



**HAL**  
open science

# Enhanced Turbo Codes for NR: Performance Evaluation for eMBB and URLLC

Charbel Abdel Nour

► **To cite this version:**

Charbel Abdel Nour. Enhanced Turbo Codes for NR: Performance Evaluation for eMBB and URLLC: R1-1613029. [Technical Report] 3GPP TSG-RAN WG1 Meeting #87. 2016. hal-02976894

**HAL Id: hal-02976894**

**<https://hal.science/hal-02976894>**

Submitted on 23 Oct 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Agenda item: 7.1.5.1

Source: Orange and Institut Mines-Telecom

Title: Enhanced Turbo Codes for NR: Performance Evaluation for eMBB and URLLC

Document for: Discussion

---

## 1 Introduction

This document describes the error rate performance of the enhanced Turbo Coding scheme proposed in R1-164635 at RAN1#85 meeting and detailed in R1-167413.

The simulation conditions of the decoding process are the following:

- Max-Log-MAP decoding of codes C1 and C2 with application of scaling factors to extrinsics: 0.6 for the first decoding iteration, 1.0 for the last decoding iteration, 0.7 for the other iterations; a short training is used for having an estimate of forward and backward recursions at trellis edges only at first iteration.
- Floating-point representation of the LLR (Log-Likelihood Ratio) values;
- 8 decoding iterations (1 iteration = decoding of code C1 + decoding of code C2);
- AWGN and fast fading Rayleigh transmission channel;
- QPSK, 16QAM and 64QAM modulations;
- The lowest points in the curves were obtained with at least 50 erroneous blocks.

## 2 Interleaver and puncturing parameters

The proposed enhanced Turbo code represents a rate compatible design where the same interleaver parameters are kept for all the considered coding rates. Therefore, only the puncturing is varied from one coding rate to the other. In addition, the puncturing is incremental with the coding rate. In other words, the puncturing for a higher coding rate is the same as the one with the lower rate with an additional punctured position. Therefore, incremental redundancy is natively supported between all proposed puncturing schemes. In addition, the extension to one-bit codeword granularity can be easily supported with a guarantee on stable performance.

### 2.1 Almost Regular Permutation (interleaver design)

The interleaving function is detailed in R1-167413 and given by the following equation:

$$\pi(i) = (Pi + S(i \bmod Q)) \bmod K \quad (1.1)$$

where  $i$  denotes the address of the data symbol after interleaving and  $\pi(i)$  represents its corresponding address before interleaving.

## 2.2 eMBB scenario

### 2.2.1 Puncturing masks

Considered constituent code RSC(1, 15/13)<sub>8</sub>, Feedback (FB) polynomial = 13 and Feedforward (FF) polynomial = 15. **For rate  $R=1/3$  there is no puncturing.**

$R$	Parity puncturing mask for systematic = 111111111111111
8/23	1101111111111111
4/11	1101111111101111
8/21	1101111101101111
2/5	1101011101101111
8/19	1101011101101110
4/9	1101011101001110
8/17	1101011100001110
1/2	1101011100001010
8/15	1101010100001010
4/7	1101010000001010
8/13	1101010000000010
2/3	1100010000000010
8/11	1100000000000010
4/5	0100000000000010
8/9	0100000000000000

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.2.2 Puncturing mask $R=1/5$

Two possibilities can be used:

**First possibility:** constituent code RSC(1, 15/13, 17/13)<sub>8</sub> there is no puncturing.

**Second possibility:** constituent code RSC(1, 15/13, 17/13, 11/13)<sub>8</sub> the puncturing mask is:

<b>Systematic</b>	0000000000000000
<b>Parity (FF=15)</b>	1111111111111111
<b>Parity (FF=17)</b>	1111111111111111
<b>Parity (FF=11)</b>	1010101010101010

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

The second possibility exhibits better performance by around 0.4 dB at  $10^{-4}$  of BLER.

### 2.2.3 ARP interleaver parameters for $K=96$ bits

ARP interleaver parameters ( $K=96$ )		
$Q$	$P$	$(S(0)...S(Q-1))$
16	37	(8, 26, 59, 72, 66, 57, 38, 32, 72, 79, 60, 38, 13, 10, 52, 54)

### 2.2.4 ARP interleaver parameters for $K=208$ bits

ARP interleaver parameters ( $K=208$ )		
$Q$	$P$	$(S(0)...S(Q-1))$
16	147	(8, 156, 31, 174, 10, 115, 98, 62, 152, 97, 16, 156, 37, 84, 112, 68)

### 2.2.5 ARP interleaver parameters for $K=400$ bits

ARP interleaver parameters ( $K=400$ )		
$Q$	$P$	$(S(0)...S(Q-1))$
16	383	(8, 80, 311, 394, 58, 55, 250, 298, 56, 197, 280, 40, 229, 40, 136, 192)

### 2.2.6 ARP interleaver parameters for $K=992$ bits

ARP interleaver parameters ( $K=992$ )		
$Q$	$P$	$(S(0)...S(Q-1))$
16	729	(8, 390, 691, 60, 514, 53, 110, 244, 696, 427, 596, 10, 797, 742, 364, 410)

The same ARP parameters are used for all coding rates of a particular frame size  $K$ . Therefore, incrementally moving from any high coding rate to any lower rate is possible by transmitting only corresponding parity bits from the puncturing mask. These puncturing masks apply for the 4 different frame sizes.

## 2.3 URLLC and control channel scenarios

The interleaver parameters designed for block sizes of eMBB ( $K=96$ ,  $K=208$ ,  $K=400$  and  $K=992$  bits) can be reused for URLLC and control channel scenarios down to coding rate  $R = 1/5$  with puncturing masks provided for eMBB. Then, for coding rates  $R < 1/5$ , puncturing masks and polynomials used for URLLC and control channel provided next are applied.

### 2.3.1 Puncturing masks for Rates $R \geq 1/3$

These masks apply only for sizes specific for URLLC ( $K=32$  and  $K=80$  bits). For  $R=1/3$  the considered constituent code is  $RSC(1, 15/13)_8$  there is no puncturing.

$R$	Parity puncturing mask for systematic = 1111
1/2	1010
2/3	1000

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.3.2 Puncturing mask $R=1/6$

Considered constituent code RSC(1, 15/13, 17/13, 11/13)<sub>8</sub> the puncturing mask is:

<b>Systematic</b>	0000
<b>Parity (FF=15)</b>	1111
<b>Parity (FF=17)</b>	1111
<b>Parity (FF=11)</b>	1111

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.3.3 Puncturing mask $R=1/8$

Considered constituent code RSC(1, 15/13, 17/13, 11/13, 16/13)<sub>8</sub> the puncturing mask is:

<b>Systematic</b>	0000
<b>Parity (FF=15)</b>	1111
<b>Parity (FF=17)</b>	1111
<b>Parity (FF=11)</b>	1111
<b>Parity (FF=16)</b>	1111

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.3.4 Puncturing mask $R=1/10$

Considered constituent code RSC(1, 15/13, 17/13, 11/13, 16/13, 12/13)<sub>8</sub> the puncturing mask is:

<b>Systematic</b>	0000
<b>Parity (FF=15)</b>	1111
<b>Parity (FF=17)</b>	1111
<b>Parity (FF=11)</b>	1111
<b>Parity (FF=16)</b>	1111
<b>Parity (FF=12)</b>	1111

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.3.5 Puncturing mask $R=1/12$

Considered constituent code RSC(1, 15/13, 17/13, 11/13, 16/13, 12/13, 17/13)<sub>8</sub> the puncturing mask is:

<b>Systematic</b>	0000
<b>Parity (FF=15)</b>	1111
<b>Parity (FF=17)</b>	1111
<b>Parity (FF=11)</b>	1111
<b>Parity (FF=16)</b>	1111
<b>Parity (FF=12)</b>	1111
<b>Parity (FF=17)</b>	1111

Note : "1" means "non-punctured (transmitted) symbol" and "0" means "punctured (non-transmitted) symbol".

### 2.3.6 ARP interleaver parameters for $K=32$ bits

<b>ARP interleaver parameters (<math>K=32</math>)</b>		
$Q$	$P$	$(S(0)...S(Q-1))$
4	9	(3, 5, 3, 5)

### 2.3.7 ARP interleaver parameters for $K=48$ bits

<b>ARP interleaver parameters (<math>K=48</math>)</b>		
$Q$	$P$	$(S(0)...S(Q-1))$
4	11	(3, 27, 47, 23)

### 2.3.8 ARP interleaver parameters for $K=80$ bits

<b>ARP interleaver parameters (<math>K=80</math>)</b>		
$Q$	$P$	$(S(0)...S(Q-1))$
4	11	(3, 63, 3, 63)

The same ARP parameters are used for coding rates  $2/3$ ,  $1/2$ ,  $1/3$ ,  $1/6$ ,  $1/8$ ,  $1/10$  and  $1/12$  of a particular frame size  $K$ .

## 3 Performance results

The performance results presented in section 3 were obtained using the puncturing patterns and the interleaving parameters described in the previous sections. We start by QPSK constellation. Then 16QAM and 64QAM simulations are provided followed by some results over Rayleigh fading channel.

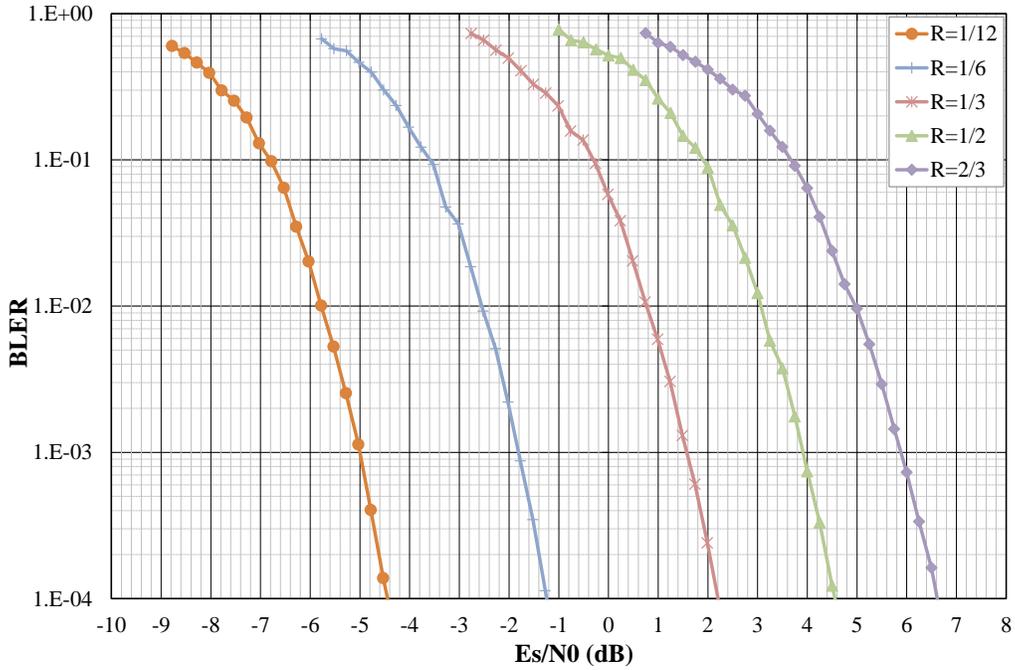


Figure 1: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 32$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC and control channel scenarios.**

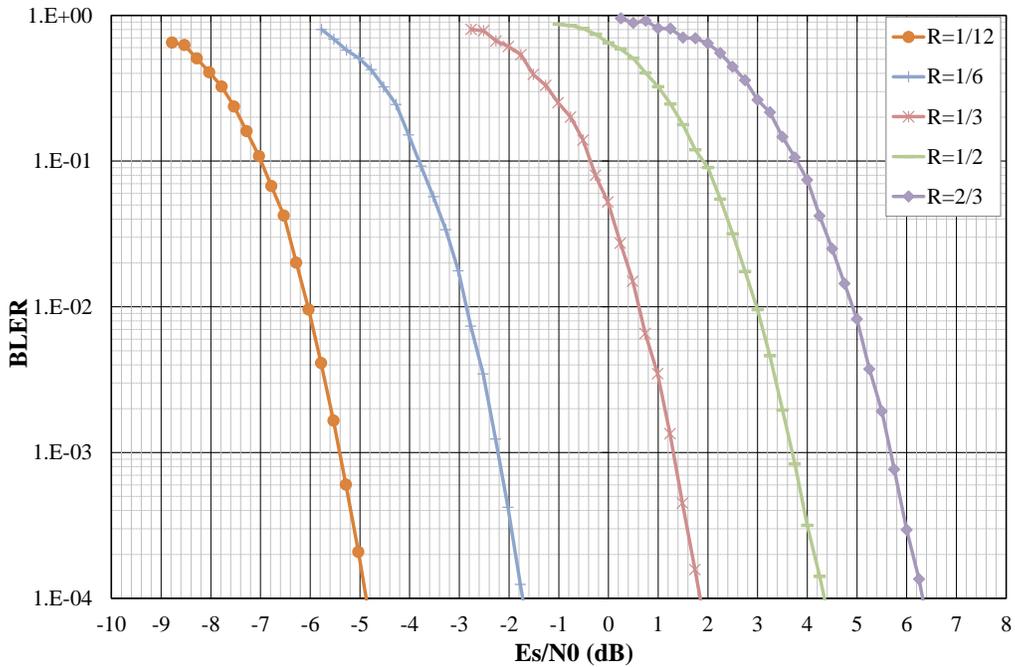


Figure 2: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 48$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC and control channel scenarios.**

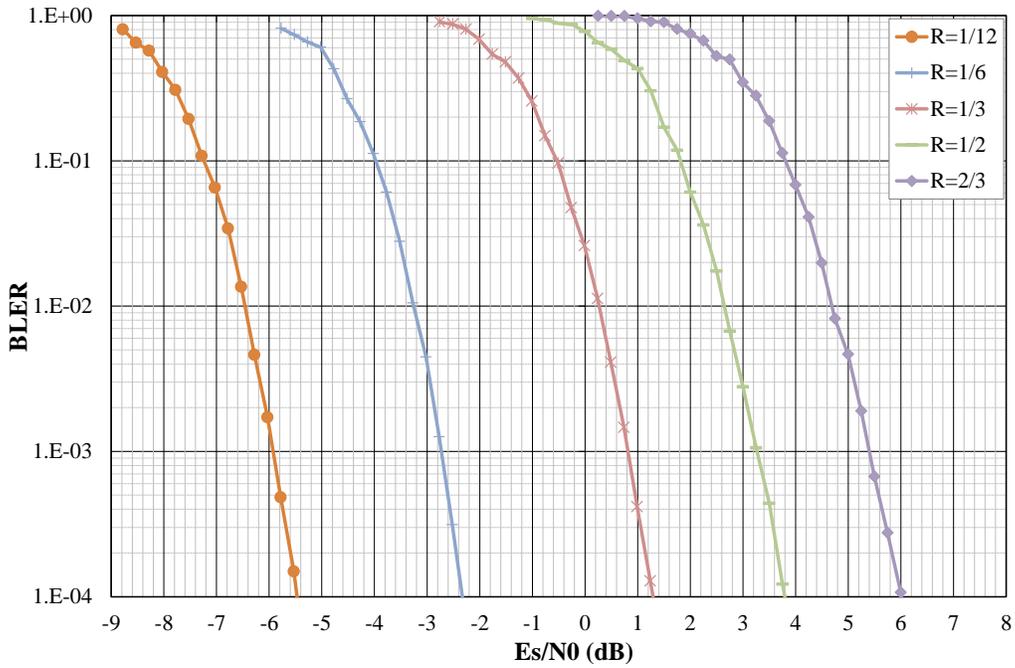


Figure 3: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 80$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC and control channel scenarios**.

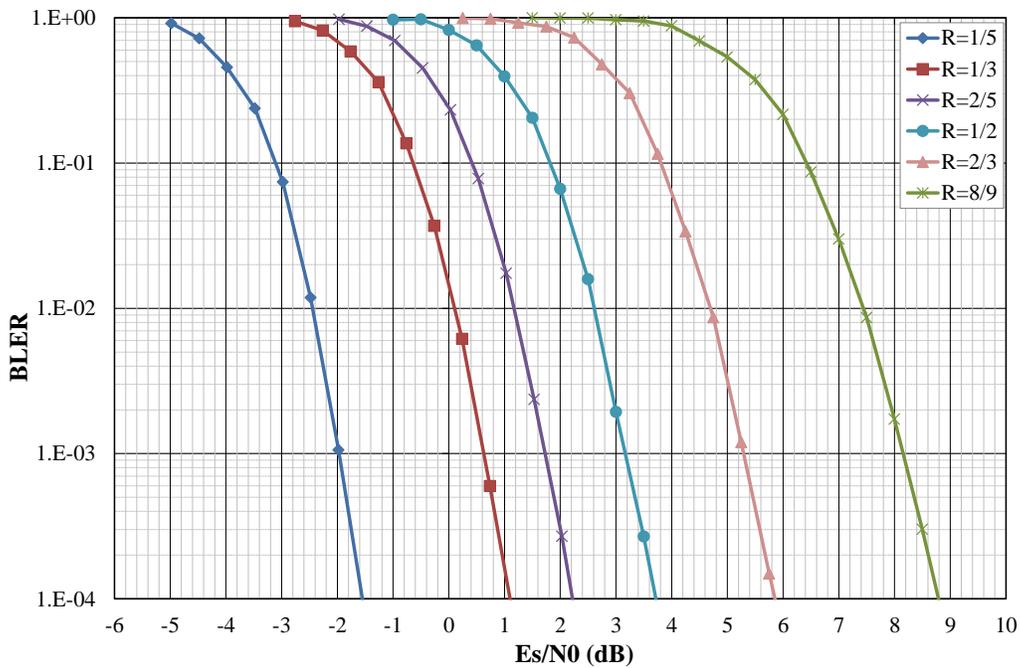


Figure 4: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 96$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB Scenario**.

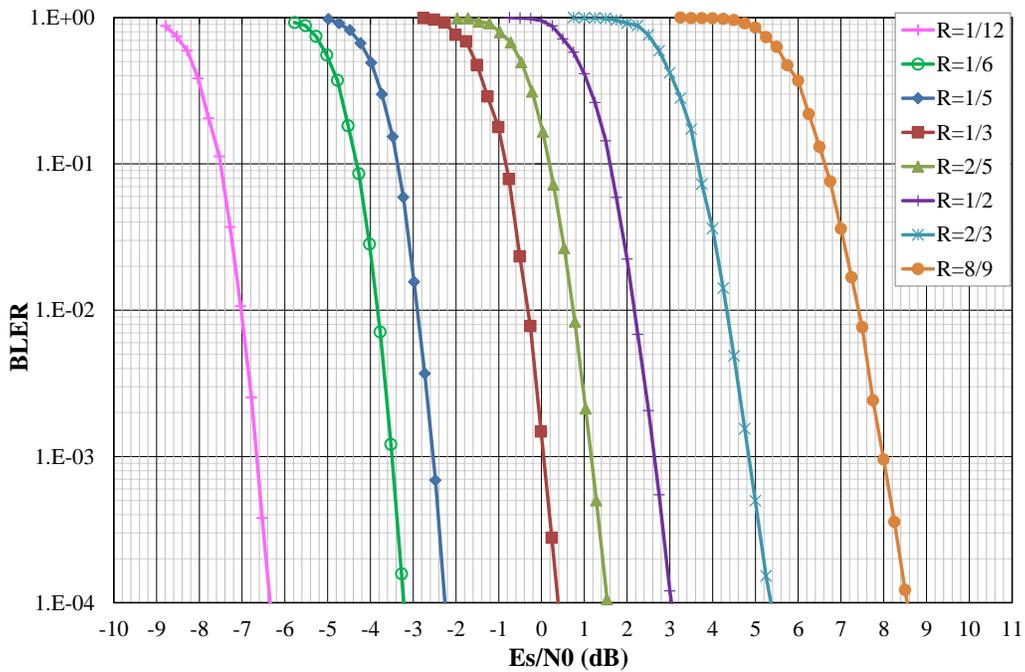


Figure 5: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 208$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB, URLLC and control channel Scenarios.**

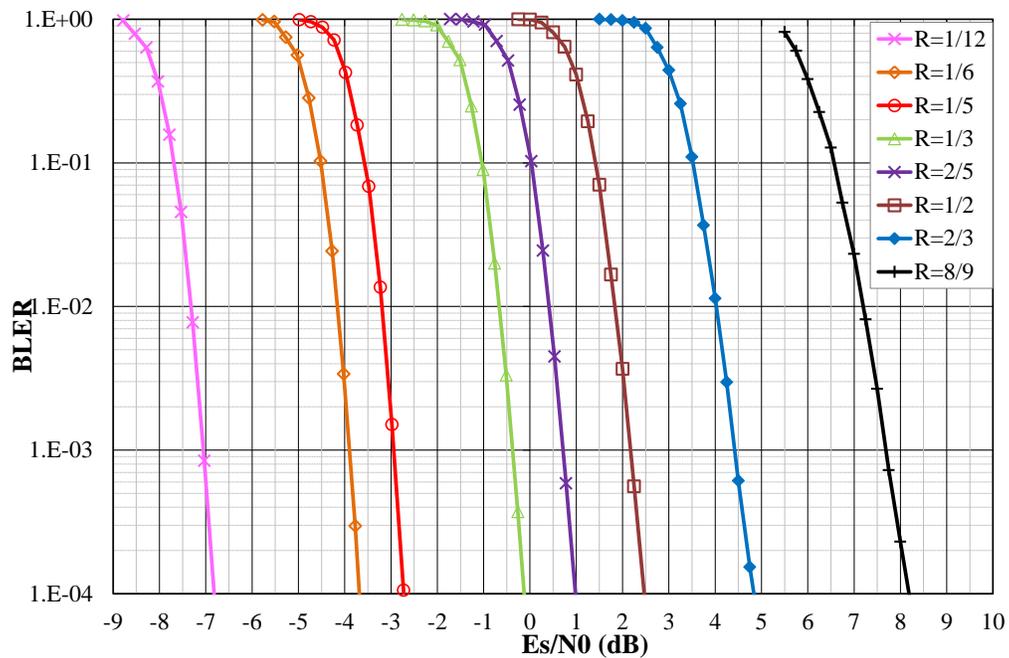


Figure 6: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 400$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB, URLLC and control channel Scenarios.**

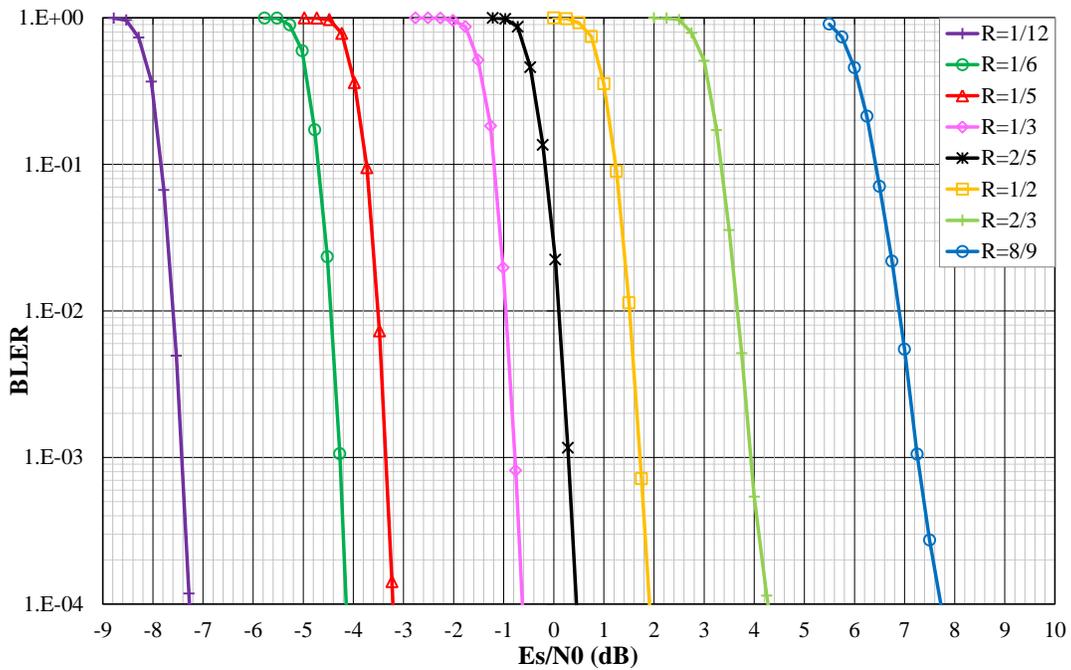


Figure 7: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 992$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB, URLLC and control channel Scenarios.**

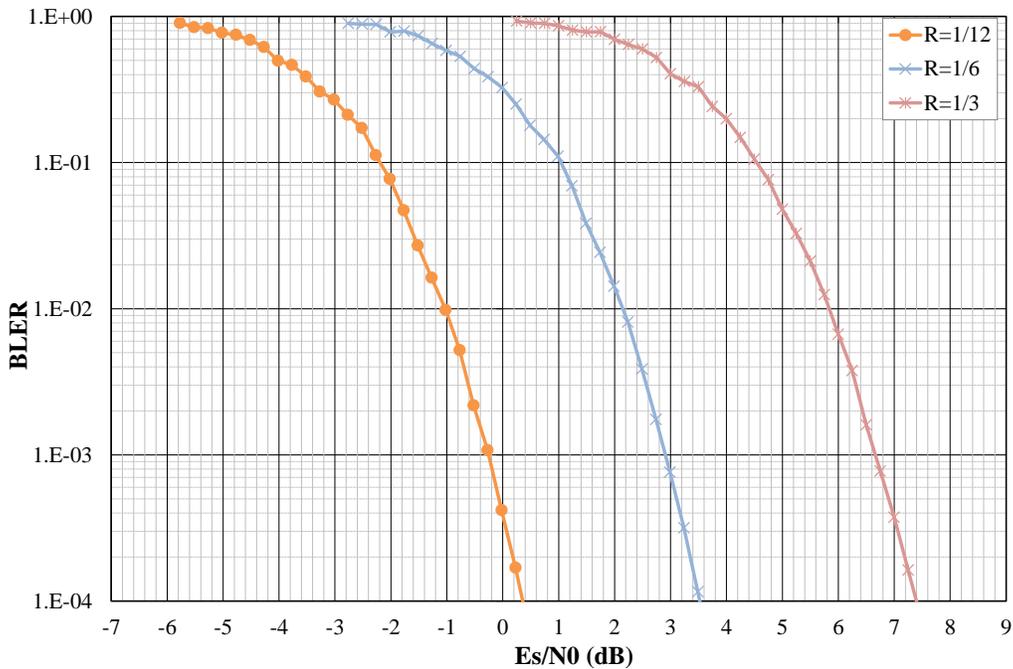


Figure 8: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **16QAM modulation**, information block size  $K = 32$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC scenario.**

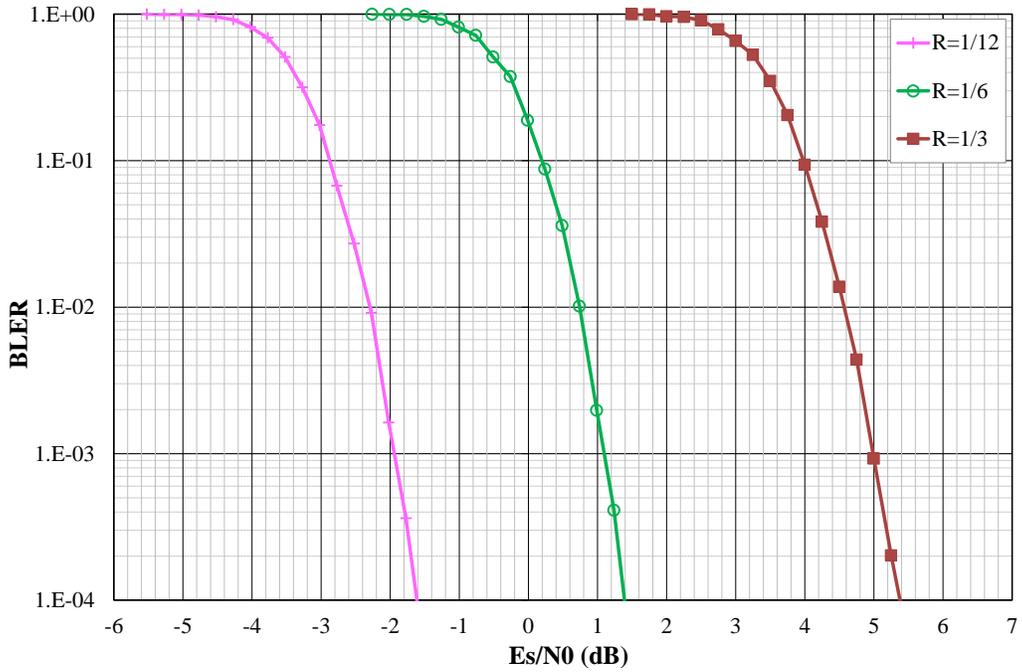


Figure 9: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **16QAM modulation**, information block size  $K = 208$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC scenario**.

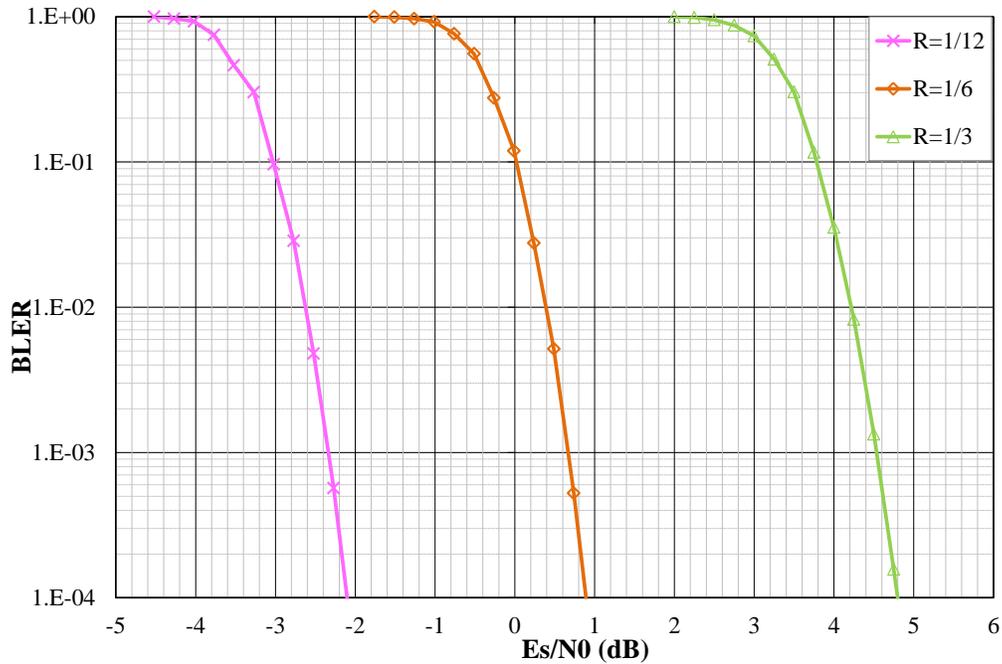


Figure 10: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **16QAM modulation**, information block size  $K = 400$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC scenario**.

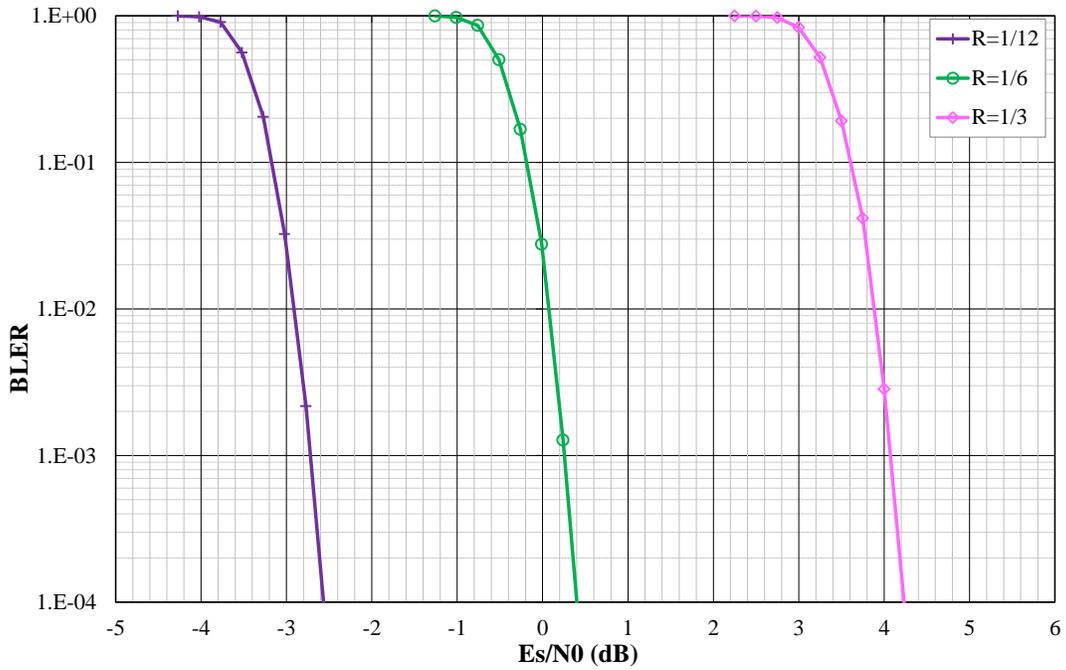


Figure 11: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **16QAM modulation**, information block size  $K = 992$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **URLLC scenario**.

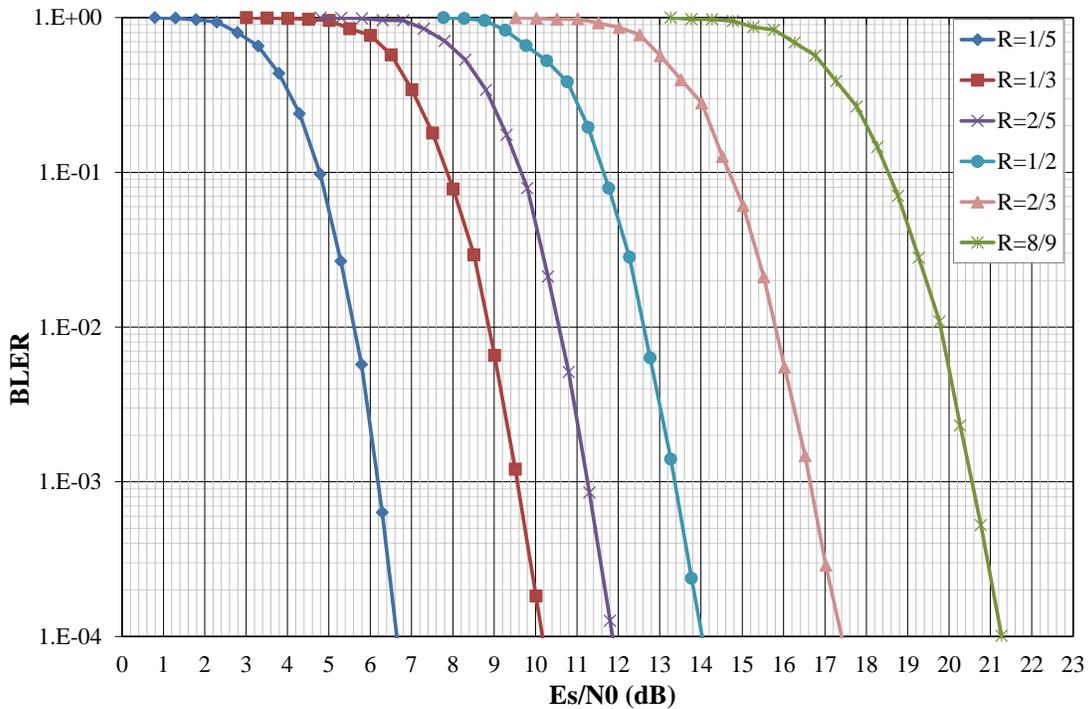


Figure 12: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . **64QAM modulation**, information block size  $K = 96$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB scenario**.

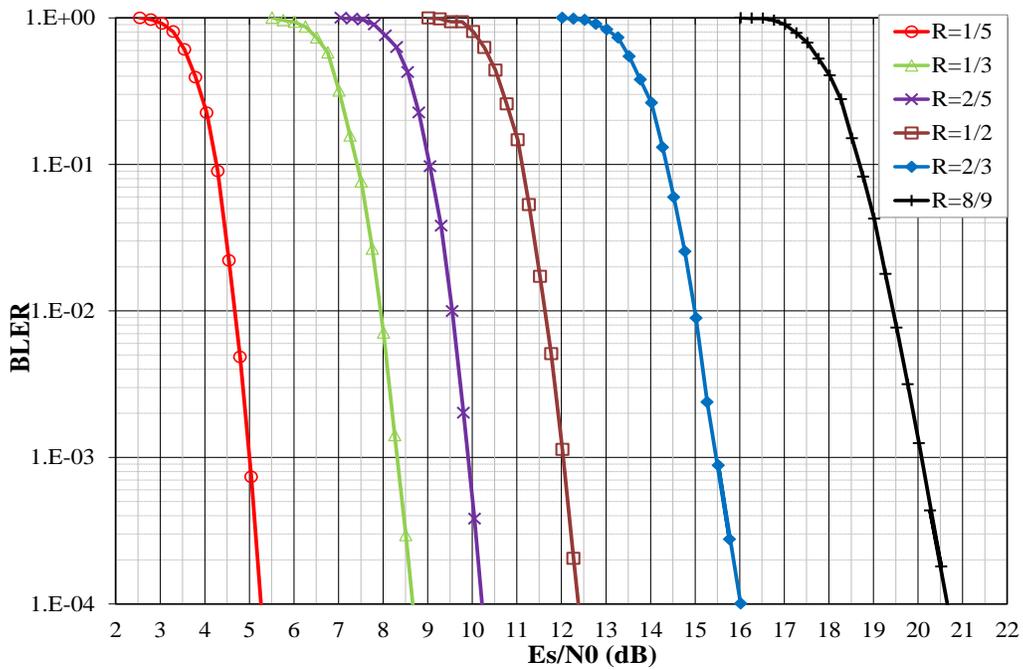


Figure 13: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **64QAM modulation**, information block size  $K = 400$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB scenario**.

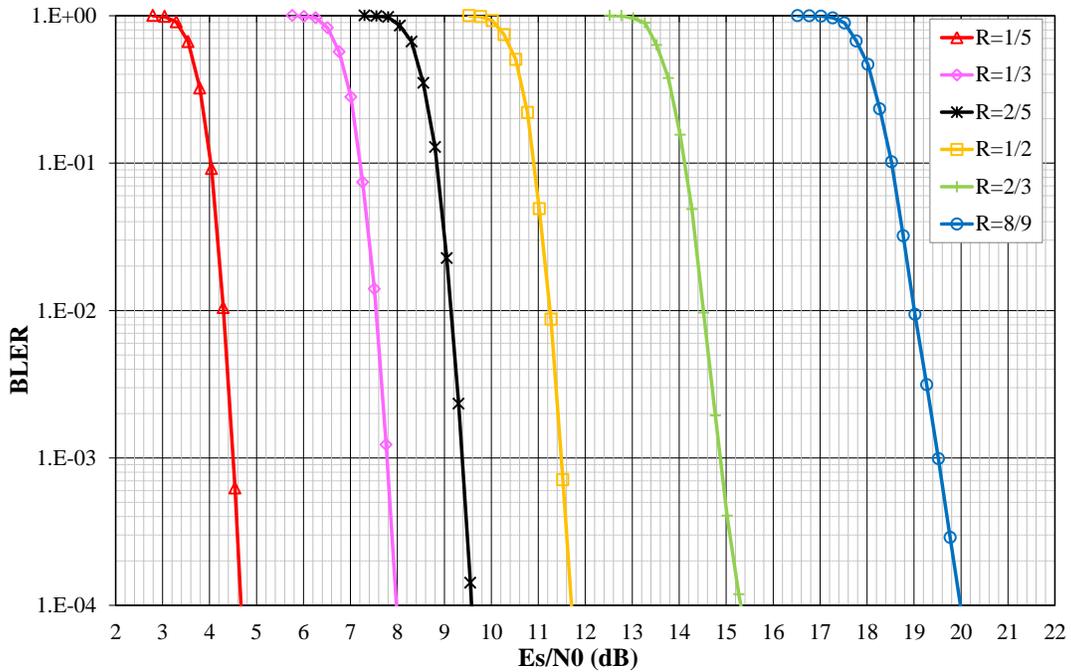


Figure 14: Performance evaluation of the enhanced turbo code in AWGN channel in terms of Block Error Rate vs Es/N0. **64QAM modulation**, information block size  $K = 992$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination. **eMBB scenario**.

### Rayleigh fading channel

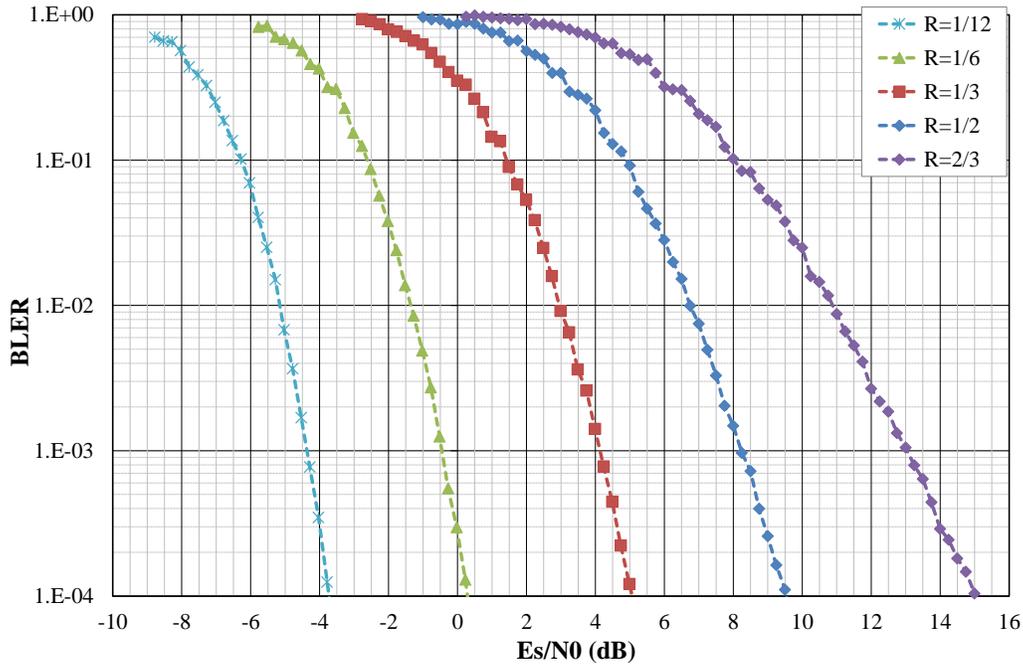


Figure 15: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 32$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

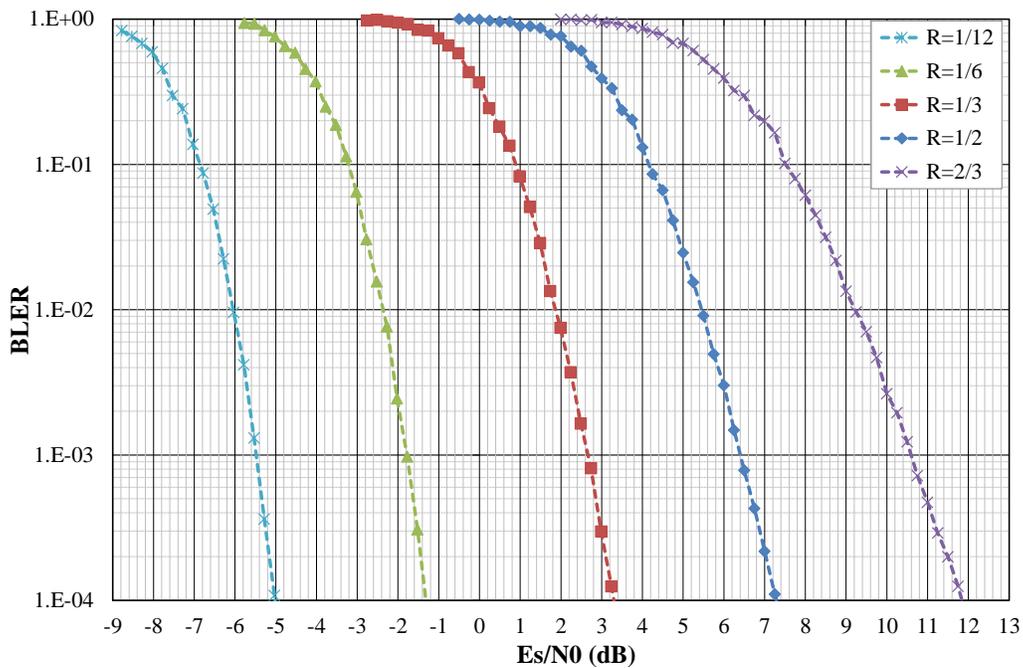


Figure 16: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs Es/N0. **QPSK modulation**, information block size  $K = 80$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

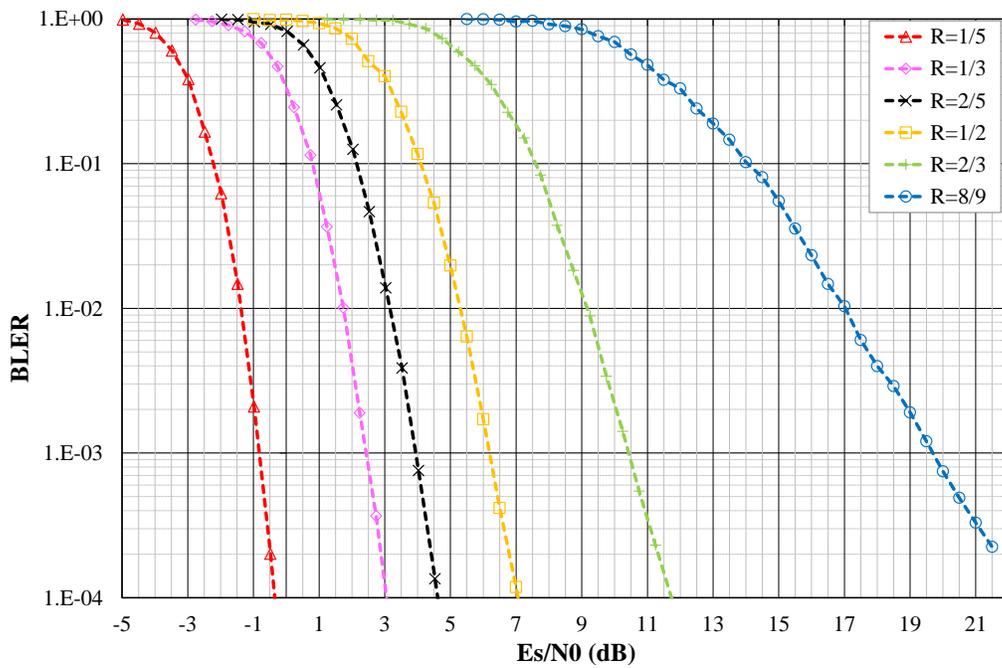


Figure 17: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 96$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

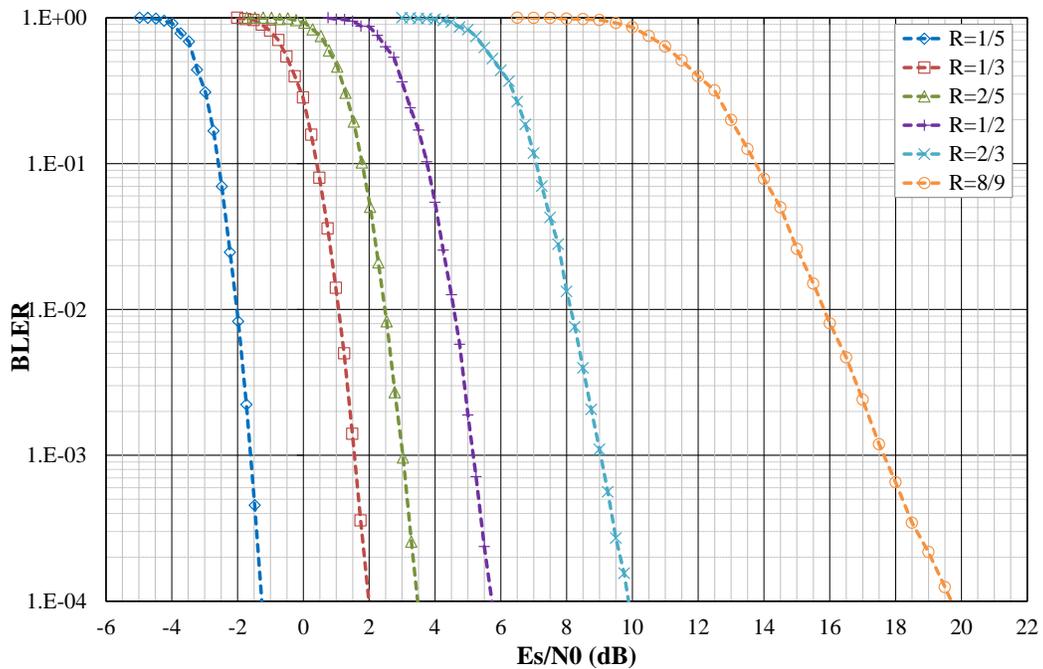


Figure 18: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 208$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

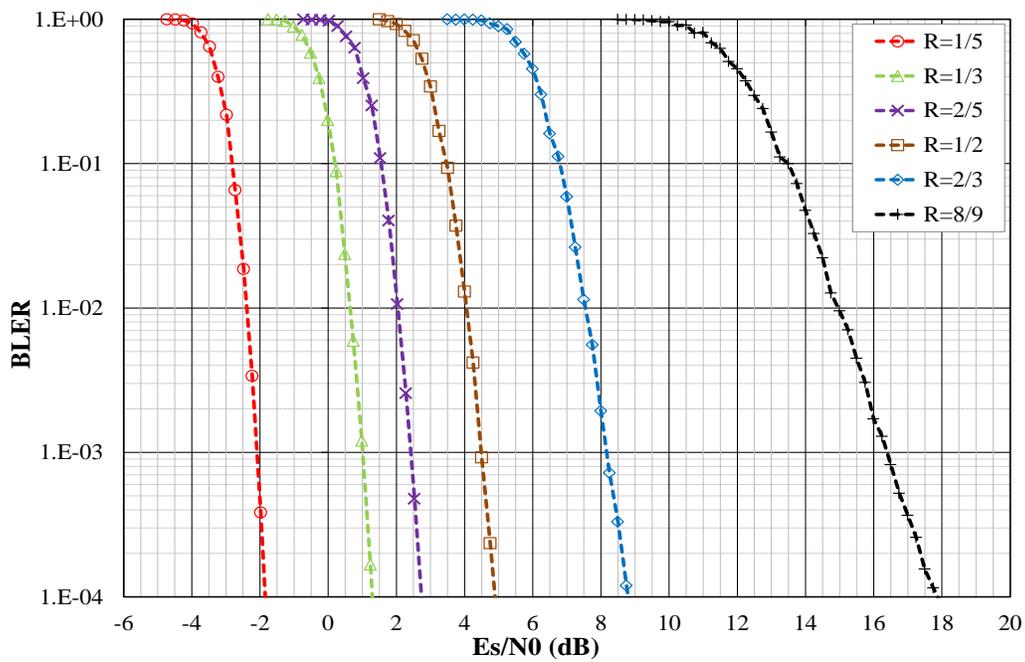


Figure 19: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 400$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

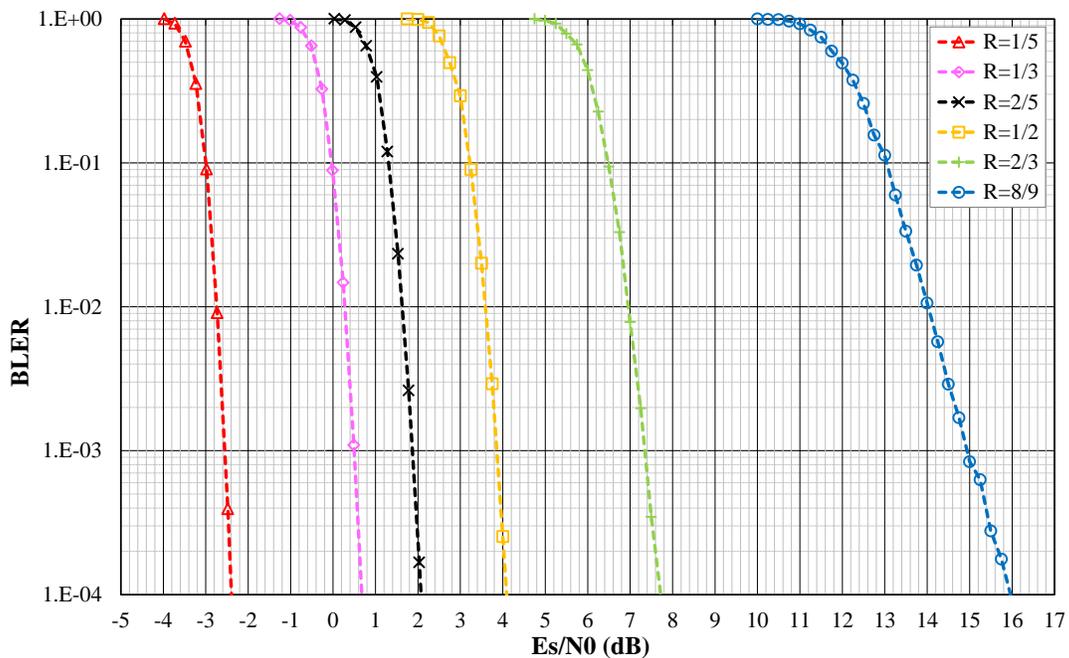


Figure 20: Performance evaluation of the enhanced turbo code in **Rayleigh fading channel** in terms of Block Error Rate vs  $E_s/N_0$ . **QPSK modulation**, information block size  $K = 992$  bits, 8 decoding iterations of the Max-log-MAP algorithm, tail-biting termination.

## 4 List-like decoding

List decoding can also be performed on Turbo codes. Large improvements can be achieved, greatly approaching Maximum Likelihood (ML) performance. As provided in [1], Polar codes benefit largely from list32 decoding for short frame sizes. This can be clearly seen in Figure 21.

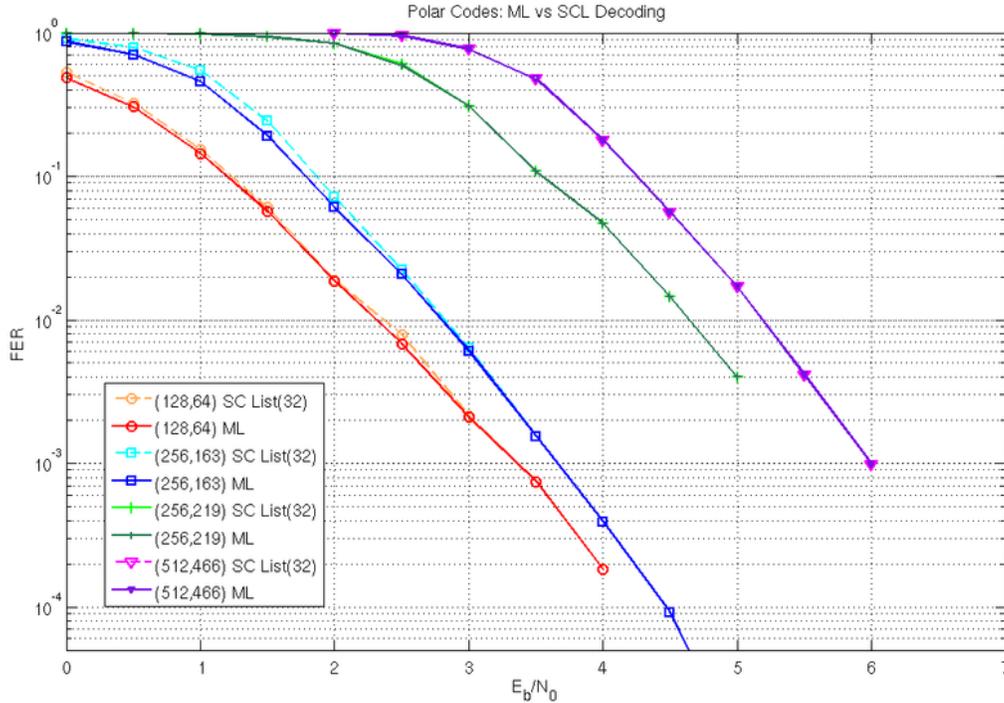


Figure 21: Performance comparison of a polar code with list32 decoding and ML decoding over AWGN channel in terms of Block Error Rate vs  $E_b/N_0$ . QPSK modulation, for several information block sizes. Fig. courtesy of [1].

We can clearly see that list32 performance reaches quasi-ML performance for Polar codes. Similarly, performance evaluations were performed for ML performance for Turbo codes in [1]. Results indicate an additional gain from going from Log-Map decoding to ML ranging from 0.5 to 0.9 dB. An example of these results is provided in Figure 22. There could be clear additional gains when applying Quasi-ML decoding for Turbo codes.

Ordered Statistics Decoding with the help of an outer CRC can also be performed for turbo codes as proposed in [2]. Hardware implementation was also provided for short frame sizes. Gains of more than 2dB were achieved for such solutions indicating the potential of such schemes as shown in Figure 23.

Both of these decoding techniques were also mentioned in [3].

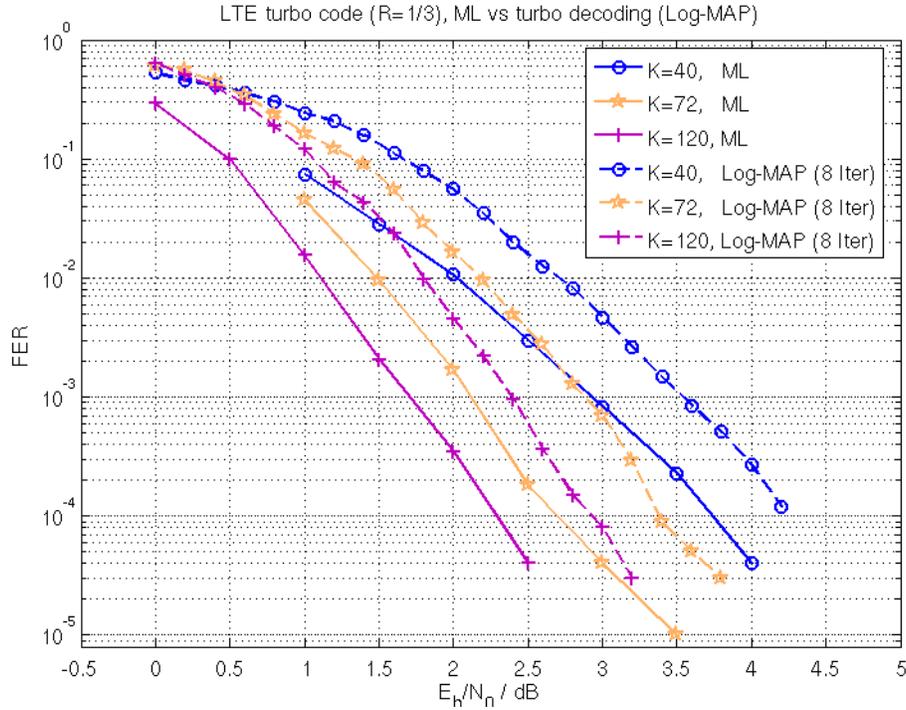


Figure 22: Performance comparison of the LTE turbo code with Log-Map decoding and ML decoding over AWGN channel in terms of Block Error Rate vs  $E_s/N_0$ . QPSK modulation, several information block sizes, 8 decoding iterations of the Log-MAP algorithm. Fig. courtesy of [1].

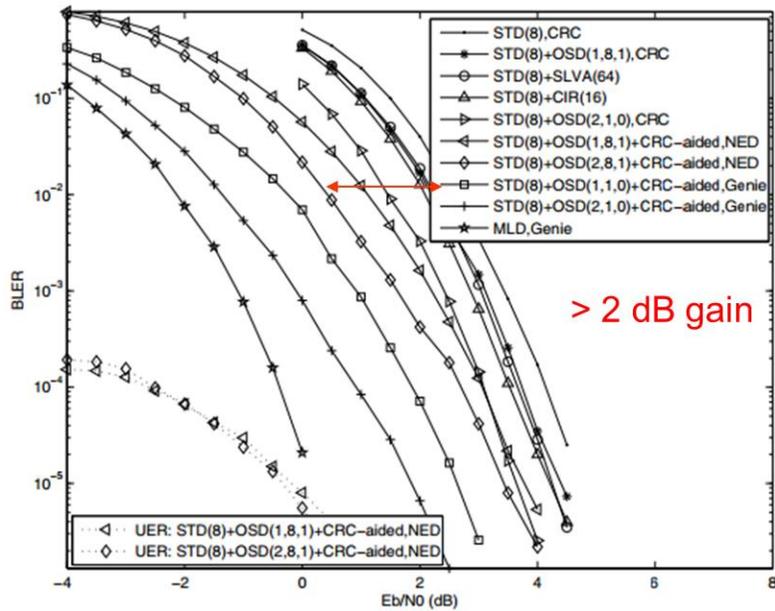


Figure 23: Performance comparison of the LTE turbo code with OSD and outer CRC over AWGN channel. Information frame size  $K=40$  bits. Fig. courtesy of [2].

Simplified or reduced complexity List decoding can also be performed for Turbo codes. Indeed, these techniques can be applied to component convolutional decoders at the price of a reduced

complexity since a large number of the performed computations can be shared with the classical Max-log MAP decoding. References and details are provided in Accelercomm's contribution [4] on the subject. Large gains can also be obtained in this case at the price of a reduced complexity.

## 5 Properties of the ARP interleaver

It has been shown in [5] that ARP interleavers have a degree of parallel processing  $pQ$  dividing  $K$ , where  $Q$  is the ARP period also dividing  $K$  and  $p$  any integer. In other words, ARP interleavers are contention free for any multiple of the ARP period. Later, it has been mentioned in [6] that "we believe many ARP interleavers (if not all) are Maximum Contention Free (MCF) and therefore ARP interleavers are stronger with respect to the degree of parallel processing than what is stated in [5]".

The proposed enhanced Turbo code follows a Rate-compatible version of the general guidelines for the ARP model and interleaver design detailed in [7]. This design ensures periodic connectivity inside the ARP period between a position in the natural order and another in the interleaved order among the possible  $Q$  positions. Thanks to this feature, this type of interleaving can be easily shown to be MCF.

## 6 Conclusion

**Observation 1:** Enhanced Turbo codes can greatly improve the performance of LTE turbo codes.

**Observation 2:** Enhanced Turbo codes can greatly improve the performance with respect to LDPC, especially for short frame sizes.

**Observation 3:** Quasi-ML or List-like decoding can improve the performance of Turbo codes up to 0.9dB on AWGN channels. Larger gains can be obtained over fast fading channels.

**Observation 4:** Ordered statistics decoding with Hybrid decoding of Turbo and CRC codes can provide gains of up to 2 dB for short frame sizes.

**Observation 5:** Several reduced complexity list decoding algorithms for turbo codes exist. They contribute to largely reducing the complexity while improving performance for short frame sizes.

**Observation 6:** Proposed ARP interleaver is maximum contention free eliminating memory access conflicts for all possible sub-block parallelism.

## 7 References

- [1] Helmling, Michael and Scholl, Stefan: *Database of Channel Codes and ML Simulation Results*. University of Kaiserslautern, 2016. URL: [www.uni-kl.de/channel-codes](http://www.uni-kl.de/channel-codes).
- [2] Y. Wei, M. Jiang, B. Xia, W. Chen and Y. Yang, "A CRC-Aided Hybrid Decoding Algorithm for Turbo Codes," in IEEE Wireless Communications Letters, vol. 2, no. 5, pp. 471-474, October 2013.
- [3] R1-1610931, "WF on channel coding observations for short block size", RAN1 meeting 86bis, October 2016, Lisbon, Portugal.
- [4] R1-1612308, "BLER performance of list decoding for enhanced turbo codes", RAN1 meeting 87, November 2016, Reno, USA.

- [5] C. Berrou, S. Kerouedan Y. Saouter, C. Douillard, and M. Jezequel, "Designing good permutations for turbo codes: towards a single model," in Proc. International Conference on Communications, Paris, France, June 2004, vol. 1, pp. 341-345.
- [6] O. Y. Takeshita, "On maximum contention-free interleavers and permutation polynomials over integer rings," in *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1249-1253, March 2006.
- [7] R. Garzón-Bohórquez, C. A. Nour and C. Douillard, "Improving Turbo Codes for 5G with parity puncture-constrained interleavers," *9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Brest, 2016, pp. 151-155.