On the Performance of Short Tail-Biting Convolutional Codes for Ultra-Reliable Communications

Lorenzo Gaudio, Tudor Ninacs, Thomas Jerkovits and Gianluigi Liva
Institute of Communications and Navigation of the German Aerospace Center (DLR),
Münchner Strasse 20, 82234 Weßling, Germany.
Email:{lorenzo.gaudio,tudor.ninacs,thomas.jerkovits,
gianluigi.liva}@dlr.de

***Abstract*—Motivated by the increasing interest in powerful short channel codes for low-latency ultra-reliable communications, we analyze the performance of tail-biting convolutional codes with different memories, block lengths and code rates over the additive white Gaussian noise channel. The analysis is carried out both through Monte Carlo simulations and by upper bounding the error probability via Poltyrev's tangential sphere bound at very low error rates. For the simulations, the near maximum likelihood wrap-around Viterbi algorithm is considered. We then compare the performance of tail-biting convolutional codes both with finite-length performance bounds and with that of other channel codes that have been recently considered for ultra-reliable satellite telecommand links. For the shortest block lengths, tail-biting convolutional codes outperform significantly state-of-the-art iterative coding schemes, while as expected their performance degrades visibly with increasing block lengths.**

***Keywords*—Convolutional codes, tail-biting, ultra-reliable communications, finite-length bounds**

I. INTRODUCTION

Ultra-reliable low-latency communications [1]–[3] are one of the underlying motivations for the rising interest in the design of efficient short and moderate-length channel codes. In fact, for long packet transmissions powerful (turbo-like) error correcting codes can be designed with near Shannon limit performance under iterative decoding [4]–[6]. Instead, in the moderate block length regime (i.e., the one of interest for ultra-reliable low-latency communications) a sizable loss (e.g. 1 dB or more) with respect to finite-length performance bounds has to be accounted for when binary turbo and low-density parity-check (LDPC) codes are used [7]. The loss can be largely reduced by opting for turbo and LDPC codes constructed over large-order finite fields [8]–[13], though at the cost of a non-negligible increase of decoding complexity.

It is well established that convolutional codes provide an excellent latency vs. performance trade-off (see e.g. [14]–[16]), especially when the bit error rate (BER) is used as a performance metric, thanks to windowed Viterbi decoding [17]. When short packets have to be transmitted, hence, terminated convolutional codes represent a promising candidate solution. Nevertheless, the rate loss that a zero tail termination introduces at short block lengths may be unacceptable. A tail-biting approach [18] would eliminate the rate loss and hence it deserve particular attention when comparing channel codes for short packets. For these reasons, tail-biting convolutional codes (TBCCs) are currently considered within the 5G standardization for ultra-reliable low-latency communications [19]–[21].

In this paper, we analyze the performance of short TBCCs over the binary input additive white Gaussian noise (bi-AWGN) channel and we compare it with theoretical benchmarks on the block error probability in the finite-length regime [22]–[24]. The analysis is carried out both via Monte Carlo simulations with the wrap-around Viterbi algorithm (WAVA) [25] and by upper bounding the block error probability under maximum-likelihood (ML) decoding via Poltyrev's tangential sphere bound (TSB) [26], [27]. Tail-biting encoders with different memories are considered from literature [28], [29], and the trade-offs between decoding complexity and the achievable performance is discussed. For each set of generator polynomials, the performance is evaluated with various packet lengths identifying the block length regimes for which the analyzed codes approach the theoretical limits.

The rest of the paper is organized as follows. Section II introduces preliminary definitions. In Section III the set of generator polynomials used in the comparisons is provided and the weight distribution of the analyzed codes is derived. Section IV provides an overview of the decoding algorithm used in the simulation setup. In Section V, the performance is analyzed in terms of codeword error rate (CER) vs. the channel signal-to-noise ratio (SNR). Conclusions follow in Section VI.

II. PRELIMINARIES

We denote by (n, k) the parameters of the code, with n being the block length and k the code dimension (in bits). At each clock, the encoder gets at its input k_b bits and produces at the output n_b bits. The encoder memory is denoted by m and it is expressed in bit k_b-tuples. Following the notation of

This work has been accepted for publication at the 11th International ITG Conference on Systems, Communications and Coding, SCC 2017

TABLE I
SUMMARY OF THE CODES SELECTED FOR THE ANALYSIS.

n_b	k_b	Generators	m	(n,k)	$A(X)$
2	1	$[515, 677]$	8	$(128, 64)$	$1 + 576X^{12} + 1152X^{13} + 1856X^{14} + 4800X^{15} + \ldots$
		$[5537, 6131]$	11		$1 + 64X^{14} + 960X^{15} + 1356X^{16} + 2304X^{17} + \ldots$
		$[75063, 56711]$	14		$1 + 8X^{16} + 1856X^{18} + 19392X^{20} + 342272X^{22} + \ldots$
		$[515, 677]$	8	$(256, 128)$	$1 + 1152X^{12} + 2304X^{13} + 3712X^{14} + 9600X^{15} + \ldots$
		$[5537, 6131]$	11		$1 + 128X^{14} + 1920X^{15} + 2688^{16} + 4608X^{17} + \ldots$
		$[75063, 56711]$	14		$1 + 3328X^{18} + 21120X^{20} + 108160X^{22} + 620032X^{24} + \ldots$
3	1	$[435, 526, 717]$	8	$(192, 64)$	$1 + 64X^{17} + 128X^{18} + 384X^{19} + 448X^{20} + \ldots$
		$[4653, 5435, 6257]$	11		$1 + 192X^{22} + 576X^{24} + 2048X^{26} + 4480X^{28} + \ldots$
		$[47671, 55245, 63217]$	14		$1 + 384X^{27} + 256X^{28} + 384X^{29} + 1344X^{30} + \ldots$
		$[435, 526, 717]$	8	$(384, 128)$	$1 + 128X^{17} + 256X^{18} + 768X^{19} + 896X^{20} + \ldots$
		$[4653, 5435, 6257]$	11		$1 + 384X^{22} + 1152X^{24} + 4096X^{26} + 8960X^{28} + \ldots$
		$[47671, 55245, 63217]$	14		$1 + 768X^{27} + 512X^{28} + 768X^{29} + 2688X^{30} + \ldots$

[29, Sec. 2.6] we denote by $\nu \leq k_b \cdot m$ the overall constraint length, with $\nu = m$ for $k_b = 1$. The code rate is given by $R = k/n = k_b/n_b$. The connections between the shift-registers, inputs and outputs are described with the aid of the generator polynomials of the code, which are provided in their octal notation. A TBCC can be represented by a tail-biting trellis with $\ell = k/k_b$ sections and

$$S = 2^\nu$$

states per section. The codewords correspond to all paths for which the initial and final states coincide. The distance spectrum of a TBCC is denoted by A_i, with A_i being the multiplicity of codewords with Hamming weight i, with $i = 0, \ldots, n$. In the remainder of the paper, the distribution of the weights is given in terms of the weight enumerator function (WEF)

$$A(X) = \sum_i A_i X^i.$$

We consider the transmission over a bi-AWGN with a non-systematic TBCC. We denote the information message as $\mathbf{u} = (u_0, \ldots, u_{k-1})$ and the TBCC word as $\mathbf{c} = (c_0, \ldots, c_{n-1})$. The codeword $\mathbf{c}$ is then binary phase shift keying (BPSK)-modulated. The modulated vector $\mathbf{x}$ is transmitted over an additive white Gaussian noise (AWGN) channel. At the receiver side, the vector $\mathbf{y}$ given by the observations $(y_0, \ldots, y_{n-1})$ is input to the channel decoder, which outputs a decision $\hat{\mathbf{u}}$ on the transmitted message. Throughout the paper, we will analyze the performance of various convolutional codes in terms of their block error probability $\mathrm{P_B} = \Pr\{\hat{\mathbf{u}} \neq \mathbf{u}\}$ as a function of the ratio E_b/N_0, being E_b the energy per information bit and N_0 the single-sided noise power spectral density. When Monte Carlo simulations are used, the estimated block error probability is denoted as CER.

III. TAIL-BITING CONVOLUTIONAL CODES: SELECTION

The TBCCs analyzed in the following sections have been selected from lists of generator polynomials leading to large minimum distance codes [28]–[30]. Due to the specific block lengths considered in this paper, for each generator polynomial set and block length pair we derived the distance spectrum of the resulting TBCC. The codes selected for the analysis are summarized in Table I. To derive the WEF of the codes, we took advantage of their trellis representation [31]. The trellis branches can be labeled with monomials in the form X^d, with the exponent d corresponding to the encoder output Hamming weight. Consider the $S \times S$ transition matrix $\boldsymbol{T}(X)$, where the entry $t_{i,j}$ corresponds to the label (i.e., monomial in X) of the trellis branch going from state i to state j. Since the TBCC words are associated to paths for which the first and last state coincide, we have that

$$A(X) = \mathrm{tr}\left[\boldsymbol{T}^\ell(X)\right].$$

The selected codes have code rates $1/3$ and $1/2$ and information lengths of 64 and 128 bits. For both rate-$1/3$ and rate-$1/2$ codes, k_b is equal to 1 and memories 8, 11 and 14 are considered. The memory-8 codes targets a low-complexity profile. The memory-11 codes, while still practical, may not allow high-speed decoding. Codes with memory-14 encoders have been provided for completeness, i.e., to explore the achievable performance gains in the short block length regime with large memory codes. Their decoding with Viterbi-like algorithms (see Sec. IV) may be considered impractical in current mobile wireless communication devices.

IV. WRAP-AROUND VITERBI ALGORTIHM

ML decoding over a tail-biting trellis can be performed by starting S Viterbi decoders in parallel, each operating a sub-trellis with equal initial/final state, for all the possible states. Each Viterbi decoder outputs the most likely codewords for the given initial (final) state. Among the S so-obtained codewords, a further ML search is performed, providing the final decision. It is easy to check that the complexity of this algorithm is proportional to S^2, thus decoding becomes quickly impractical for increasing memory sizes. In this paper, we adopt the sub-optimal WAVA introduced in [25]. The complexity of the WAVA is only proportional to the number of states S.

The WAVA is a circular decoding algorithm used to process tail-biting trellises iteratively. It runs the Viterbi algorithm successively for more iterations, improving the reliability of the decision at each iteration. The algorithm reaches near-to-optimal performance [25], i.e. the loss with respect to ML

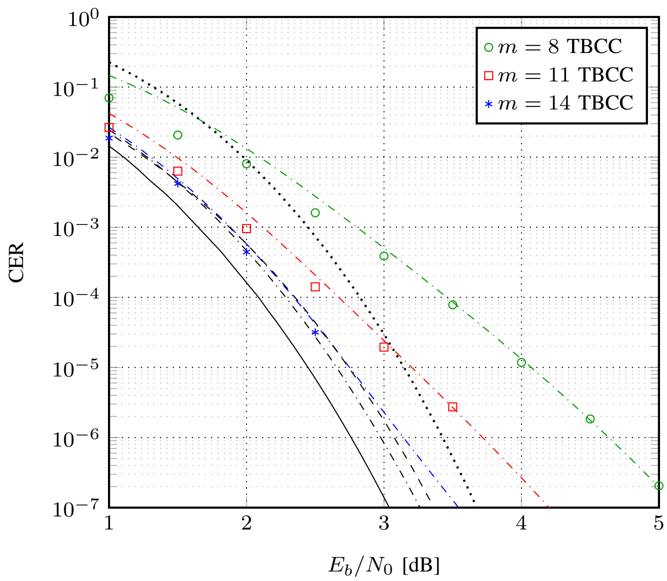

Fig. 1. Codeword error rate vs E_b/N_0 for $(192, 64)$ TBCCs. The markers denote simulation results, whereas the dashed lines provide the corresponding codes TSBs on the ML block error probability.

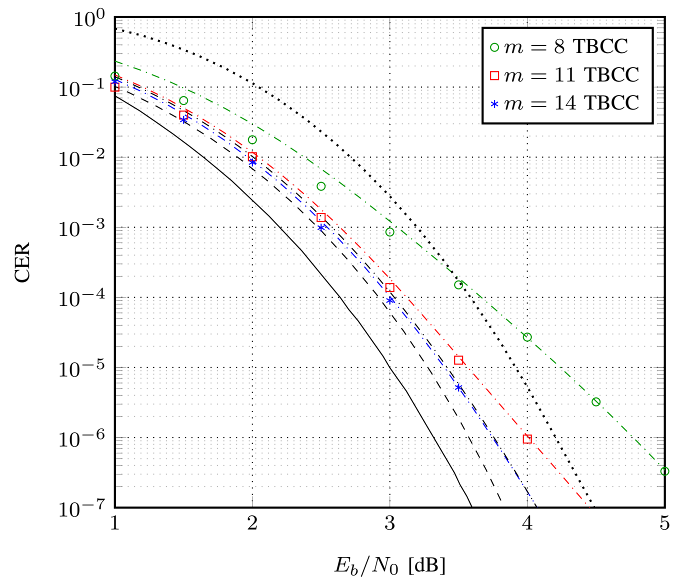

Fig. 2. Codeword error rate vs E_b/N_0 for $(128, 64)$ TBCCs. The markers denote simulation results, whereas the dashed lines provide the corresponding codes TSBs on the ML block error probability.

decoding is negligible, given that a large enough number of iterations is performed. The algorithm relies on the notion of tail-biting paths, i.e. paths whose initial and final state coincide.

The algorithm is very similar to the Viterbi algorithm with equally probable initial states. In its log domain implementation, at the beginning of the decoding process all initial state metrics are set to zero. A first iteration of the Viterbi algorithm is performed over the code trellis, producing for each of the S final states one path survivor. We assume next that the path metric is given by the squared Euclidean distance between the (modulated) branch labels and the corresponding channel observations. Among the survivors, the most likely one (i.e., the one at minimum Euclidean distance from the received vector) is selected. If this most likely path is also a tail-biting one, then decoding process stops and the most likely tail-biting path is output. If the most likely path is not tail-biting, then the initial state metrics of the trellis are updated with the metrics computed at the corresponding final states of the previous iteration. A new iteration of the Viterbi algorithm is performed. The process continues until either the termination condition is satisfied, i.e. the path with minimum final state metric is a tail-biting one, or a maximum number of iterations is reached.[1] In the former case, the algorithm outputs the path with minimum final state metric. In the latter case, the decoder outputs again the path corresponding to the path with minimum final state metric even if the path does not satisfy the tail-biting constraint.[2]

[1] It was shown in [25] that four iterations are typically sufficient to obtain near-ML performance.

[2] This choice may help in reducing the number of bit errors, while leading anyhow by default to a block error.

V. Performance over the AWGN Channel

In this section, Monte Carlo simulation results on the CER for the codes introduced in Section III are provided and compared with theoretical benchmarks. In particular, the results are compared with

i. Shannon's 1959 sphere packing bound (SPB) [22], providing a lower bound on the block error probability achievable by any code with block length n and 2^k codewords. The bound applies to the unconstrained input AWGN channel.
ii. Gallager's random coding bound (RCB) [23], which gives an upper bound to the average error probability of (n, k) random codes. The bound has been computed for the bi-AWGN channel.
iii. A tighter RCB, obtained by applying Poltyrev's TSB [26], [27] to the average weight enumerator of the $(n, m = n-k, 2)$ binary parity-check ensemble [32], given by

$$A_i = \begin{cases} 1 & i = 0 \\ \binom{n}{i} 2^{-(n-k)} & i > 0, i \leq n. \end{cases}$$

We refer to this bound as the tangential sphere bound for random codes (TSB-RC).
iv. The normal approximation [24] computed for the bi-AWGN channel.

The block error probability provided by ii. and iii. is achievable. Though, in the following, we shall see that turbo and LDPC codes manage only to approach the looser of the two random coding bounds, i.e., Gallager's RCB.

The numerical results are complemented by the evaluation of the TSB for the simulated codes, allowing an estimation on the block error probability at low P_B values. For each simulation point 100 block errors have been collected. The WAVA with a maximum of four iterations has been used,

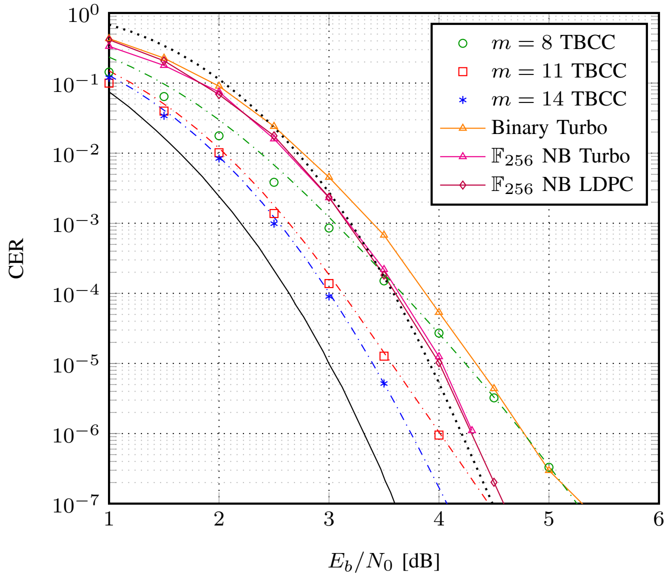

Fig. 3. Codeword error rate vs E_b/N_0 for $(128, 64)$ TBCCs. The markers denote simulation results, whereas the dashed lines provide the corresponding codes TSBs on the ML block error probability.

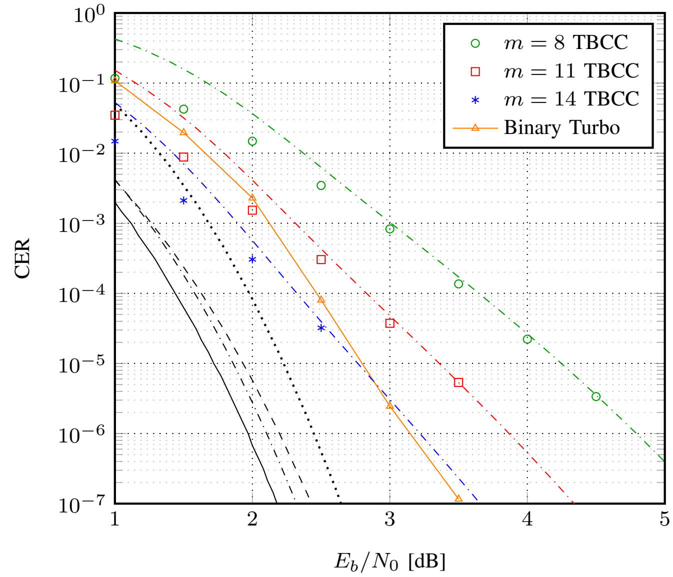

Fig. 4. Codeword error rate vs E_b/N_0 for $(384, 128)$ TBCCs. The markers denote simulation results, whereas the dashed lines provide the corresponding codes TSBs on the ML block error probability.

sufficient to obtain performance close to ML decoding [25]. In Figures 1 to 5, the sphere packing bound (——), the normal approximation (- - -), the tangential sphere bound for random codes (-·-·-) and Gallager's RCB (······) are provided as reference.

Results for the information length $k = 64$ bits and rates $R = 1/3$ and $R = 1/2$ are depicted in Figures 1 and 2. The rate $1/3$ case (Figure 1) shows how down to moderate-low error rates (i.e. CER$\approx 10^{-4}$) both memory-11 and memory-14 codes allow performing close to the TSB-RC, which for these code parameters nearly matches the normal approximation. The memory-8 code performs relatively close to Gallager's RCB down to a CER$\approx 10^{-3}$. Only the memory-14 code attains a performance close to the bounds down to low error rates (CER$\approx 10^{-7}$), whereas the memory-8 code loses almost 2 dB. The memory-11 TBCC provides a reasonable complexity-performance trade-off, attaining CER$\approx 10^{-7}$ within 1 dB from the normal approximation. The differences are less pronounced for the rate $1/2$ case (Figure 2). Here, both the memory-11 and the memory-14 attain a CER lower than that predicted by Gallager's RCB, down to CER$\approx 10^{-7}$, performing remarkably close to both the normal approximation and the TSB-RC.

In Figure 3, the performance of the rate-$1/2$ codes is compared with some binary and non-binary turbo and LDPC code designs. More specifically, the codes included in the comparison are

- a binary turbo code with 16-states tail biting component codes from [33];
- a non-binary turbo code constructed over a finite field of order 256 from [11];
- a protograph-based non-binary LDPC code constructed over a finite field of order 256 from [11], [13] and proposed for the recent upgrade of the Consultative Committee for Space Data Systems (CCSDS) satellite telecommand recommendation [34].

Despite its low decoding complexity, the memory-8 TBCC outperforms the 16-state turbo code down to CER$\approx 10^{-5}$, and tightly matches its performance down to CER$\approx 10^{-7}$. The memory-11 TBCC outperforms down to low error rate both the non-binary turbo and LDPC codes, with a comparable decoding complexity.

Figures 4 and 5 replicate the analysis of the $k = 64$ bits case for codes of dimension $k = 128$ bits. As expected, the limited changes in the distance spectrum with respect to the shorter block length counterpart (resulting in some case in larger multiplicities of low-weight codewords) observable in Table I lead to a visible loss with respect to the theoretical benchmarks. In particular, for the rate-$1/3$ case, a judiciously-designed binary turbo code with 16-states tail biting component codes provides nearly the same performance of the (much more complex) memory-14 TBCC.

To further emphasize the block length regime in which TBCCs provide a competitive solution, we then derived the E_b/N_0 required to achieve a CER of 10^{-4} and 10^{-6} as a function of the block length n. The results are depicted in Figures 6 and 7, and are shown for the rate $1/2$ case. When a moderate error rate (i.e., CER$\approx 10^{-4}$) is targeted, block lengths up to $n = 192$ bits can be considered for memory-11 and memory-14 codes, which perform within 1 dB from the SPB. When lower error rates are required (i.e., CER$\approx 10^{-6}$), then the applicability of TBCCs shall be limited to shorter blocks.

A final remark deals with the application of the TSB to estimate the performance of TBCCs under WAVA decoding. In fact, given the distance spectrum of a code, the TSB provides a tight upper bound on the block error probability

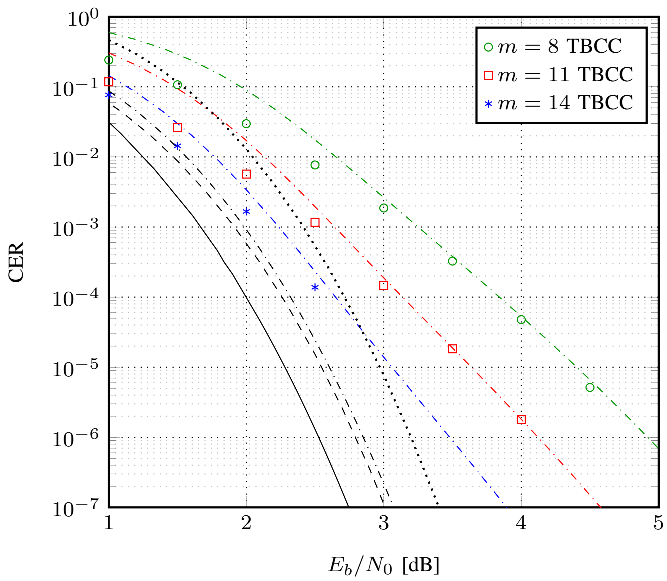

Fig. 5. Codeword error rate vs E_b/N_0 for $(256, 128)$ TBCCs. The markers denote simulation results, whereas the dashed lines provide the corresponding codes TSBs on the ML block error probability.

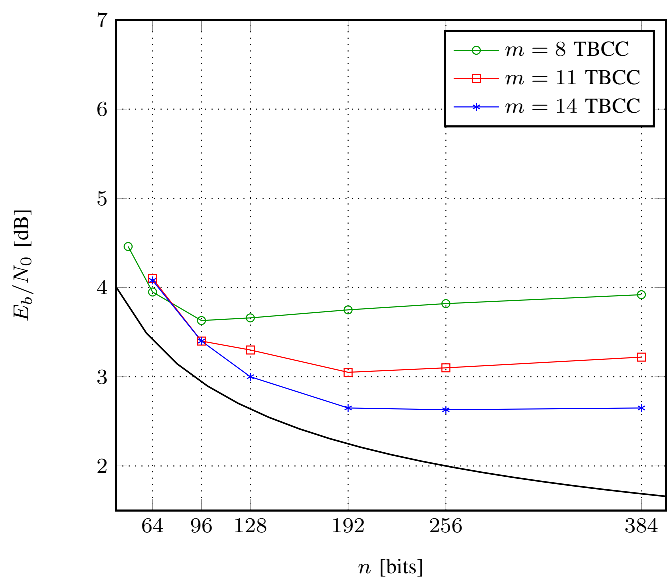

Fig. 6. E_b/N_0 required to achieve a CER of 10^{-4}. Various rate-$1/2$ TBCCs compared with the SPB (——).

under ML decoding. The WAVA is a sub-optimum decoding algorithm for which the TSB does not represent an actual upper bound on the decoding error probability. Anyhow, for all the simulated codes, the CER under WAVA has been consistently below the TSB at high error rates, and tightly close to it at low error rates, providing an empirical evidence of the near-ML performance of the decoding algorithm. At lower error rates (where simulation results are not available, i.e., below CER$= 10^{-7}$) and for other TBCCs, the performance predicted by the TSB shall be taken *cum grano salis*, unless the near-ML performance of the WAVA is proved in a strong sense.

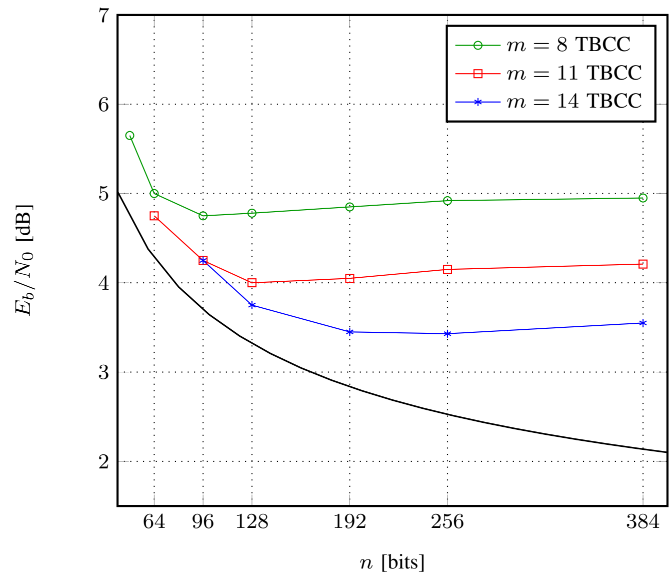

Fig. 7. E_b/N_0 required to achieve a CER of 10^{-6}. Various rate-$1/2$ TBCCs compared with the SPB (——).

VI. Conclusions

We analyzed the performance of TBCCs with different memories, block lengths and code rates over the AWGN. For the short block lengths, TBCCs outperform significantly both binary and non-binary iterative coding schemes, achieving block error rates that are remarkably close to theoretical benchmarks down to moderate and low block error rates. As expected their performance degrades visibly with increasing block lengths, limiting their appeal to blocks of roughly 200 bits or less. The use of robust decoding algorithms (as e.g. [35]) enabling error detection may represent a further direction to explore for the application of TBCCs to ultra-reliable communication links.

Acknowledgments

The authors would like to thank Prof. Giulio Colavolpe for his feedback on an early version of this manuscript.

References

[1] P. Popovski, "Ultra-reliable communication in 5G wireless systems," in *1st International Conference on 5G for Ubiquitous Connectivity (5GU)*, Nov 2014, pp. 146–151.

[2] S. A. Ashraf, F. Lindqvist, R. Baldemair, and B. Lindoff, "Control channel design trade-offs for ultra-reliable and low-latency communication system," in *IEEE Globecom Workshops*, Dec 2015, pp. 1–6.

[3] G. Durisi, T. Koch, and P. Popovski, "Towards massive, ultra-reliable, and low-latency wireless communication with short packets," *Proceedings of the IEEE*, to appear, 2016.

[4] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Geneva, Switzerland, May 1993.

[5] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: M.I.T. Press, 1963.

[6] T. Richardson, M. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[7] S. Dolinar, D. Divsalar, and F. Pollara, “Code performance as a function of block size,” Jet Propulsion Laboratory, Pasadena, CA, USA, TMO progress report 42-133, May 1998.

[8] C. Poulliat, M. Fossorier, and D. Declercq, “Design of regular $(2, d_c)$-LDPC codes over GF(q) using their binary images,” *IEEE Trans. Commun.*, vol. 56, no. 10, pp. 1626–1635, 2008.

[9] L. Costantini, B. Matuz, G. Liva, E. Paolini, and M. Chiani, “On the performance of moderate-length non-binary LDPC codes for space communications,” in *Proc. 5th Adv. Sat. Mobile Sys. Conf. (ASMS)*, Cagliari, Italy, Sep. 2010.

[10] G. Liva, E. Paolini, T. D. Cola, and M. Chiani, “Codes on high-order fields for the CCSDS next generation uplink,” in *2012 6th Advanced Satellite Multimedia Systems Conference (ASMS) and 12th Signal Processing for Space Communications Workshop (SPSC)*, Sept 2012, pp. 44–48.

[11] G. Liva, E. Paolini, B. Matuz, S. Scalise, and M. Chiani, “Short turbo codes over high order fields,” *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2201–2211, June 2013.

[12] B. Y. Chang, D. Divsalar, and L. Dolecek, “Non-binary protograph-based LDPC codes for short block-lengths,” in *Proc. IEEE Inf. Theory Workshop (ITW)*, Lausanne, Switzerland, Sep. 2012.

[13] L. Dolecek, D. Divsalar, Y. Sun, and B. Amiri, “Non-binary protograph-based LDPC codes: Enumerators, analysis, and designs,” *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 3913–3941, July 2014.

[14] T. Hehn and J. B. Huber, “LDPC codes and convolutional codes with equal structural delay: a comparison,” *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1683–1692, June 2009.

[15] S. V. Maiya, D. J. Costello, and T. E. Fuja, “Low latency coding: Convolutional codes vs. ldpc codes,” *IEEE Trans. Commun.*, vol. 60, no. 5, pp. 1215–1225, May 2012.

[16] C. Rachinger, J. B. Huber, and R. R. Müller, “Comparison of convolutional and block codes for low structural delay,” *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 4629–4638, Dec. 2015.

[17] J. Heller and I. Jacobs, “Viterbi decoding for satellite and space communication,” *IEEE Transactions on Communication Technology*, vol. 19, no. 5, pp. 835–848, October 1971.

[18] G. Solomon and H. Van Tilborg, “A connection between block and convolutional codes,” *SIAM Journal on Applied Mathematics*, vol. 37, no. 2, pp. 358–369, 1979.

[19] J. Sachs, “5G Enabling reliable low latency communication for connected industries,” in *Johannesberg Summit*, Stockholm, Sweden, May 2015.

[20] 5G Forum, “5G Vision, Requirements, and Enabling Technologies,” White Paper, 2016.

[21] IMT-2020 Promotion Group, “5G Wireless Technology Architecture,” White Paper, 2016.

[22] C. E. Shannon, “Probability of error for optimal codes in a Gaussian channel,” *Bell Syst. Tech. J.*, vol. 38, pp. 611–656, May 1959.

[23] R. Gallager, *Information theory and reliable communication*. New York, NY, USA: Wiley, 1968.

[24] Y. Polyanskiy, H. 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.

[25] R. Y.Shao, S. Lin, and M. P. Fossorier, “Two decoding algorithms for tailbiting codes,” *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1658–1665, Oct. 2003.

[26] G. Poltyrev, “Bounds on the decoding error probability of binary linear codes via their spectra,” *IEEE Trans. Inf. Theory*, vol. 40, no. 4, pp. 1284–1292, Jul 1994.

[27] I. Sason and S. Shamai, *Performance analysis of linear codes under maximum-likelihood decoding: A tutorial*. Now Publishers, 2006.

[28] P. Stahl, J. B. Anderson, and R. Johannesson, “Optimal and near-optimal encoders for short and moderate-length tail-biting trellises,” *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2562–2571, Nov 1999.

[29] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding (Second Edition)*. John Wiley & Sons, 2015.

[30] I. E. Bocharova, R. Johannesson, B. D. Kudryashov, and P. Stahl, “Tailbiting codes: bounds and search results,” *IEEE Trans. Inf. Theory*, vol. 48, no. 1, pp. 137–148, Jan 2002.

[31] J. K. Wolf and A. J. Viterbi, “On the weight distribution of linear block codes formed from convolutional codes,” *IEEE Trans. Commun.*, vol. 44, no. 9, pp. 1049–1051, Sep 1996.

[32] G. Liva, E. Paolini, and M. Chiani, “Bounds on the error probability of block codes over the q-ary erasure channel,” *IEEE Trans. Commun.*, vol. 61, no. 6, pp. 2156–2165, June 2013.

[33] T. Jerkovits and B. Matuz, “Turbo code design for short blocks,” in *Proc. 7th Advanced Satellite Mobile Systems Conference*, Maiorca (Spain), September 2016.

[34] *Next Generation Uplink*, Green Book, Issue 1, Consultative Committee for Space Data Systems (CCSDS) Report Concerning Space Data System Standards 230.2-G-1, Jul. 2014.

[35] A. R. Williamson, M. J. Marshall, and R. D. Wesel, “Reliability-output decoding of tail-biting convolutional codes,” *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 1768–1778, June 2014.