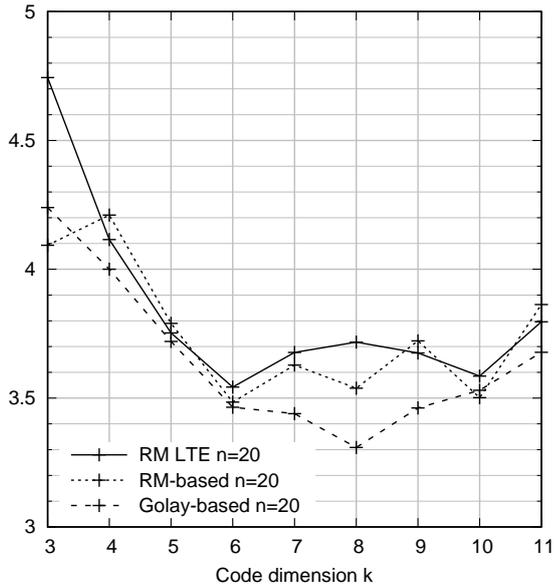


Figure 8: SNR versus code dimension at a target WER of 10^{-2} . Comparison of very short linear block codes of length 20.

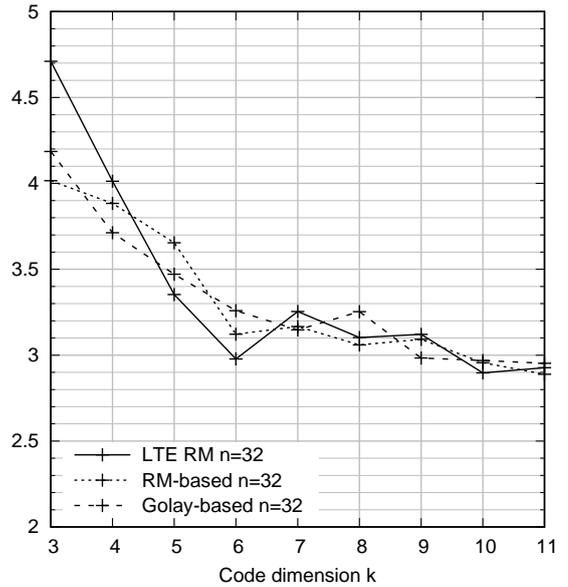


Figure 9: SNR versus code dimension at a target WER of 10^{-2} . Comparison of very short linear block codes of length 32.

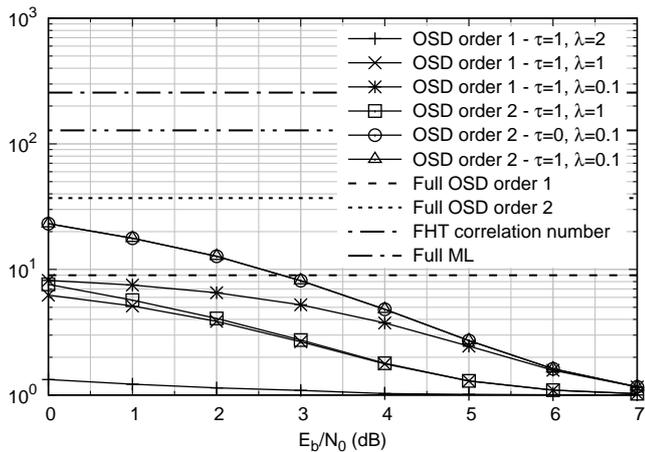


Figure 10: Average number of accepted patterns versus SNR while decoding the (32,8) LTE Reed-Muller code.

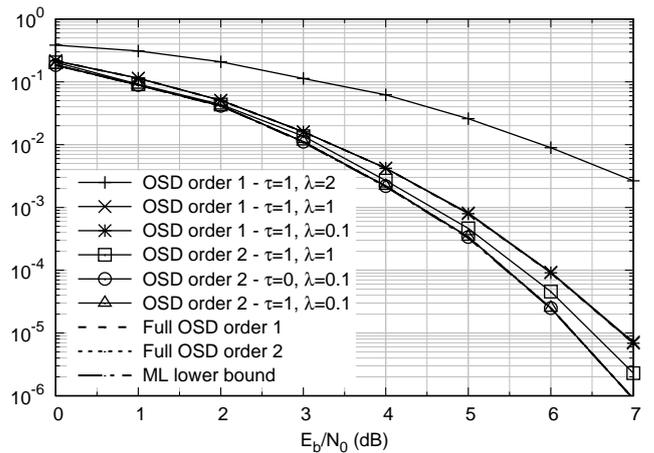


Figure 11: Word error rate versus signal-to-noise ratio for decoding the (32,8) LTE Reed-Muller code.

6. Conclusions

The OSD soft-decision universal near optimal decoder with complexity improvements was described and used to compare the performance of short-length linear binary codes on the BI-AWGN channel, the fast Rayleigh fading channel and the quasi-static flat Rayleigh fading channel. The word error rate versus the SNR was plotted for LDPC, Reed-Muller, Polar, Turbo, BCH, and random codes. From simulation results, we conclude that the BCH and random codes outperform other code families. This behaviour

changes when we concatenate the considered codes with a CRC and perform joint decoding. Resulting in an almost equal performance approaching that of the random code over the considered range of SNR values.

A performance comparison of very short-length codes considered for 5G signalling was also carried out. Complexity of the OSD was analysed and shown to be competitive with other ML decoders without sacrificing the ML performance. At a code length of 20 bits the Golay-based code outperforms the other considered codes, at a length of 32 bits, no common best code to all considered dimensions is revealed.

Acknowledgment

Johannes Van Wouwerghem would like to thank the Research Foundation in Flanders (FWO) for funding his PhD fellowship. Amira Alloum would like to thank her colleagues of 5G-NR RAN1 channel coding working group for the valuable discussions.

References

- [1] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 2006.
- [2] C.E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell System Technical Journal*, vol. 38, no. 3, pp. 611-656, May 1959.
- [3] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.
- [4] R.E. Blahut, *Algebraic codes for data transmission*, Cambridge University Press, 2003.
- [5] Y. Polyanskiy, H.V. Poor, and S. Verdú, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307-2359, May 2010.
- [6] T.J. Richardson and R.L. Urbanke, *Modern coding theory*, Cambridge University Press, 2008.
- [7] G. Liva, L. Gaudio, T. Ninacs, T. Jerkovits, "Code Design for Short Blocks: A Survey," arXiv:1610.00873 [cs.IT], Oct. 2016.
- [8] J. Van Wouwerghem, A. Alloum, J.J. Boutros, M. Moeneclaey, "Performance comparison of short-length error-correcting codes," *Communications and Vehicular Technologies (SCVT), 2016 Symposium on*, pp.1-6, Oct 2016.
- [9] J. Van Wouwerghem, A. Alloum, J.J. Boutros, M. Moeneclaey, "On Performance and Complexity of OSD for Short Error Correcting Codes in 5G-NR," *First International Balkan Conference on Communications and Networking, Tirana, 2017 Symposium on*, June 2017.
- [10] B. Dorsch, "A decoding algorithm for binary block codes and J-ary output channels," *IEEE Trans. Inf. Theory*, vol. 20, pp. 391-394, May 1974.
- [11] J. Fang, "Décodage pondéré optimal des codes linéaires en blocs et quelques sujets sur la complexité du décodage", *PhD Thesis*, ENST, March 1987
- [12] M. P. C. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379-1396, Sep. 1995.
- [13] M. Fossorier, "Reliability-based soft-decision decoding with iterative information set reduction," *IEEE Trans. Inf. Theory*, vol. 48, pp. 3101-3106, Dec. 2002.
- [14] A. Valembois and M. P. C. Fossorier, "Box and match techniques applied to soft-decision decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 796-810, Dec. 2004.
- [15] W. Jin and M. P. C. Fossorier, "Probabilistic Sufficient Conditions on Optimality for Reliability Based Decoding of Linear Block Codes," *IEEE International Symposium on Information Theory*, Seattle, WA, pp. 2235-2239, 2006.
- [16] Y. Wu and C.N. Hadjicostis, "Soft-Decision Decoding of Linear Block Codes Using Preprocessing and Diversification," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 378-393, Jan. 2007.
- [17] Y. Wu and C.N. Hadjicostis, "Soft-Decision Decoding Using Ordered Recodings on the Most Reliable Basis," *IEEE Trans. Inf. Theory*, vol. 53, no. 2, pp. 829-836, Feb. 2007.
- [18] W. Jin and M. P. C. Fossorier, "Reliability-Based Soft-Decision Decoding With Multiple Biases," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 105-120, Jan. 2007.
- [19] Y. Wu and M. P. C. Fossorier, "Soft-decision decoding using time and memory diversification," *IEEE International Symposium on Information Theory*, Toronto, ON, 2008, pp. 76-80, 2008.
- [20] J. Van Wouwerghem, J.J. Boutros, and M. Moeneclaey, "On Constructions of Reed-Muller Subcodes," *IEEE Comm. Letters*, vol. 22, no. 2, pp. 220-223, Feb. 2018.
- [21] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. on Comm.*, vol. 44, pp. 1261-1271, Oct. 1996.
- [22] E. Arıkan, "Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.
- [23] R. Mori and T. Tanaka, "Performance of polar codes with the construction using density evolution," *IEEE Comm. Letters*, vol. 13, pp. 519-521, July 2009.
- [24] S. Dolinar and D. Divsalar, "Weight distribution for turbo codes using random and nonrandom permutations", *TDA Rep. 42-122*, Aug. 1995.
- [25] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE International Symposium on Information Theory*, St. Petersburg, Russia, pp. 1-5, 2011.
- [26] T. Erseghe, "Coding in the Finite-Blocklength Regime: Bounds Based on Laplace Integrals and Their Asymptotic Approximations," *IEEE Trans. on Inf. Theory*, vol. 62, no. 12, pp. 6854-6883, Dec. 2016.
- [27] J. G. Proakis, *Digital Communications*, 4th ed., New York: McGraw-Hill, 2000.
- [28] E. Biglieri, *Coding for Wireless Channels*, New York: Springer, 2005.
- [29] J. J. Boutros, A. Guillen i Fabregas, E. Biglieri, and G. Zemor, "Low-Density Parity-Check Codes for Nonergodic Block-Fading Channels," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4286-4300, Sept. 2010.
- [30] S. Sesia, M. Baker, I. Toufik, "LTE-the UMTS long term evolution: from theory to practice," *John Wiley & Sons*, 2011.
- [31] Nokia, ALSB, "R1-1703107 and 1703774, Short block codes for eMBB control channels," GPP TSG-RAN WG1 Meeting #88 and #88b.
- [32] Samsung, "R1-1705427, Channel coding for very short length control information," 3GPP TSG-RAN WG1 Meeting #88b.
- [33] Qualcomm, "R1-1705636, Evaluation of the coding schemes for very small blocklength," 3GPP TSG-RAN WG1 Meeting #88b.
- [34] Huawei, HiSilicon, "R1-1704248, On channel coding for very small control block lengths," 3GPP TSG-RAN WG1 Meeting #88b.
- [35] Wang, Tao, Daiming Qu, and Tao Jiang. "Parity-Check-Concatenated Polar Codes." *IEEE Communications Letters* pp.2342-2345, Dec 2016.
- [36] A.E. Brouwer and T. Verhoeff, "An updated table of minimum-distance bounds for binary linear codes," *IEEE Trans. on Inf. Theory*, vol. 39, no. 2, pp. 662-677, Mar. 1993.