

Performance Improvement of Space Missions Using Convolutional Codes by CRC-Aided List Viterbi Algorithms

*Original*

Performance Improvement of Space Missions Using Convolutional Codes by CRC-Aided List Viterbi Algorithms / Schiavone, Riccardo; Garelo, Roberto; Liva, Gianluigi. - In: IEEE ACCESS. - ISSN 2169-3536. - ELETTRONICO. - 11:(2023), pp. 55925-55937. [10.1109/ACCESS.2023.3283473]

*Availability:*

This version is available at: 11583/2979350 since: 2023-08-05T14:57:49Z

*Publisher:*

IEEE

*Published*

DOI:10.1109/ACCESS.2023.3283473

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

Received 9 May 2023, accepted 29 May 2023, date of publication 6 June 2023, date of current version 9 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3283473

## RESEARCH ARTICLE

# Performance Improvement of Space Missions Using Convolutional Codes by CRC-Aided List Viterbi Algorithms

**RICCARDO SCHIAVONE<sup>1</sup>**, **ROBERTO GARELLO<sup>1</sup>**, (Senior Member, IEEE),  
**AND GIANLUIGI LIVA<sup>2</sup>**, (Senior Member, IEEE)

<sup>1</sup>Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy

<sup>2</sup>German Aerospace Center (DLR), Institute of Communications and Navigation, 82234 Wessling, Germany

Corresponding author: Riccardo Schiavone (riccardo.schiavone@polito.it)

This work was supported in part by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” through the Program “RESTART” under Grant PE00000001.

**ABSTRACT** Recently, CRC-aided list decoding of convolutional codes has gained attention thanks to its remarkable performance in the short blocklength regime. This paper studies the convolutional and CRC codes of the Consultative Committee for Space Data System Telemetry recommendation used in space missions by all international space agencies. The distance spectrum of the concatenated CRC-convolutional code and an upper bound on its frame error rate are derived, showing the availability of a 3 dB coding gain when compared to the maximum likelihood decoding of the convolutional code alone. The analytic bounds are then compared with Monte Carlo simulations for frame error rates achieved by list Viterbi decoding of the concatenated codes, for various list sizes. A remarkable outcome is the possibility of approaching the 3 dB coding gain with nearly the same decoding complexity of the plain Viterbi decoding of the inner convolutional code, at the expense of slightly increasing the undetected frame error rates at medium-high signal-to-noise ratios. Comparisons with CCSDS turbo codes and low-density parity check codes highlight the effectiveness of the proposed solution for onboard utilization on small satellites and cubesats, due to the reduced encoder complexity and excellent error rate performance.

**INDEX TERMS** Forward error correction, list Viterbi algorithm, convolutional codes, CRC, telemetry, space missions.

## I. INTRODUCTION

The method of combining a convolutional code (CC) with a cyclic redundancy check (CRC) code has gained widespread acceptance in the context of (hybrid) automatic repeat request (ARQ) protocols. This approach, for instance, has been integrated into the cellular communication standards of both 3G [2] and 4G LTE [3]. Typically, the CC acts as an inner error-correcting code, while the CRC functions as an outer error-detecting code that validates the accuracy of the decoded codeword. The concept of powerful concatenations of CCs with high-rate outer binary linear block codes was introduced in [4], demonstrating efficient decoding through

the utilization of list Viterbi algorithms (LVAs) [5], which mirrors the approach employed for decoding turbo polar codes combined with CRC codes [6], [7].

In the short block length regime, concatenating an inner convolutional code with an outer CRC code (referred to hereafter as “CRC+CC”) with moderate/small memory has shown remarkable performance, closely approaching finite block length bounds [8] even at low error rates, while maintaining doable decoding complexity [4]. Extensive research has been conducted on CRC+CC schemes [1], [4], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21].

Despite the availability in Consultative Committee for Space Data Systems (CCSDS) telemetry synchronization and channel coding recommendation [22] of more performant

The associate editor coordinating the review of this manuscript and approving it for publication was Yuan Gao<sup>1</sup>.

error-correcting codes such as turbo codes and low-density parity-check (LDPC), CCs continue to be used in various space missions due to their minimal encoder complexity and excellent performance, especially for on-board applications in small satellites and cubesats. It is worth noting that when CCs are employed in a telemetry (TM) link, the CCSDS TM and the advanced orbiting systems (AOS) space data link protocol [23], [24] mandate the use of an outer CRC code. The inclusion of a CRC code is crucial for error detection following Viterbi decoding of the inner CC. However, by adopting the decoding technique proposed in [4], the outer CRC code becomes an integral part of the error correction system and it is no longer solely used for error detection.

### A. CONTRIBUTIONS

The main contributions of this paper are summarized as follows.

- We present a method to compute the distance spectrum of the CRC+CC scheme defined by the CCSDS Telemetry Recommendation. The distance spectrum is used to compute analytic upper bounds on the frame error rates (FERs) of the concatenated scheme. The bounds show that a 3 dB gain is potentially available moving from convolutional code decoding to concatenated code decoding.
- By means of Monte Carlo simulations, we evaluate both the FER and the undetected frame error rate (UFER) of the CRC+CC scheme when decoded by means of list Viterbi algorithms, comparing the results for various list sizes. We show that the 3 dB gain can be achieved with reasonable list size, while incurring in a small penalty in terms of undetected frame error rates compared with the Viterbi algorithm (VA) decoding of the inner CC only.
- We analyze the on-ground decoding complexity and we show that it is manageable at medium and high SNR, with a slight increase with respect to the Viterbi complexity.
- We compare the CRC+CC scheme with modern coding options of the CCSDS TM recommendation. Results show that the gap is very limited, especially at high rates. Since the encoder complexity of convolutional codes is negligible, unlike that of turbo codes (that have an interleaver) and low density parity check codes (that have a non-sparse large generator matrix), the performance makes this solution extremely appealing for small satellites with limited on-board computational complexity.

The article significantly extends the work presented in preliminary form in [1], where we showed the potential of the CCSDS CRC+CC. Among the numerous additions compared to [1], we can mention:

- The presentation of the method used to compute the distance spectrum of the CRC+CC scheme.
- The extension of the Monte-Carlo simulations of the Frame Error Rate of the CRC+CC concatenation to higher list sizes.

- The Monte-Carlo simulations of the undetected error rate of the CRC+CC scheme for various list sizes and the analysis of the obtained results and trade-off.
- The comparisons between the CRC+CC codes with the turbo and LDPC coding options of the CCSDS TM recommendation in terms of frame error rate, spectral efficiencies and encoding complexities.

### B. ORGANIZATION

The work is structured as follows. We introduce the notation and we review aspects of the CCSDS telemetry recommendation in Section II. In Section III list Viterbi algorithms are recalled. We detail in Section IV the iterative parallel-list Viterbi algorithm. The construction of the CRC+CC trellis is presented in Section V, together with the derivation of the distance spectrum of the CRC+CC concatenation of the CCSDS recommendations and the error rate upper bounds. Numerical results on the frame error rate and the undetected frame error rates are presented in Section VI. A comparison with the other coding schemes options of the CCSDS TM recommendation is presented in Section VII. Conclusions follow in Section VIII.

## II. CCSDS TELEMETRY RECOMMENDATION

### A. TRANSFER FRAMES

The TM and the AOS recommendations from the CCSDS [23], [24] provide essential functionalities for data transfer utilizing a protocol data unit known as the transfer frame (TF). Within the synchronization and channel coding sublayer, auxiliary functions are offered to facilitate TF transmission across the space link. These functions encompass error-control coding and decoding, TF delimiting and synchronization, as well as bit transition generation and removal.

Various families of channel codes can be chosen as coding options, including convolutional codes, parallel/serial turbo codes, Reed-Solomon codes, concatenated Reed-Solomon and convolutional codes, and low-density parity-check codes. It is worth mentioning that the decoders for the latter three code families possess a native transfer frame validation property, enabling error detection. In contrast, this capability is not inherent in CC and turbo codes. Consequently, when utilizing the latter two code families, the CRC (Transfer Frame Error Control Field) defined in [23], and [24] is compulsory. However, for the other three code families, the presence of the CRC field is optional.

With the exception of CCs, the TF is encoded into a codeword, and then an attached synchronization marker (ASM) is appended as prefix. The ASM is a fixed binary sequence that fulfills the delimiting function of the TF required for frame synchronization. For CCs, the procedure differs, and the TF is first preceded by a 32-bit long ASM pattern, denoted as  $(1ACFFC1D)_{\text{HEX}}$ , and the ASM is also encoded.

It is valuable to highlight that when there are no TFs available, a frame containing dummy data, and called only idle data (OID) frame, is encoded to carry on the link

availability. Subsequently, before being convolutionally coded, the information sequences are arranged as continuous sequences comprising data/OID TFs separated by ASM patterns.

A high-level view of the CCSDS convolutionally coded systems is shown in Fig. 1, where the synchronization and channel coding operations of a TM CCSDS compatible transmitter are depicted.

**B. CONVOLUTIONAL AND CYCLIC REDUNDANCY CHECK CODES**

The recursive and systematic CRC encoder with generator polynomial

$$g(D) = 1 + D^5 + D^{12} + D^{16}$$

is used to generate the 16-bit vector denoted as *Transfer Frame Error Control Field* and whose encoder circuit is depicted in Fig. 2.

The circuit is loaded with all ones at time  $t = 0$ , i.e., at the beginning of each TF. The switch remains in position  $s_{in}$  for the entire frame duration and is then switched to position  $s_{out}$  (and  $s_m$  is open) to generate the 16 parity bits.

The memory  $v = 6$  (64 states) non-recursive CC encoder with polynomial generator matrix

$$G(D) = \left[ 1 + D + D^2 + D^3 + D^6, 1 + D^2 + D^3 + D^5 + D^6 \right]$$

is adopted in [22]. Its encoding circuit is depicted in Fig. 3.

In theory, the CC encoder is not terminated. However, in practice, the encoding of the ASM forces the start/end states to have fixed values. Let us denote by  $\sigma_i$  the  $i$ -th state of the trellis. Since the CC encoder is feedforward, the CC encoder is forced to the binary state  $\sigma_{start} = (101110)$  at the beginning of a frame and to  $\sigma_{end} = (110101)$  at the end of it. Those bits are the last 6 bits of ASM preceding the TF and the first 6 bits of the ASM following it, respectively. In the remainder of the work, we denote by  $K$  the length of the TF in bits, prior to CRC encoding. Denoting by  $m = 16$  and  $h = 32$  the number of CRC parity bits and the length of the ASM, respectively, we get that the code rate is given by

$$R = \frac{K}{K + m + h} R_{CC}$$

where  $R_{CC}$  is the code rate of the convolutional encoder (equal to 1/2 for the non-punctured case).

*Remark 1:* The CRC+CC code, having the termination conditions imposed by the ASM and the CRC encoder being initialized to the all-one state, is not strictly a binary linear code. However, the codewords generated by CRC+CC form a coset of the CRC+CC binary linear code. In this study, we examine the performance of the code over the binary-input additive white Gaussian noise (AWGN) channel. Due to the channel’s symmetry, analyzing the affine CRC+CC concatenation can be reduced to analyzing the linear CRC+CC concatenation. This reduction is achieved by (i) initializing the CRC circuit loading only zeros (ii) enforcing the CC starting and ending states to be the all-zero

**TABLE 1. CCSDS convolutional encoder puncturing patterns with corresponding rates. 1 indicates that the bit is transmitted, 0 that the bit is punctured.**

Rate $R_{CC}$	Puncturing Pattern
2/3	[1,1,0,1]
3/4	[1,1,0,1,1,0]
5/6	[1,1,0,1,1,0,0,1,1,0]
7/8	[1,1,0,1,0,1,0,1,1,0,0,1,1,0]

state. For the remainder of this paper, we will leverage this observation and focus our analysis on the linear CRC+CC concatenation.

The convolutional encoder of [22] has a rate of  $R_{CC} = 1/2$ , but it is possible to increase the rate by using a puncturer in cascade with the encoder itself. The periodic puncturing patterns with the corresponding rates are shown in Table 1. In the periodic pattern, a “1” indicates a bit which is transmitted, while a “0” indicates a punctured bit which is not transmitted. The bits of the puncturing patterns at even positions, beginning with the position 0, refer to the output bits of the first polynomial of the generator matrix, while the bits at odd positions refer to the output bits of the second polynomial of the generator matrix. For instance, given the CC encoder output

$$\left[ c_t^{(1)}, c_t^{(2)}, c_{t+1}^{(1)}, c_{t+1}^{(2)}, c_{t+2}^{(1)}, c_{t+2}^{(2)}, c_{t+3}^{(1)}, c_{t+3}^{(2)}, c_{t+4}^{(1)}, c_{t+4}^{(2)}, c_{t+5}^{(1)}, \dots \right]$$

by setting the code rate to  $R_{CC} = 2/3$  through the application of the puncturing pattern [1, 1, 0, 1] we obtain the transmitted sequence

$$[c_t^{(1)}, c_t^{(2)}, c_{t+1}^{(2)}, c_{t+2}^{(1)}, c_{t+2}^{(2)}, c_{t+3}^{(2)}, c_{t+4}^{(1)}, c_{t+4}^{(2)}, \dots].$$

This means that the mother rate-1/2 encoder is the same for all convolutional codes of the CCSDS standard and only the correct puncturing pattern is needed to adjust the rate of the code.

**III. CRC-AIDED LIST VITERBI DECODING ALGORITHMS**

In this section, we will review LVAs and their application in improving the performance of systems that employ convolutional codes in conjunction with an outer CRC code, while also considering their impact on error detection capabilities.

We indicate with  $\mathcal{I}_i$  the set of CC trellis state indexes at time  $t$  that are connected to state  $i$  at time  $t + 1$ . Additionally, we represent with  $\lambda_i^{\sigma_j \rightarrow \sigma_i}$  the VA state transition metric over the edge connecting state  $\sigma_j$  with state  $\sigma_i$  at time  $t$ . When considering the binary-input AWGN channel, the state transition metric is defined as the correlation between the modulated edge label and the corresponding observation at the channel output. Furthermore,  $\Lambda_t^{\sigma_i}$  stands for the cumulative metric for state  $\sigma_i$  in the  $t$ -th trellis section.

The parallel-list Viterbi algorithm (PLVA) [5] is obtained with a straightforward modification of the VA which allows to

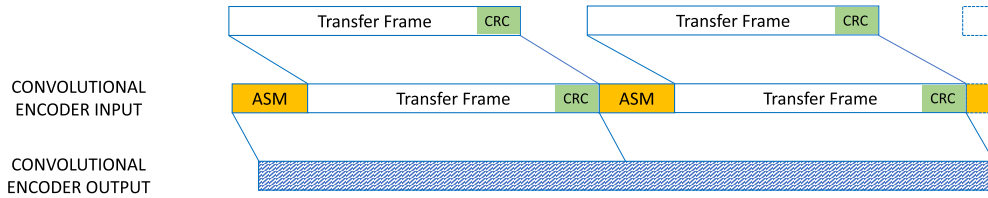


FIGURE 1. High-level view of the CCSDS TM synchronization and channel coding option of a convolutional coded transmission.

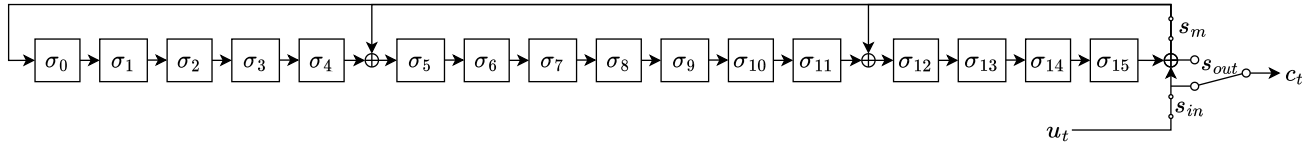


FIGURE 2. Encoder circuit of the transfer frame error control field of a CCSDS TM compatible transmitter.

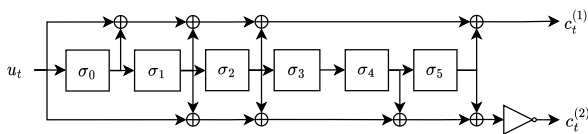
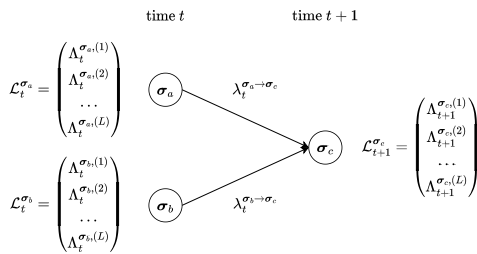


FIGURE 3. Encoder circuit of the convolutional encoder of a CCSDS TM compatible transmitter.



$$\Lambda_{t+1}^{\sigma_c} = \max^{(\ell)} \left\{ \Lambda_t^{\sigma_a(1)} + \lambda_t^{\sigma_a \rightarrow \sigma_c}, \dots, \Lambda_t^{\sigma_a(L)} + \lambda_t^{\sigma_a \rightarrow \sigma_c}, \Lambda_t^{\sigma_b(1)} + \lambda_t^{\sigma_b \rightarrow \sigma_c}, \dots, \Lambda_t^{\sigma_b(L)} + \lambda_t^{\sigma_b \rightarrow \sigma_c} \right\}$$

FIGURE 4. Generation of the list at a generic trellis state using the parallel-list Viterbi algorithm.

output an list  $\mathcal{L}$  containing the  $L$  trellis paths with the highest likelihood. Here is how the PLVA operates.

At time  $t = 0$ , decoding starts from the all-zero state with  $\mathcal{L}_0^{\sigma_0}$  containing only the element  $\Lambda_0^{\sigma_0(1)} = 0$ . Then, for each state  $\sigma_i$  in the trellis branches at time  $t$ ,  $\mathcal{L}_t^{\sigma_i}$  is constructed as follows

$$\Lambda_{t+1}^{\sigma_i} = \max_{\substack{j \in \mathcal{I}_t \\ z=1, \dots, L}}^{(\ell)} \left( \Lambda_t^{\sigma_j(z)} + \lambda_t^{\sigma_j \rightarrow \sigma_i} \right) \quad (1)$$

where  $\max^{(\ell)}$  is an operator that returns the  $\ell$ -th largest argument. The process is illustrated in Fig. 4. Once the final trellis section is reached, the  $L$  paths stored in the final state list are extracted and each path is tested using the outer CRC code. If none of the paths satisfy the CRC constraints, a decoding error is raised. Otherwise, the path with the highest cumulative metric among those satisfying the CRC constraints is output as the final decision.

Another decoding possibility is the use of the serial-list Viterbi algorithm (SLVA) [5]. This algorithm operates

iteratively, outputting the  $\ell$ -th most likely path after the  $\ell$ -th iteration to save computational resources. Decoding halts once a path satisfies the CRC check, potentially stopping the search at an early phase. Initially, SLVA is equivalent to the VA. If the selected path by the VA does not pass the CRC check, SLVA scans the trellis to identify the path with the smallest likelihood difference compared to the maximum likelihood (ML) path. It then verifies whether the CRC code constraints are met for that path. If the constraints are not satisfied, the algorithm proceeds to another iteration. At the  $\ell$ -th step, SLVA scans the paths that have not been examined yet, focusing on those with the smallest likelihood difference from the  $(\ell - 1)$  most likely paths, and extracts the  $\ell$ -th most likely path. This process continues until the message associated with the  $\ell$ -th most likely path passes the CRC check, or until  $L$  iterations are reached. By leveraging the tree structure of the trellis, significant computational savings can be achieved, and the SLVA can be speeded up using techniques described in [25], and [26].

*Remark 2:* The combination of the CRC code and CCs was initially introduced in the CCSDS telemetry synchronization and channel coding recommendation [22] to serve as an error detection mechanism for the receiver. However, when using the decoders described in this section, the role of the outer CRC code has evolved beyond pure error detection and has become integrated into the error correction process [4]. Despite this transformation, the approach still maintains a degree of error detection capability. Specifically, a decoding error can be identified when none of the paths included in the final list satisfy the CRC code constraints. It's important to note that the size of the list, denoted as  $L$ , significantly impacts the undetected frame error rate. As  $L$  increases, the likelihood of undetected errors rises. It's worth mentioning that regardless of the list size, the undetected frame error rate remains upper bounded by the ML FER of the CRC+CC code. Further mathematical analysis of the undetected frame error rate, which holds significance in the design of telemetry links, will be explored in future studies. In Section VI, we present simulation results to complement these discussions.

#### IV. ITERATIVE PARALLEL-LIST VITERBI ALGORITHM

In this section, we present an implementation approach based on the PLVA with a decoder that exhibits decreasing algorithmic complexity as the signal-to-noise ratio (SNR) increases. The iterative PLVA is employed by running multiple instances, starting with a small list size  $L$  (e.g.,  $L = 1$ ), and incrementing  $L$  whenever the CRC code constraints are not met for all paths in the list or when  $L$  exceeds the maximum list size  $L_{\max}$ . This approach emulates the behavior of the SLVA and results in reduced average algorithmic complexity with increasing SNR.

The iterative PLVA incorporates a **scheduler** that determines how  $L$  is increased whenever the CRC conditions are not satisfied. The scheduler, denoted as  $sched(\cdot)$ , takes the current value of  $L$  at the  $i$ -th iteration ( $L^{(i)}$ ) and outputs the value of  $L$  at the  $(i + 1)$ -th iteration ( $L^{(i+1)}$ ), where  $L^{(i+1)} > L^{(i)}$ . For instance, a straightforward constant increase in the list size can be achieved with  $L^{(i+1)} = sched(L^{(i)}) = L^{(i)} + 1$ , or a doubling of the list size at each iteration with  $L^{(i+1)} = sched(L^{(i)}) = 2 \cdot L^{(i)}$ . It is important to note that the chosen scheduler impacts both the algorithm's delay and complexity in the worst-case scenario, which occurs when  $L$  reaches  $L_{\max}$ .

The procedure of the iterative PLVA is described in Algorithm 1. Given a received sequence  $\mathbf{y}$ , a maximum list size  $L_{\max}$ , and the scheduler  $sched(\cdot)$ , the algorithm produces a list  $\mathcal{L}$  that contains the messages associated with the  $L$  most likely paths over the trellis. The list is sorted in decreasing order of their likelihood. Additionally, a flag named NACK is used to indicate whether none of the found messages satisfy the CRC code constraints (NACK = 1) or if at least one does (NACK = 0). Lastly,  $\ell$  represents the position of the most likely message that satisfies the CRC conditions, if any, within the list.

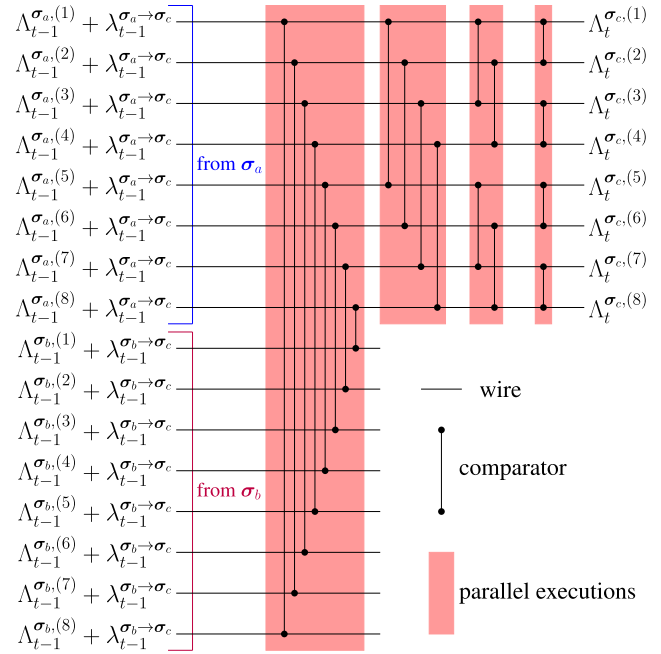
**Algorithm 1** Step-by-Step Mechanism of the Iterative Parallel-List Viterbi Algorithm Applied to the Received Sequence  $\mathbf{y}$  Under the Constraint of a Maximum List Size  $L_{\max}$  and Utilizing the List Increment Function  $sched(\cdot)$

```

1: procedure iterative_PLVA( $\mathbf{y}$ ,  $L_{\max}$ ,  $sched(\cdot)$ )
2:    $L \leftarrow 1$ 
3:    $\mathcal{L} \leftarrow \emptyset$ 
4:   NACK  $\leftarrow 1$ 
5:   while ( $L \leq L_{\max}$  and NACK = 1) do
6:      $(\mathcal{L}, \text{NACK}, \ell) = \text{PLVA}(\mathbf{y}, L)$ 
7:      $L \leftarrow sched(L)$ 
8:   end while
9:   return ( $\mathcal{L}$ , NACK,  $\ell$ )
10: end procedure

```

For software implementation, sorting the list at each trellis node for the PLVA with list size  $L$  takes into account that each node, in the case of a rate-1/2 convolutional code, has two predecessors, each with an already sorted list. To merge these lists efficiently, a minimum complexity merge operation can be performed using  $L$  comparisons in series. Starting



**FIGURE 5.** Example of the bitonic merger to sort 2 trellis nodes with  $L = 8$  elements each. The executions which can be parallelized are highlighted in red boxes.

from the best element of each list, the list index of the node whose element was selected in the previous comparison is incremented. This approach results in a sorting delay of  $L$ . Another option for sorting the list elements is to utilize modern CPU/GPUs with multiple computing cores, enabling parallel execution of several comparisons using a sorting network [27]. The same sorter can also be implemented in hardware (ASIC, FPGA, ...).

In [14] a bitonic merger, which is a sorting network based on the bitonic sorter [28], but with delay  $O(\log_2(L))$  is proposed instead of  $O(\log_2^2(L))$  of the original bitonic sorter. It requires  $\frac{1}{2} \log_2(L) + L$  comparators, instead of  $L \log_2^2(L)$  of the bitonic sorter. We depict in Fig. 5 an instance of the sorting network when we merge two nodes with list size  $L = 8$ , and we refer to [14, Algorithm 2] for the generalization of the sorting network for any list size  $L$ .

#### V. DISTANCE SPECTRUM OF CRC+CC CODES

##### A. DISTANCE SPECTRUM CALCULATION

The distance spectrum of a zero-tail (ZT) terminated convolutional code can be derived by the approach outlined in [29], which relies on the trellis representation of the convolutional code. As observed in [4], when both the CC and the CRC encoders are non-systematic, the CRC+CC code can be described by the encoder of a CC with larger memory. More specifically, the memory of the resulting CC is  $(\nu + m)$ , where  $m$  is the degree of the CRC code generator polynomial and  $\nu$  is the memory of the inner convolutional code. The generator matrix transfer function of the CRC+CC code is

$$\begin{aligned}
\mathbf{G}(D) &= \mathbf{g}_{CRC}(D) \cdot \mathbf{G}_{CC}(D) \\
&= [\mathbf{g}_{CRC}(D) \cdot \mathbf{G}^{(1)}(D), \dots, \mathbf{g}_{CRC}(D) \cdot \mathbf{G}^{(n)}(D)] \quad (2)
\end{aligned}$$

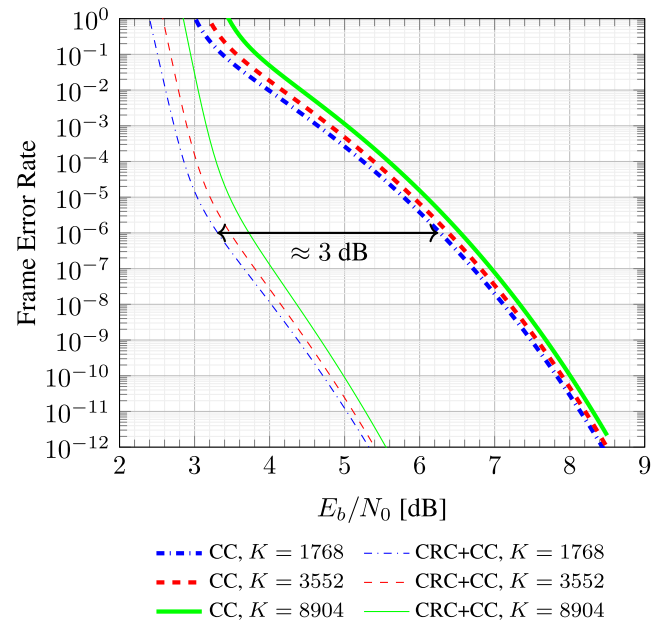
where  $G_{CC}(D)$  is the generator matrix transfer function of the inner CC, while  $g_{CRC}(D)$  is the generator polynomial of the CRC code. The trellis diagram of the code defined by Eq. (2) possesses  $2^{\nu+m}$  states per section. When the memory  $(\nu + m)$  is moderate-small, it is possible to label the trellis branches with monomials in the form  $x^d$ , where  $d$  is the Hamming weight of the encoder output associated to the given branch. Denote by  $T(x)$  the  $2^{\nu+m} \times 2^{\nu+m}$  transition matrix, where the  $i, j$  entry corresponds to the label (i.e., the monomial  $x^d$ ) of a branch connecting state  $i$  to state  $j$  in the trellis diagram. The weight enumerating function (WEF) of the CRC+CC code can be obtained by computing first  $W(x) = T^{K+\nu+m}(x)$ , and then by isolating the element in position  $(0, 0)$ ,  $W_{0,0}(x)$ . Renaming  $A(x) = W_{0,0}(x)$ , we can recognize that  $A(x) = \sum_w A_w x^w$  is the WEF of the code, with  $A_0, A_1, \dots$  being the distance spectrum (i.e., the weight enumerator) [29]. Despite the fact that the CRC code of the CCSDS CRC+CC code is systematic, this is not a problem for the computation of the distance spectrum and the above-mentioned analysis is still valid. This is due to the fact that both a systematic and a non-systematic CRC code, with equal generator polynomial, generate the same codebook, but with different input-output relationship. This means that the input messages entering the convolutional code are the same for both the systematic and the non-systematic CRC codes, and thus also their weight enumerators are the same. For the CCSDS CRC+CC code, the memory amounts to  $(\nu + m) = 22$ , rendering the straightforward application of the above-mentioned technique challenging. The approach can be simplified by performing a limited search over the code trellis as proposed in [14]. This result in the calculation of the lower tail of the distance spectrum, only, which in any case is sufficient to properly estimate the upper bounds.

**B. ASYMPTOTIC CODING GAIN ANALYSIS**

In this section, we delve into the analysis of the achievable asymptotic coding gain provided by the CCSDS CRC+CC code when decoded according to the approach outlined in Section III. The analysis revolves around establishing a union upper bound on the FER of both the solely CC and the CRC+CC code. To initiate the investigation, we derive the inner CC’s distance spectrum. The WEF of the code (limited to the lower tail of the distance spectrum) for various values of  $K$  is presented in Table 2. A notable observation is the doubling of the minimum distance of the inner CC achieved through the concatenation with the outer code. Furthermore, it is worth noting that the  $A_w$  terms exhibit linear growth in relation to  $K$  solely for the CC, as noted in [30], whereas these terms display quadratic growth for the CRC+CC scheme. By leveraging the doubled minimum distance and the controlled rate loss introduced by the outer CRC code, asymptotically the CRC+CC concatenation achieves a coding gain of approximately 3 dB over the inner CC under ML decoding. This highlights the effectiveness of the CRC+CC concatenation in improving the overall performance of the system.

**TABLE 2. Distance spectrum of the CC and of the CRC+CC concatenated codes of the CCSDS standard for different TF lengths.**

code	$K$	$d_{\min}$	$A_{d_{\min}}, A_{d_{\min}+1}, A_{d_{\min}+2}, \dots$
CC	1768	10	19580, 0, 67477, 0, 342205, ...
CRC+CC	1768	20	7431, 0, 28005, 0, 175576, ...
CC	3552	10	39204, 0, 135269, 0, 686517, ...
CRC+CC	3552	20	16351, 0, 91945, 0, 610136, ...
CC	8904	10	98076, 0, 338645, 0, 1719453, ...
CRC+CC	8904	20	59091, 0, 557162, 0, 3581187, ...
CC	16368	10	180180, 0, 622277, 0, 3160005, ...
CRC+CC	16368	20	197358, 0, 1800329, 0, 11847522, ...



**FIGURE 6. Comparison of truncated union bounds on the ML FER between the CCSDS TM recommended CCs and CRC+CC codes under binary phase-shift keying (BPSK) modulation and AWGN channel conditions.**

For a specific distance spectrum, we can establish an upper bound on the error probability of ML decoding the code by using the union bound as

$$P_B \leq \frac{1}{2} \sum_{w>0} A_w \operatorname{erfc} \left( \sqrt{wR} \frac{E_b}{N_0} \right). \tag{3}$$

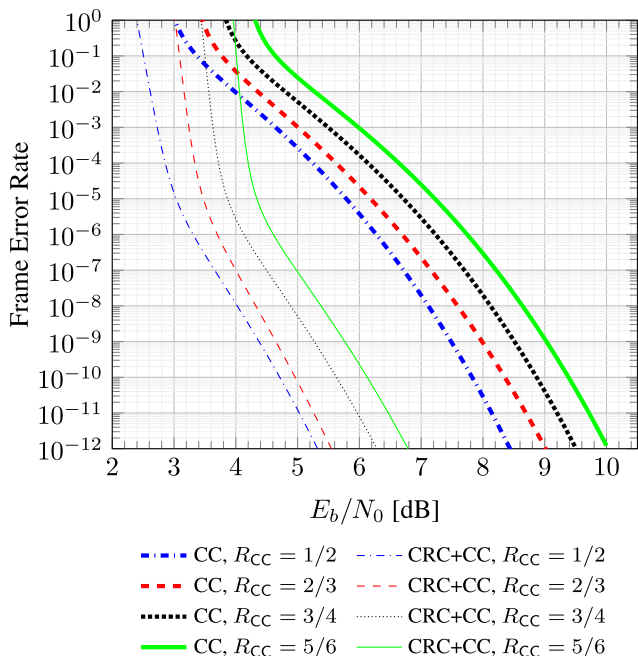
In Fig. 6, we present truncated versions of the upper bounds on  $P_B$  for both the CC and CRC+CC codes, corresponding to different code lengths  $K$ . The curves in Fig. 6 are obtained by truncating the summation in equation (3) up to  $w = 120$ . By examining frame error rates below  $10^{-6}$ , where the bound is expected to be highly accurate, we can already observe a significant coding gain of approximately 3 dB that can be achieved through the concatenation of CRC+CC.

**C. PUNCTURED CODES**

In [22] the output of the CC encoder described in Section II-B may be punctured to achieve higher code rates. Also for these higher-rate codes we have computed the lower tail of the distance spectra, which are reported in Table 3, and

**TABLE 3.** Distance spectra of the punctured CCs and CRC+CC codes of the CCSDS standard for various rates of the encoder. The TF length is fixed to  $K = 1768$ .

code	$R_{cc}$	$d_{min}$	$A_{d_{min}}, A_{d_{min}+1}, A_{d_{min}+2}, \dots$
CC	2/3	6	891, 14229, 42607, 139960, ...
CRC+CC	2/3	14	1756, 21066, 76351, 341467, ...
CC	3/4	5	4738, 18328, 94331, 524544, ...
CRC+CC	3/4	10	808, 2646, 15199, 80484, ...
CC	5/6	4	4971, 24449, 230378, 1754473, ...
CRC+CC	5/6	8	787, 4618, 36036, 317668, ...



**FIGURE 7.** Comparison of truncated union bounds on the ML FER between the CCSDS TM recommended punctured CCs and punctured CRC+CC codes under BPSK modulation and AWGN channel conditions. The TFs have length  $K = 1768$ .

the truncated union bounds on the FER under ML decoding which are depicted in Fig. 7.

Also in these cases the minimum distance is doubled (even more for the  $R_{cc} = 2/3$ ) when compared with the distance of the corresponding inner CCs of the same rates, resulting also in these cases in up to approximately 3 dB coding gain for the CRC-aided systems with respect to the non-aided ones.

## VI. NUMERICAL RESULTS

In the context of CRC-aided list Viterbi decoding, this section presents the results of Monte Carlo simulations for the CRC+CC concatenation of the CCSDS standard. The simulations are conducted considering different lengths of the uncoded TF, specifically  $K \in \{1768, 3552, 8904\}$  bits, which correspond to commonly encountered CCSDS input lengths: 1784, 3568, and 8920 bits when the additional 16 bits for CRC are included. The system assumes BPSK modulation over the AWGN channel, with perfect frame and carrier synchronization at the receiver. The received signal at the

channel output is modeled as

$$y_i = x_i + n_i,$$

where  $x_i \in \pm 1$  represents the transmitted symbol and  $n_i \sim \mathcal{N}(0, \sigma^2)$  represents Gaussian noise. The FER is estimated, and the SNR is expressed in terms of  $E_b/N_0$ , where  $E_b$  denotes the energy per information bit and  $N_0 = 2\sigma^2$  represents the noise single-sided power spectral density. The simulations aim to evaluate the performance of the CRC+CC concatenation, focusing on the achievable FER for the considered TF lengths. By conducting extensive Monte Carlo simulations, we gain insights into the system's behavior and its robustness against noise and channel impairments. The results provide valuable information on the effectiveness of CRC-aided list Viterbi decoding and demonstrate the potential benefits of the CRC+CC concatenated scheme in terms of improved error performance and SNR gain.

### A. PRACTICAL CODING GAINS FOR SPACE MISSIONS

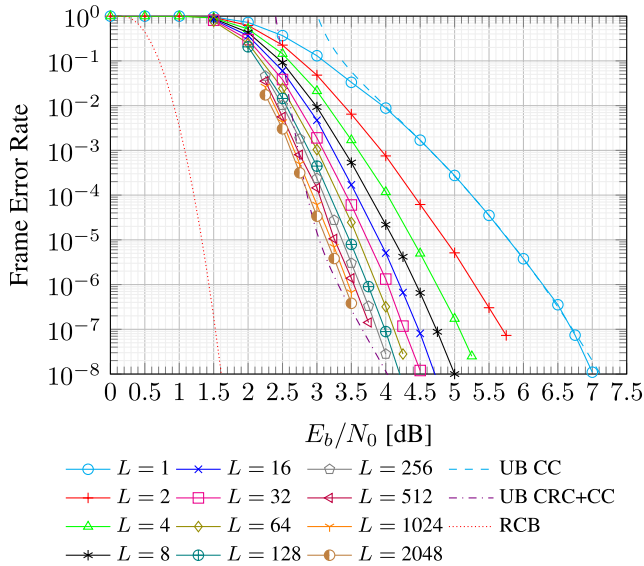
Fig. 8 reports the simulation outcomes for  $K = 1768$  (additional results for  $K = 3552$  and  $K = 8904$  are provided in the Appendix, see Fig. 14 and Fig. 15). The results are extended to various list sizes of the list Viterbi algorithms. Note that for the same list size, the performance of SLVA and PLVA is identical. Thus, we do not specify the list decoder in the figures. The Viterbi decoding of the inner CC corresponds to  $L = 1$  in the figures. On each chart, the (truncated) union bounds on the ML FER are illustrated for both the CC and the CRC+CC concatenation. In the figures, we also depict the random coding bound (RCB) introduced by Gallager [31] for the given  $R$  and  $K$ . The RCB evaluates an upper bound on the average FER of the code ensemble made of all the codes with rate  $R$  and information block size  $K$ .

Examining Fig. 8, with  $K = 1768$ , it becomes evident that higher SNR values allow for a reduction in the list size  $L$  while still approaching the ML decoding bound. For example, at  $E_b/N_0 = 4$  dB, a maximum list size of  $L = 64$  is sufficient to achieve a loss of only 0.5 dB from the ML bound for the CRC+CC code. Similarly, at  $E_b/N_0 = 4.5$  dB, the same level of performance is attained with a reduced list size of  $L = 32$ , yielding a coding gain of approximately 2.5 dB compared to the plain CC under ML FER. Comparable trends can be observed for other transfer frame lengths.

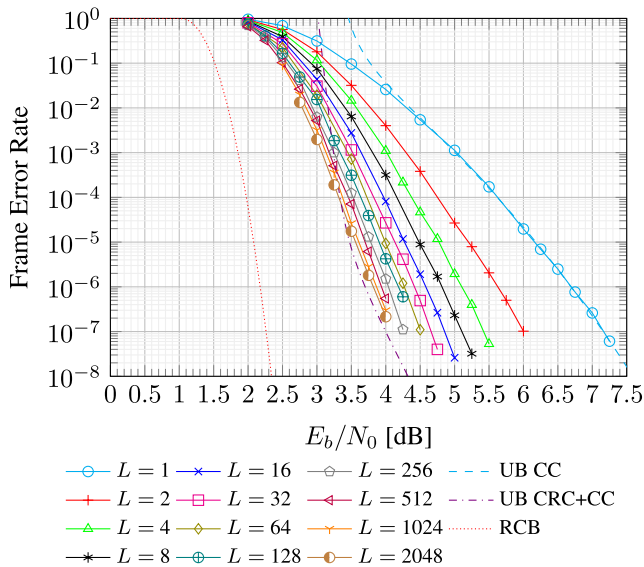
#### 1) PUNCTURED CODES

Results on the FER of several punctured codes are provided in Fig. 9 for  $R_{cc} = 2/3$ . (In the Appendix in the Fig. 16 and Fig. 17, we report the results for  $R_{cc} = 3/4$  and  $R_{cc} = 5/6$ , respectively).  $K = 1768$  is chosen as TF length. On the same chart, union bound on the FER under ML decoding are depicted as reference. The results are similar to the non punctured case and they show already coding gains w.r.t. the use of the VA that are larger than 2.5 dB at a FER of  $10^{-7}$ , for the CRC-aided list decoders when the maximum list size is 32. An interesting result worth to mention is that the coding loss of the CRC+CC code performance with respect to the RCB diminishes for increasing the code rate. For instance,





**FIGURE 8.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 1/2$ .

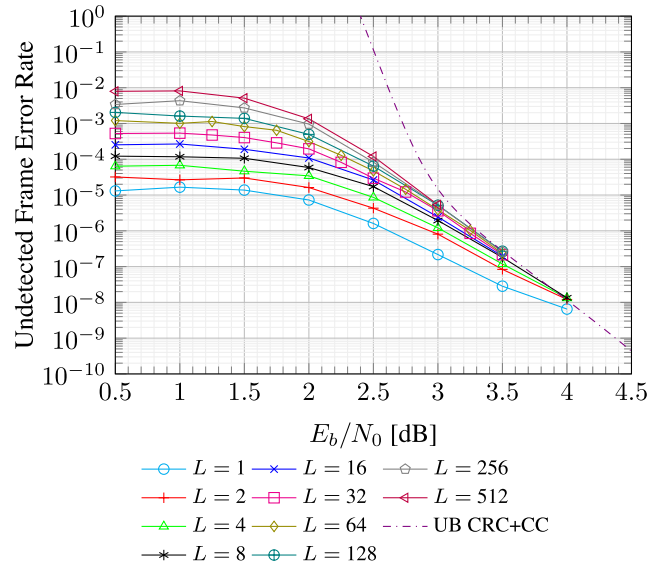


**FIGURE 9.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 2/3$ .

for a target FER of  $10^{-5}$ , for the  $R_{CC} = 1/2$  code the coding loss is of about 1.75 dB, while it decreases to 1.4 dB when  $R_{CC} = 2/3$ .

**B. UNDETECTED FRAME ERROR RATES**

While it is true that the CRC+CC under LVA decoding yields better error rates than the inner CC one under VA decoding, the gains come at the price of reducing the error-detecting capabilities of the system, thus increasing the UFER. For list decoders where the list contains the  $L$ -most likely candidates (this is the case of LVAs), the UFER of the system increases



**FIGURE 10.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the UFER as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 1/2$ .

as a function of the employed list size: larger list sizes mean larger UFERs. In the limit case in which the list contains all the codebook, the UFER coincides with the FER of the CRC+CC under ML decoding. It means that the UFER is always upper bounded by FER of the CRC+CC code.

We have investigated the UFER of the CC encoder of rate  $R_{CC} = 1/2$  for a TF of length  $K = 1768$  for various list sizes of CRC-aided LVAs via Monte-Carlo simulations. We have simulated up to counting 100 undetected errors for different values of  $E_b/N_0$ . Fig. 10 shows the obtained results which confirms that higher list sizes penalize the UFER, but the penalty decreases when the SNR increases. For instance, when the VA (which is equivalent to a LVA with  $L = 1$ ) is employed at the decoder side, the penalty at  $E_b/N_0 = 3.5$  dB is less than 0.5 dB from the truncated union bound on the FER of the CRC+CC code, while at  $E_b/N_0 = 4$  dB the margin from the bound is reduced to approximately 0.1 dB. This result suggests that the union bound on the FER of the CRC+CC scheme can be regarded as a relatively tight bound for low values of the UFER. Being the FER union bound an upper bound on the UFER of a LVA for all list sizes, this also means that at medium and high SNR there is little penalty in the UFER when using a LVA instead of the VA. This is true even for very large values of  $L$ .

In certain scenarios the required UFER may be very low. Although the constraints are much more pronounced for the uplink than for the downlink, let us suppose very extreme requirements like  $FER=10^{-9}$  and  $UFER=10^{-12}$ . Looking at Fig. 6 for a TF of length  $K = 1768$ , we have a FER of  $10^{-9}$  at around  $E_b/N_0 = 7.5$  dB with the VA, but the UFER of  $10^{-12}$  can already be reached at  $E_b/N_0 = 5.2$  dB, meaning that the use a LVA decoder can still provide more than 2 dB coding gain with respect to the VA decoding without exceeding the target UFER.

### C. COMPLEXITY OF THE ITERATIVE PLVA

In Fig. 11 we report the outcomes of the complexity of the application of the iterative PLVA algorithm. We use  $\bar{C}_{iPLVA}$  as complexity metric, which represents the average complexity normalized to the one of the VA. We indicate with  $L_{max}$  the maximum list size and we use as scheduler one which increase twofold the list size each time none of the codewords in the list satisfy the CRC code constraints.

To compute  $\bar{C}_{iPLVA}$ , we introduce the variable  $I$  that defines the number of iterations of the iterative PLVA. Then, owing to the fact that the complexity of the  $j$ -th iteration is  $L^{(j)} \times$  larger than the VA, we compute the normalized complexity as

$$\bar{C}_{iPLVA} = \mathbb{E}_I \left[ \sum_{j=1}^I L^{(j)} \right],$$

where the expectation is over  $I$ . We evaluate the expression via Monte Carlo simulations.

When examining the computed values of  $\bar{C}_{iPLVA}$  for a TF of length  $K = 1768$ , as illustrated in Fig. 11, and for various maximum list sizes  $L_{max}$ , it is apparent that at low  $E_b/N_0$  values,  $\bar{C}_{iPLVA}$  corresponds to the summation of all powers of two up to the employed maximum list size. This implies that, regardless of the list size, very few messages can be accurately decoded during the initial stages. However, for a target FER of  $10^{-7}$ , where  $E_b/N_0 > 4$  dB for ML decoding of the CRC+CC scheme, the average complexity of the iterative PLVA decoder approaches a value of 1 for all maximum list sizes  $\bar{C}_{iPLVA}$ , indicating that nearly all codewords in the concatenated scheme are correctly decoded during the first iteration, with only a small fraction requiring additional iterations. In particular, when focusing on Fig. 8 ( $K = 1768$ ), by looking at the FER results at  $E_b/N_0 = 4.5$  dB, the Viterbi decoder achieves a FER of  $2 \cdot 10^{-3}$ . This suggests that with the use of the iterative PLVA decoder, only approximately 0.2% of messages necessitate the execution of PLVA decoders with a list  $L \geq 2$ .

### VII. COMPARISON AGAINST THE OTHER CCSDS CHANNEL CODING OPTIONS

In this section we compare the performance of CCSDS CRC+CC codes under LVA decoding with the ones of other channel coding schemes for the CCSDS TM recommendation with similar length and rate. The comparison is performed on the AWGN channel with BPSK modulated signals. We use the CCSDS CRC+CC codes decoded using LVAs with  $L_{max} = 2048$  and compare them with the performance results reported in the CCSDS TM Green Book [32].

#### A. FRAME ERROR RATE

We report in Table 4 the rates of the various analyzed codes. The results are depicted in Fig. 12. The considered frame length is  $K = 1768$  bits.

The results of the (255, 223) Reed-Solomon (RS) codes concatenated with CC codes with an interleaver of length 5 blocks and each code is represented by RS+CC- $R_{CC}$ ,

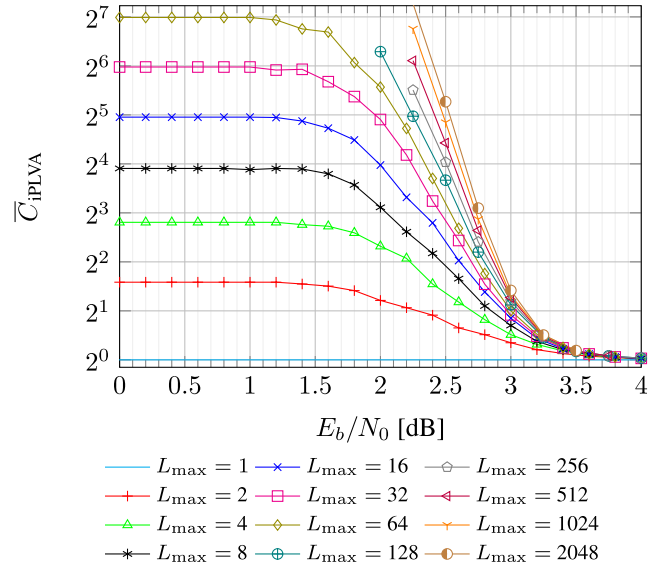


FIGURE 11. Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the  $\bar{C}_{iPLVA}$  as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 1/2$  which is decoded using the iterative PLVA with  $L_{max}$  in the legend.

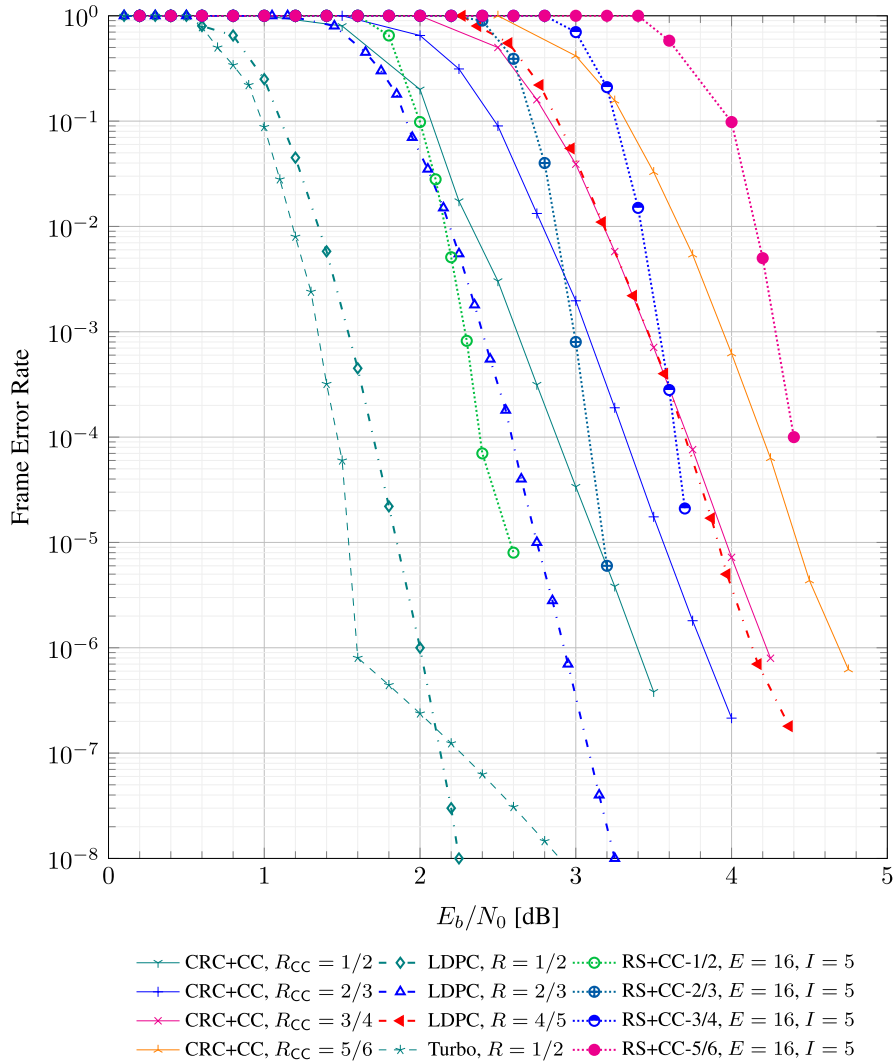
TABLE 4. Effective rates of the CCSDS channel coding options compared in Fig. 12.

code description	effective rate	input frame size
RS+CC-1/2	0.437	8920
Turbo code, $R = 1/2$	0.495	1768
CRC+CC, $R_{CC} = 1/2$	0.495	1768
LDPC code, $R = 1/2$	0.500	1024
RS+CC-2/3	0.583	8920
RS+CC-3/4	0.656	8920
CRC+CC, $R_{CC} = 2/3$	0.661	1768
LDPC code, $R = 2/3$	0.667	1024
RS+CC-5/6	0.729	8920
CRC+CC, $R_{CC} = 3/4$	0.743	1768
LDPC code, $R = 4/5$	0.800	1024
CRC+CC, $R_{CC} = 5/6$	0.826	1768

$E = 16, I = 5$ . Note that this code is more complex and longer than the CRC+CC on, with a frame size of 8920.

The LDPC codes used in the comparison have frame size of 1024 bits and they are iteratively decoded via belief propagation algorithm with 200 iterations, while the turbo code possesses a frame size of 1784 bits and it is decoded with 10 iterations of the BCJR algorithm.

Looking at Fig. 12, we can see that the coding gain over the CRC+CC code of the LDPC codes decreases when the code rate increases. For instance, looking at a FER of  $10^{-5}$ , at a code rate of  $\approx 1/2$  the coding gain is nearly 1.2 dB, while it reduces to 0.7 dB at a rate of around  $2/3$ , and it is of only 0.4 dB when the code rate is  $4/5$  for the LDPC code (and slightly higher for the CRC+CC code). An advantage of the CRC+CC codes to the LDPC codes is in their encoder, which can be realized with a simpler structure and it is the same for all rates, while the LDPC codes of the CCSDS TM recommendation possess a more complex encoder structure, with a different generator matrix for each rate-frame size pair.



**FIGURE 12.** Comparison of the transfer frame error rate for some channel coding options of the CCSDS versus the CRC+CC concatenations of the CCSDS TM recommendation decoded via LVAs with  $L_{\max} = 2048$  over the AWGN channel and BPSK modulation. The CRC+CC TFs have a length  $K = 1768$ .

When compared with the RS+CC codes, with similar rates, the CRC+CC codes perform slightly better while possessing shorter frames.

The rate 1/2 turbo code with  $K = 1768$  decreases its gain when the SNR increases due to the error floor phenomenon [33] because its minimum distance is 17, which is smaller than the CRC+CC one. Looking at a FER of  $10^{-5}$  the coding gain is of approximately 1.5 dB and it decreases down to 0.7 dB at a FER of  $10^{-10}$ .

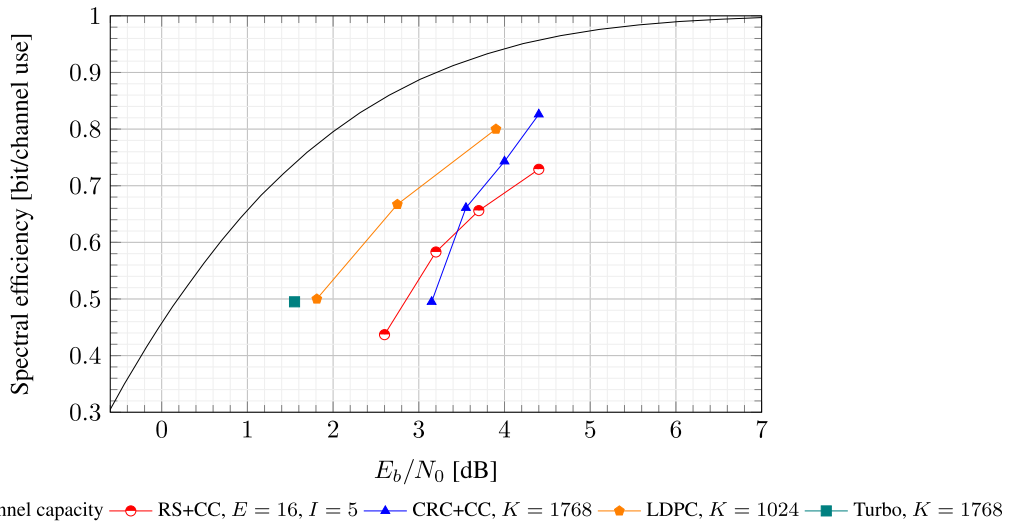
**B. DISTANCE FROM CAPACITY**

To summarize the results we have compared the spectral efficiency in bits per channel use for the various code families, compared them with the capacity of the binary input AWGN channel. For the CRC+CC codes we have taken into account the performance of a LVA with  $L_{\max} = 2048$ . The results are shown in Fig. 13 for the target FER of  $10^{-5}$ . From Fig. 13, we can see that by increasing the code rate, the gap

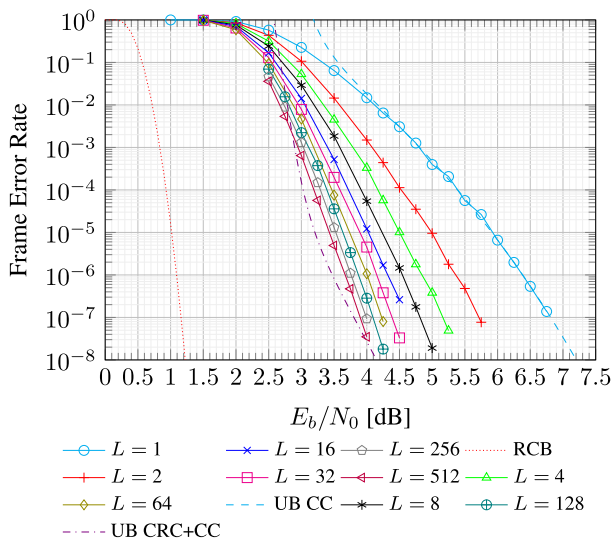
between the CRC+CC performance and the channel capacity limit decreases as the gap from LDPC codes.

**C. ENCODER COMPLEXITY**

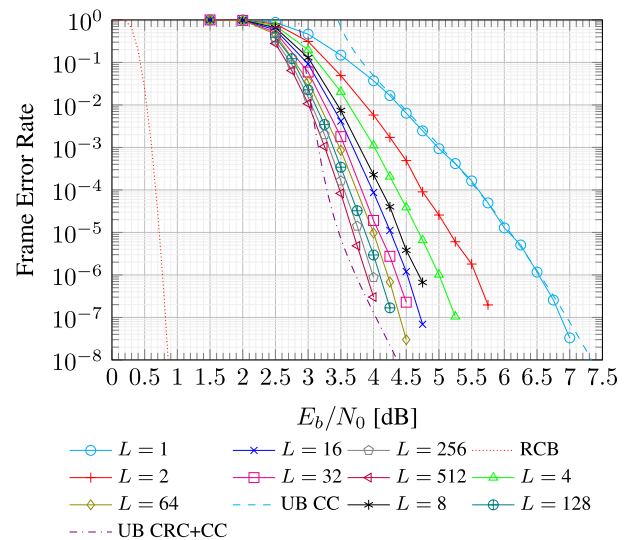
The CRC (Fig. 2) and CC (Fig. 3) encoders are shift registers, then their on-board implementation complexity is negligible. Moreover the same circuit is used for any code-rate, through puncturing application. The LDPC encoders are based on their generator matrices. The LDPC parity check matrices are sparse, but this is not the case for the generator matrices which are dense, then the number of elementary operations for the large CCSDS TM LDPC codes is not negligible, even when some simplifications obtained by exploiting their block-circulant structure are applied [34]. Moreover, the CCSDS LDPC codes require a different generator matrix for each code-rate, making the on-board complexity not negligible. As for turbo codes, the most critical part of implementing their encoder is the realization



**FIGURE 13.** Comparison of the spectral efficiency in bits per symbol for a target Transfer Frame Error Rate of  $10^{-5}$  for some coding options of the CCSDS TM recommendation versus the CRC+CC concatenations of the CCSDS TM recommendation decoded via LVAs with  $L_{max} = 2048$  over the AWGN channel with BPSK modulation.



**FIGURE 14.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 3552$  bits, employing a CC encoder with a code rate of  $R_{CC} = 1/2$ .



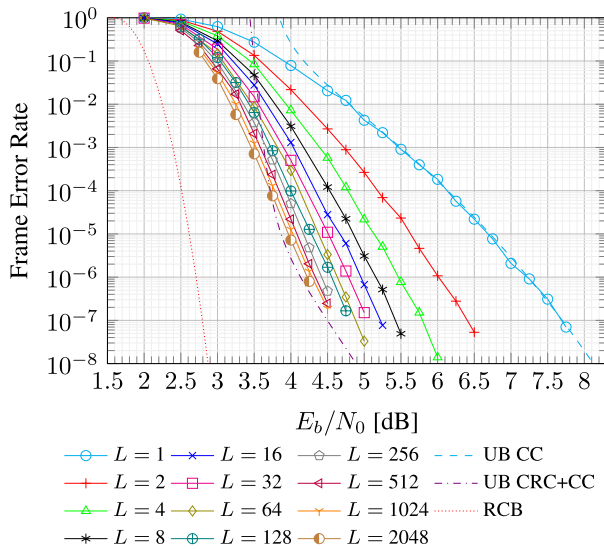
**FIGURE 15.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 8904$  bits, employing a CC encoder with a code rate of  $R_{CC} = 1/2$ .

of the long permutation that represents the interleaver, which involves a significant complexity that cannot be ignored. As a consequence, CRC+CC encoders are by far the best solution in case of on-board implementation for a satellite with limited resources.

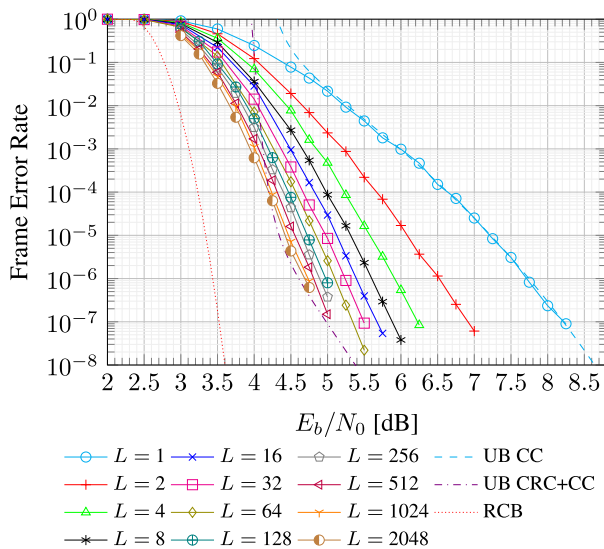
### VIII. CONCLUSION

In this study, we have investigated the application of list Viterbi decoding to the concatenation of CRC code with the inner convolutional code recommended by the CCSDS TM protocol. We have introduced a method for computing the concatenation's spectrum and employed it to derive union bounds on the maximum-likelihood FER.

Through the analysis of these bounds, we have investigated the additional coding gain achievable with CRC-aided list decoding compared to the decoding of the inner convolutional code by means of the simpler Viterbi algorithm. Our findings indicate that list decoding can potentially offer an extra coding gain of approximately 3 dB. The extra coding gain achievable by the CRC-aided list decoding with respect to the plain Viterbi decoding of the inner convolutional code has been analyzed by means of union bounds on the maximum-likelihood FER. Notably, this gain can be attained with moderate list sizes, leading to a minimal increase in average complexity when compared to the Viterbi algorithm. The use of list decoders slightly increases the undetected frame



**FIGURE 16.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 3/4$ .



**FIGURE 17.** Performance comparison of the CCSDS CRC+CC code using BPSK modulation over an AWGN channel. The graph showcases the FER as a function of the SNR. The evaluation focuses on a TF of length  $K = 1768$  bits, employing a CC encoder with a code rate of  $R_{CC} = 5/6$ .

error rate, but the increase is small at moderate SNRs and diminishes at higher SNRs. If the ground systems receivers of space missions that rely on convolutional codes are equipped to support list Viterbi decoders, a remarkable outcome can be achieved. The transmit power can be halved, introducing only a marginal increase in ground-based complexity, and with only a slight increment in the undetected frame error rates. This signifies the tremendous potential of list decoding techniques, which not only enable significant power savings but also maintain the undetected frame error rates at a nearly unchanged level. Such an approach promises substantial benefits in terms of energy efficiency and computational requirements for ground-based operations. As an alternative,

the extra-gain can be used to counter impairments such as additional interference, jamming, scintillation and others. When compared with other CCSDS coding options, the CRC-aided list decoding has some coding losses that decrease when the code-rate increases. However, the convolutional code solution with CRC-aided list decoding on ground highly simplifies the on-board transmission chain in both flexibility for the various code rates and very limited resource utilization for both the encoding and the controllers at the transmitter side, contrary to the higher complexity of LPDC encoders (large dense generator matrices, different for each code-rate) and turbo codes (interleaver permutation implementation). This makes the CRC+CC solution a very interesting candidate for new small satellite missions, in addition to providing the opportunity to significantly improve the performance of already functioning missions using convolutional codes.

**APPENDIX**  
**NUMERICAL RESULTS OF CRC-AIDED LIST VITERBI DECODING SYSTEMS**

This section presents additional outcomes concerning the FER analysis of the CRC+CC concatenation scheme applied to the CCSDS TM recommendation with CRC-aided list decoding. The investigation employs Monte Carlo simulations to evaluate the FER for different lengths denoted as  $K$  of the uncoded TF. The specific TF lengths examined are  $K \in \{1768, 3552, 8904\}$  bits, corresponding to standard CCSDS input lengths of 1784, 3568, and 8920 bits when considering the inclusion of the 16-bit CRC. The simulation assumes a BPSK modulated signal transmitted over an additive white Gaussian noise (AWGN) channel, with ideal frame and carrier synchronization at the receiver. The obtained results are presented in Figure 14 and Figure 15 for TF lengths of  $K = 3552$  and  $K = 8904$ , respectively, with a fixed code rate of  $R_{CC} = 1/2$ . Moreover, Figure 16 and Figure 17 depict the results for a fixed TF length of  $K = 1768$ , while employing code rates of  $R_{CC} = 3/4$  and  $R_{CC} = 5/6$ , respectively.

**ACKNOWLEDGMENT**

The authors would to thank Dr. Dariush Divsalar and Prof. William Ryan of the UCLA Communications Systems Laboratory group led by Richard D. Wesel, Dr. Andrea Modenini of ESA/ESTEC, and Prof. Monica Visintin of Politecnico di Torino for fruitful technical discussions on the topic.

An earlier version of this paper was presented in part at the 9th ESA International Workshop on Tracking, Telemetry and Command Systems for Space Applications (TT&C), 2022 [DOI: 10.1109/TTC55771.2022.9975785].

**REFERENCES**

- [1] R. Schiavone, R. Garello, and G. Liva, "Application of list Viterbi algorithms to improve the performance in space missions using convolutional codes," in *Proc. 9th Int. Workshop Tracking, Telemetry Command Syst. Space Appl. (TTC)*, Nov. 2022, pp. 1–8.
- [2] *Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD)*, document TS 125 212, V10.1.0, May 2011.
- [3] *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Multiplexing and Channel Coding*, document TS 136 212, V17.1.0, Apr. 2022.

- [4] C. Lou, B. Daneshrad, and R. D. Wesel, "Convolutional-code-specific CRC code design," *IEEE Trans. Commun.*, vol. 63, no. 10, pp. 3459–3470, Oct. 2015.
- [5] N. Seshadri and C.-E.-W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 313–323, Feb. 1994.
- [6] K. R. Narayanan and G. L. Stuber, "List decoding of turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 6, pp. 754–762, Jun. 1998.
- [7] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2213–2226, May 2015.
- [8] 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.
- [9] H. Yang, E. Liang, and R. D. Wesel, "Joint design of convolutional code and CRC under serial list Viterbi decoding," 2018, *arXiv:1811.11932*.
- [10] H. Yang, S. V. S. Ranganathan, and R. D. Wesel, "Serial list Viterbi decoding with CRC: Managing errors, erasures, and complexity," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.
- [11] H. Yang, E. Liang, H. Yao, A. Vardy, D. Divsalar, and R. D. Wesel, "A list-decoding approach to low-complexity soft maximum-likelihood decoding of cyclic codes," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
- [12] E. Liang, H. Yang, D. Divsalar, and R. D. Wesel, "List-decoded tail-biting convolutional codes with distance-spectrum optimal CRCS for 5G," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.
- [13] H. Yang, L. Wang, V. Lau, and R. D. Wesel, "An efficient algorithm for designing optimal CRCS for tail-biting convolutional codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 292–297.
- [14] R. Schiavone, "Channel coding for massive IoT satellite systems," M.S. thesis, Dept. Electron. Telecommun., Politecnico di Torino, Turin, Italy, 2021.
- [15] W. Sui, H. Yang, B. Towell, A. Asmani, and R. D. Wesel, "High-rate convolutional codes with CRC-aided list decoding for short blocklengths," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 98–103.
- [16] J. King, A. Kwon, H. Yang, W. Ryan, and R. D. Wesel, "CRC-aided list decoding of convolutional and polar codes for short messages in 5G," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 92–97.
- [17] J. King, W. Ryan, and R. D. Wesel, "CRC-aided short convolutional codes and RCU bounds for orthogonal signaling," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 4256–4261.
- [18] L. Wang, D. Song, F. Areces, and R. D. Wesel, "Achieving short-blocklength RCU bound via CRC list decoding of TCM with probabilistic shaping," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 2906–2911.
- [19] D. Song, F. Areces, L. Wang, and R. Wesel, "Shaped TCM with list decoding that exceeds the RCU bound by optimizing a union bound on FER," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 4262–4267.
- [20] L. Wang, D. Song, F. Areces, T. Wiegart, and R. D. Wesel, "Probabilistic shaping for trellis-coded modulation with CRC-aided list decoding," *IEEE Trans. Commun.*, vol. 71, no. 3, pp. 1271–1283, Mar. 2023.
- [21] H. Yang, E. Liang, M. Pan, and R. D. Wesel, "CRC-aided list decoding of convolutional codes in the short blocklength regime," *IEEE Trans. Inf. Theory*, vol. 68, no. 6, pp. 3744–3766, Jun. 2022.
- [22] *TM Synchronization and Channel Coding*, Standard 131.0-B-4, CCSDS, Apr. 2022.
- [23] *TM Space Data Link Protocol*, Standard 132.0-B-3, CCSDS, Oct. 2021.
- [24] *AOS Space Data Link Protocol*, Standard 732.0-B-4, CCSDS, Oct. 2021.
- [25] F. K. Soong and E.-F. Huang, "A tree-trellis based fast search for finding the N-best sentence hypotheses in continuous speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Toronto, ON, Canada, vol. 1, 1991, pp. 705–708, doi: 10.1109/ICASSP.1991.150437.
- [26] M. Roder and R. Hamzaoui, "Fast tree-trellis list Viterbi decoding," *IEEE Trans. Commun.*, vol. 54, no. 3, pp. 453–461, Mar. 2006.
- [27] D. E. Knuth, *The Art of Computer Programming*, vol. 3, 2nd ed. Reading, MA, USA: Addison-Wesley, 1998.
- [28] K. E. Batcher, "Bitonic Sorting," Goodyear Aerosp. Corp., Akron, OH, USA, Tech. Rep., GER-11869, 1964.
- [29] 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.
- [30] M. P. C. Fossorier, S. Lin, and D. J. Costello, "On the weight distribution of terminated convolutional codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1646–1648, Jul. 1999.
- [31] R. G. Gallager, *Information Theory and Reliable Communication*. Hoboken, NJ, USA: Wiley, 1968.
- [32] *TM Synchronization and Channel Coding—Summary of Concept and Rationale*, Standard 130.1-G-3, CCSDS, Jun. 2020.
- [33] R. Garello, P. Pierleoni, and S. Benedetto, "Computing the free distance of turbo codes and serially concatenated codes with interleavers: Algorithms and applications," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 800–812, May 2001.
- [34] R. Abelló, M. Baldi, F. Carvalho, F. Chiaraluce, R. Fernandes, R. Garello, E. Paolini, and R. Prata, "Next generation earth-to-space telecommand coding and synchronization: Ground system design, optimization and software implementation," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–22, Dec. 2021.



with additional expertise in binarized neural networks.

**RICCARDO SCHIAVONE** received the dual degree in communications and engineering from Télécom Paris (EURECOM) and Politecnico di Torino. He is currently pursuing the Ph.D. degree in electrical, electronics and communications engineering with Politecnico di Torino. He has extensive experience in satellite communication standards and digital signal processing algorithms and real-time implementations. His research interests include synchronization and channel coding,



Department of Electronics and Telecommunications, Politecnico di Torino. In 2017, he was an Adjunct Professor with California State University, Los Angeles. His research interests include space communication systems, 5G and beyond mobile networks, and channel coding. On these topics, he has coauthored more than 150 papers. He has been the project manager of 40 research projects and an advisor of 12 Ph.D. students.

**ROBERTO GARELLO** (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Politecnico di Torino, in 1994. His Ph.D. thesis on error correction coding. During the Ph.D. degree, he was a Visiting Student with MIT, Cambridge, and ETH, Zürich. From 1994 to 1997, he was with Marconi Communications, Genoa. From 1998 to 2001, he was an Associate Professor with the University of Ancona. Since November 2001, he has been an Associate Professor with the



University of Munich (TUM). From 2012 to 2013, he was a Lecturer in channel coding with the Nanjing University of Science and Technology, China. Since 2014, he has been a Lecturer in channel codes with iterative decoding with LNT, TUM. His research interests include satellite communications, random access techniques, and error control coding. Since 2020, he has been serving as an Associate Editor for *Coding and Information Theory* for the IEEE TRANSACTIONS ON COMMUNICATIONS.

**GIANLUIGI LIVA** (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from the University of Bologna, Italy, in 2002 and 2006, respectively. Since 2006, he has been with the Institute of Communications and Navigation, German Aerospace Center (DLR), where he is currently leads the Information Transmission Group. In 2010, he was appointed as a Lecturer in channel coding with the Institute for Communications Engineering (LNT), Technical University of Munich (TUM). From 2012 to 2013, he was a Lecturer in channel coding with the Nanjing University of Science and Technology, China. Since 2014, he has been a Lecturer in channel codes with iterative decoding with LNT, TUM. His research interests include satellite communications, random access techniques, and error control coding. Since 2020, he has been serving as an Associate Editor for *Coding and Information Theory* for the IEEE TRANSACTIONS ON COMMUNICATIONS.