Extrinsic Information Transfer Analysis and Design of Block-Based Intermediate Codes

Robert G. Maunder, *Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Intermediate codes have been shown to facilitate iterative decoding convergence to the maximum-likelihood (ML) error ratio performance in serially concatenated schemes. In this paper, we propose a novel *block-based* intermediate code as an alternative to classic *convolutional* intermediate codes. Because it is block-based, our intermediate code facilitates practical implementations with reduced memory and processing requirements. Furthermore, we demonstrate that it is simpler to analyze and optimize the iterative decoding process when our block-based intermediate code is employed, instead of a convolutional intermediate code. Finally, we demonstrate that the proposed block-based intermediate code facilitates significantly reduced error ratios in practical schemes when employing short transmission frames and a limited decoding complexity.**

***Index Terms*—Block codes, combined source channel coding, iterative decoding, modulation coding, mutual information.**

I. Introduction

IT HAS been shown [1] that it is a channel encoder's distance spectrum that dictates the error ratio performance that can be achieved by the corresponding optimal maximum-likelihood (ML) decoder but at a potentially excessive computational complexity. However, at sufficiently high-channel signal-to-noise ratios (SNRs), certain iterative decoders [2]–[4] can approach the ML error ratio performance at a fraction of the ML decoder's complexity [5]. This condition is achieved when the mutual information (MI) associated with the extrinsic logarithmic likelihood ratios (LLRs) that are iteratively exchanged by the receiver's concatenated component decoders reaches the maximum possible value of one [6].

However, to facilitate iterative decoding convergence toward the maximal mutual information (MMI), the corresponding extrinsic information transfer (EXIT) chart [6] is required to exhibit an open tunnel. One sufficient condition for this case to be created is satisfied if it is particularly supported by at least two of the concatenated component decoders. Suitable component decoders may be termed as being maximal mutual information achieving (MMIA), because they generate extrinsic LLRs with MMI when provided with MMI *a priori* LLRs. One condition for a decoder to be MMIA is if it satisfies the following two requirements: 1) It exchanges extrinsic LLRs that pertain to a sequence of bits with legitimate permutations that are separated by Hamming distances of at least two,[1] and 2) it is aware of this fact and therefore exploits it[2] [7]. Alternatively, a decoder satisfies the second sufficient condition[3] of being MMIA if satisfies the following three requirements: 1) It exchanges extrinsic LLRs that pertain to the corresponding encoder's input bit sequence; 2) the corresponding encoder is recursive; and 3) the corresponding encoder is terminated into a known state [8], [9].

Conversely, iterative decoding convergence toward the ML error ratio performance is prevented when less than two of the concatenated component decoders are MMIA, which is the case for iteratively decoded bit-interleaved coded modulation (BICM-ID) [10], for example, which employs a serial concatenation [11], [12] of an outer decoder and an inner demodulator. Although the first MMIA condition previously described is typically satisfied by the outer decoder of the BICM-ID scheme, its demodulator typically satisfies none of the aforementioned MMIA conditions.[4] Furthermore, this property is shared by many other detectors, despreaders, and equalizers. In this case, a recursive convolutional code can be employed as an *intermediate code* between the arbitrary pairing of outer decoder and inner demodulator to exploit the second MMIA condition and, hence, to facilitate iterative decoding convergence toward the ML error ratio performance [4], [13]–[16].

However, a number of disadvantages are associated with convolutional intermediate codes. First, convolutional intermediate decoding requires a significant amount of processing and memory, because this approach is typically achieved by applying the relatively complex Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [17]. Furthermore, due to the data dependencies and recursive nature of BCJR, convolutional intermediate decoders

Manuscript received July 23, 2010; revised November 9, 2010; accepted January 10, 2011. Date of publication January 17, 2011; date of current version March 21, 2011. This work was supported in part by Research Councils U.K. through the India–U.K. Advanced Technology Centre and the China–UK Science Bridge on Fourth-Generation Wireless Communications. The review of this paper was coordinated by Prof. G. Bauch.

The authors are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ Southampton, U.K. (e-mail: rm@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/TVT.2011.2106809

[1]Note that a parallel concatenation of two convolutional decoders exchanges extrinsic LLRs that pertain to a sequence of uncoded bits, which has legitimate permutations that are separated by Hamming distances of as low as one. Therefore, these component decoders do not satisfy this sufficient condition.

[2]Note that a serial concatenation of two convolutional decoders exchanges extrinsic LLRs that pertain to a sequence of encoded bits, which has legitimate permutations that are separated by Hamming distances of at least two. However, the inner decoder does not exploit this fact; therefore, it does not satisfy this sufficient condition.

[3]This case is the sufficient condition that can be satisfied by parallel concatenated convolutional decoders and the inner one of two serially concatenated convolutional decoders.

[4]Although the demodulator exchanges extrinsic LLRs that pertain to a sequence of encoded bits, it does not exploit the fact that the legitimate permutations are separated by Hamming distances of at least two.

are less amenable to parallel or pipelined implementations,[5] which facilitate high-processing throughput. Furthermore, convolutional intermediate codes have long error events [20], which are limited only by the length of the BCJR trellis. As a result, it is challenging to design interleavers for directly eliminating specific combinations of error events that yield a poor distance spectrum and, hence, erode the ML decoding performance [21]. Finally, the analysis of the iterative decoding convergence of a three-stage serial concatenation that employs a convolutional intermediate code requires the consideration of two separate 3-D EXIT charts [4], one of which provides the outer decoder's EXIT function, whereas the inner decoder's EXIT function is plotted in the other chart. As a result, the visualization and interpretation of the iterative decoding trajectory is less intuitive than it is for two-stage concatenations [6]. This case obfuscates the optimization of irregular near-capacity serial concatenations [22], for example.

Against this background, we extend the principles of systematic repeat accumulate (SRA) codes [23] to create a novel block-based intermediate code in Section II. In contrast to convolutional intermediate codes, our design facilitates parallel-processing- and pipelining-based implementations, as well as having reduced memory requirements and computational complexities. In addition, due to its block-based nature, the proposed intermediate code results in short error patterns, which simplifies the interleaver design process. As shown in Section III, the iterative decoding process, which employs the proposed block-based intermediate code, can be characterized using a single 3-D EXIT chart. Hence, our approach facilitates simpler serial concatenated design and optimization while providing a more intuitive visualization and interpretation of the iterative decoding trajectory than classic convolutional intermediate codes. Furthermore, in Section IV, we demonstrate that the proposed block-based intermediate code facilitates operation at lower channel SNRs than convolutional intermediate codes invoked in practical schemes, which employ short frames and impose a limited decoding complexity. Finally, we offer our conclusions in Section V.

II. Block-Based Intermediate Code Operation

In this section, we detail our novel block-based intermediate code. We begin by showing how this approach may serially be concatenated with outer and inner codes in Section II-A. Then, the operation of the block-based intermediate code in the transmitter and receiver is detailed in Sections II-B and C, respectively.

[5]Note that, rather than using the BCJR, a convolutional intermediate decoder may be implemented using an iterative operation of suitably arranged variable and check nodes [18]. Although this approach facilitates a high level of parallel processing and pipelining, it has the disadvantage of significantly increasing the decoder's computational complexity. Similarly, the sliding-window algorithm in [19] may be used to increase the achievable level of parallel processing and pipelining. However, this approach also significantly increases the decoder's computational complexity and potentially reduces the achievable performance. Because our intention is to *reduce* the complexity of intermediate codes, these increased-complexity variations of convolutional intermediate codes are not considered any further in this paper.

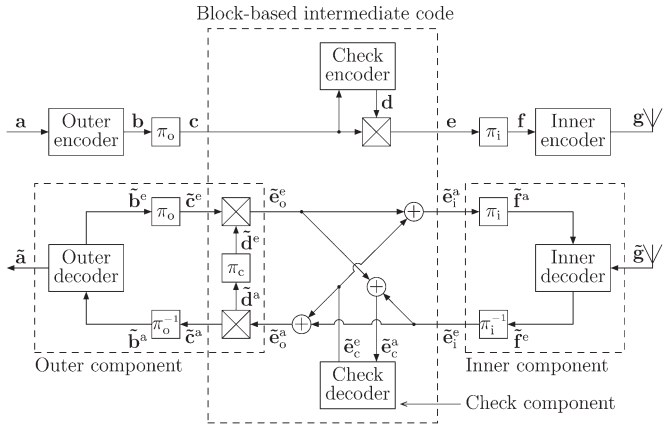


Fig. 1. Schematic of the proposed block-based intermediate code and its serial concatenation with inner and outer codes

A. Schematic

The schematic in Fig. 1 illustrates the interactions that take place within the proposed block-based intermediate code, as well as its interactions with the inner and outer codes. Here, the outer code is assumed to satisfy the first MMIA condition in the Section I. By contrast, we assume that a non-MMIA inner code is employed, such as a modulation, spreading, or equalization scheme. Therefore, the MMIA block-based intermediate code is required to facilitate iterative decoding convergence toward the ML error ratio performance, without requiring the modification of the arbitrary outer and inner codes.

In Fig. 1, multiplexing and demultiplexing are indicated by the crossed blocks, whereas interleaving is performed by the blocks that contain the symbol "π," where the superscript "-1" denotes deinterleaving, and the subscripts "o," "c," and "i" indicate outer, check, and inner interleaving, respectively. In the transmitter in Fig. 1, the bit sequences $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, $\mathbf{e}$, and $\mathbf{f}$ are generated when transforming the source sequence $\mathbf{a}$ into the sequence $\mathbf{g}$, which is modulated and transmitted over the channel. The channel introduces uncertainty about the transmitted bit values, which is expressed using LLRs in the receiver in Fig. 1. Here, the LLR sequence that pertains to a particular sequence of bits is indicated by including a diacritical tilde in the corresponding notation. Furthermore, the superscripts "a" and "e" indicate *a priori* extrinsic information, respectively, whereas the subscripts "o," "c," and "i" indicate their relevance to the outer, check, and inner decoders, respectively.

In Fig. 1, the sequences $\{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}\}$ (as well as the corresponding LLR sequences) have lengths of $\{N_{\mathbf{a}}, N_{\mathbf{b}}, N_{\mathbf{c}}, N_{\mathbf{d}}, N_{\mathbf{e}}, N_{\mathbf{f}}, N_{\mathbf{g}}\}$, which are related to each other. More specifically, we have $N_{\mathbf{e}} = N_{\mathbf{c}} + N_{\mathbf{d}} = N_{\mathbf{c}}/R_{\mathrm{c}}$, where $N_{\mathbf{d}}$ is odd, and R_{c} is the coding rate of the block-based intermediate code. In addition, because the interleavers π_{o} and π_{i} merely rearrange the *order* of their input bit sequences, we have $N_{\mathbf{c}} = N_{\mathbf{b}}$ and $N_{\mathbf{f}} = N_{\mathbf{e}}$.

B. Transmitter

The check encoder in Fig. 1 decomposes the bit sequence $\mathbf{c}$ into $(N_{\mathbf{d}} - 1)/2$ similar-length subsequences, $\{\mathbf{c}_j\}_{j=1}^{(N_{\mathbf{d}}-1)/2}$ before generating the bit sequence $\mathbf{d} = \{d_i\}_{i=1}^{N_{\mathbf{d}}}$, as detailed in the following discussion. Then, the multiplexer in Fig. 1 obtains

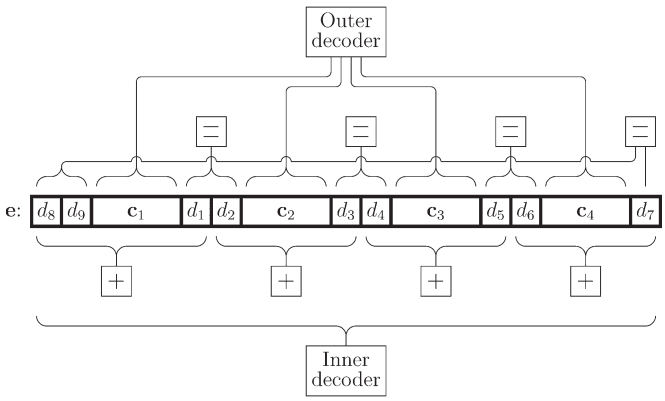


Fig. 2. Example composition of the bit sequence $\mathbf{e}$ from the sequences $\mathbf{d} = \{d_i\}_{i=1}^{N_\mathbf{d}}$ and $\mathbf{c} = \{\mathbf{c}_j\}_{j=1}^{(N_\mathbf{d}-1)/2}$ for the case where $\mathbf{d}$ and $\mathbf{c}$ comprise $N_\mathbf{d} = 9$ and $N_\mathbf{c} \geq 4$ b, respectively. Braces are used to indicate for which bits each receiver component can generate extrinsic information, where the boxed equal and plus signs represent the check interleaver π_c and the check decoder, respectively.

the bit sequence $\mathbf{e}$ by interspersing the subsequences of $\mathbf{c}$ and the bits of $\mathbf{d}$ in the order exemplified in Fig. 2, where we have $N_\mathbf{d} = 9$ and $N_\mathbf{c} \geq 4$.

Check encoding commences by setting the first bit d_1 in the sequence $\mathbf{d}$ to the modulo-two sum of all bits in the subsequence $\mathbf{c}_1$. As a result, the concatenation of $\mathbf{c}_1$ and the checksum bit d_1 has an even Hamming weight. Next, the second bit d_2 in the sequence $\mathbf{d}$ is set to the value of the preceding bit, i.e., to d_1. The case that d_1 and d_2 have equal values is indicated by the corresponding boxed equal sign in Fig. 2. Similar to the first bit d_1 in the sequence $\mathbf{d}$, the third bit d_3 is set equal to a modulo-two sum, but in this case, the modulo-two sum of all bits in $\mathbf{c}_2$ and the preceding bit d_2 is employed. As a result, the concatenation of d_2, $\mathbf{c}_2$, and d_3 has an even Hamming weight, as indicated by the corresponding boxed plus sign in Fig. 2.

The aforementioned procedure for generating d_2 and d_3 is also adopted for all but the final two of the remaining bits in $\mathbf{d}$, i.e., for $\{d_i\}_{i=4}^{N_\mathbf{d}-2}$, where $N_\mathbf{d}$ is odd. More generally, the bits of $\mathbf{d}$ with even indices $i \in \{2, 4, 6, \ldots, N_\mathbf{d} - 3\}$ are set equal[6] to the preceding bit d_{i-1}, as indicated by the corresponding boxed equal signs in Fig. 2. By contrast, the bits of $\mathbf{d}$ with odd indices $i \in \{3, 5, 7, \ldots, N_\mathbf{d} - 2\}$ are set equal to the modulo-two sum[7] of the preceding bit d_{i-1} and all bits in $\mathbf{c}_{(i+1)/2}$. This approach creates blocks with even Hamming weights, as indicated by the corresponding boxed plus signs in Fig. 2.

In the transmitter in Fig. 1, check encoding is completed by assigning the value of $d_{N_\mathbf{d}-2}$ to the final two bits of $\mathbf{d}$, i.e., to $d_{N_\mathbf{d}-1}$ and $d_{N_\mathbf{d}}$. The case that $d_{N_\mathbf{d}-2}$, $d_{N_\mathbf{d}-1}$ and $d_{N_\mathbf{d}}$ have equal values is indicated by the corresponding boxed equal sign in Fig. 2. Recall from the previous discussion that the concatenation of $\mathbf{c}_1$ and d_1 has an even Hamming weight. Although it may be increased, this weight remains even valued when additionally concatenating $d_{N_\mathbf{d}-1}$ and $d_{N_\mathbf{d}}$, because they have equal bit values.[8] This condition is indicated by the corresponding boxed plus sign in Fig. 2.

C. Receiver

The receiver in Fig. 1 employs an iterative exchange of LLRs that pertain to the bit sequence $\mathbf{e}$ between three decoding components, i.e., the outer, check, and inner components, as shown in Fig. 1. For example, the inner component converts the *a priori* LLR sequence $\tilde{\mathbf{e}}_\mathrm{i}^\mathrm{a}$ into the extrinsic LLR sequence $\tilde{\mathbf{e}}_\mathrm{i}^\mathrm{e}$ in its normal manner.

Within the outer component, the check interleaver π_c cooperates with the outer decoder to convert the *a priori* LLR sequence $\tilde{\mathbf{e}}_\mathrm{o}^\mathrm{a}$ into the extrinsic LLR sequence $\tilde{\mathbf{e}}_\mathrm{o}^\mathrm{e}$, as shown in Fig. 1. The MMIA outer decoder converts the subset $\tilde{\mathbf{c}}^\mathrm{a}$ of $\tilde{\mathbf{e}}_\mathrm{o}^\mathrm{a}$ into the extrinsic LLR sequence $\tilde{\mathbf{c}}^\mathrm{e}$ in its normal manner, whereas the check interleaver π_c converts the remaining *a priori* LLRs $\tilde{\mathbf{d}}^\mathrm{a}$ into $\tilde{\mathbf{d}}^\mathrm{e}$. This approach is achieved by swapping the position of the *a priori* LLRs in $\tilde{\mathbf{d}}^\mathrm{a}$, which corresponds to each pair of bits that is identified by a boxed equal sign in Fig. 2. For example, $\tilde{d}_3^\mathrm{e} = \tilde{d}_4^\mathrm{a}$, and $\tilde{d}_4^\mathrm{e} = \tilde{d}_3^\mathrm{a}$. Note that, for the triplet of bits $d_{N_\mathbf{d}-2}$, $d_{N_\mathbf{d}-1}$, and $d_{N_\mathbf{d}}$ that is identified by a boxed equal sign in Fig. 2, each of the corresponding extrinsic LLRs is generated as the sum of the other two *a priori* LLRs. For example, $\tilde{d}_{N_\mathbf{d}-1}^\mathrm{e} = \tilde{d}_{N_\mathbf{d}-2}^\mathrm{a} + \tilde{d}_{N_\mathbf{d}}^\mathrm{a}$. Aside from these three additions, the check interleaver π_c has a negligible complexity, otherwise requiring only LLR interleaving. Note that the check interleaver π_c is block based and satisfies the first MMIA condition provided in Section I, because it obtains extrinsic information for blocks of repeated bits in $\mathbf{d}$ by exploiting the case that their permutations are separated by Hamming distances of at least two. Because both the check interleaver π_c and the outer decoder are MMIA, the outer decoding component in Fig. 1 is MMIA.

Similarly, the check decoder can independently generate extrinsic information for each block of bits in the sequence $\mathbf{e}$ that is indicated by the boxed plus signs in Fig. 2. In this block-based regime, the check decoder employs the forward–backward algorithm [24] to consider the *a priori* LLRs of $\tilde{\mathbf{e}}_\mathrm{c}^\mathrm{a}$ and to generate the extrinsic LLR sequence $\tilde{\mathbf{e}}_\mathrm{c}^\mathrm{e}$, as shown in Fig. 1. Note that the check decoder satisfies the first MMIA condition, because it obtains extrinsic information for blocks of repeated bits in $\mathbf{e}$ by exploiting the case that their permutations are separated by even Hamming distances, which must

[6]It is necessary to repeat each bit in $\mathbf{d}$ so that the check interleaver π_c of Fig. 1 becomes MMIA, as discussed in Section II-C.

[7]Note that the encoding process in Fig. 2 is similar to a serial concatenation, which comprises the outer code, an SRA code [23], and the inner code. Similar to SRA codes, the proposed block-based intermediate code is systematic, because the bits of $\mathbf{c}$ are directly contained by the intermediate encoder's output bit sequence $\mathbf{e}$. Furthermore, the other bits in $\mathbf{e}$ are provided by the accumulated checksums of $\mathbf{d}$, as in SRA codes. However, in contrast with SRA codes, the bits of $\mathbf{c}$ are not repetition encoded and interleaved before being accumulated. Instead, these roles are fulfilled by the outer encoder and interleaver in the encoding process in Fig. 2, therefore removing unnecessary components from the scheme and reducing its complexity. As a result, the receiver in Fig. 1 requires the iterative operation of only three decoding components, *none of which are required to be convolutional*, as will be detailed in Section II-C. By contrast, the receiver for a serial concatenation of the outer, SRA, and inner codes would require the iterative operation of the following five decoding components: 1) the outer decoder; 2) the inner decoder; 3) the SRA repetition decoder; 4) the SRA check decoder; and 5) the SRA *convolutional* decoder [23]. Because our intention is to *reduce* the complexity of intermediate codes, the use of SRA codes in this role is not considered any further in this paper.

[8]Note that, although $d_{N_\mathbf{d}-1}$ and $d_{N_\mathbf{d}}$ have no effect upon the checksum d_1, they affect the encoded bit sequence $\mathbf{g}$, because they are interleaved into the inner encoder's input bit sequence $\mathbf{f}$.

be at least two. In the investigations that will be detailed in Section IV, we found that the check decoder typically invokes approximately 50% fewer add, compare, and select (ACS) operations than even the simplest of convolutional intermediate codes, which has a coding rate of unity and a single memory element. In this comparison, all calculations invoked by the forward–backward and BCJR algorithms were performed in the logarithmic domain by using an eight-entry lookup table to correct the Jacobian approximation [25].

Observe in Fig. 1 that the *a priori* LLR sequences provided for each of the outer, check, and inner decoding components is obtained as the sum of the extrinsic LLR sequences generated by the other two components $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}} = \tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}} + \tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}}$, $\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{a}} = \tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}} + \tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}}$, and $\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}} = \tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}} + \tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}}$. As a result, the three components can iteratively be operated using any component activation order, provided that the same component is not activated twice in a row. This way, the iterative decoding gradually obtains much more extrinsic information until convergence is achieved, whereupon it becomes impossible to glean any more information. As described in the Section I, convergence toward the ML error ratio performance is facilitated when the channel SNR is sufficiently high, because two of the decoding components are MMIA, i.e., the outer and check components. This convergence is characterized in Section III.

III. Analysis of Iterative Decoding Convergence

Let us now analyze the iterative decoding convergence of the scheme introduced in Section III-A, which employs the proposed block-based intermediate code in Fig. 1. Section III-B shows that this analysis can be achieved using a single 3-D EXIT chart rather than requiring two charts. As a result, the design and optimization of irregular near-capacity serially concatenated schemes, as well as the visualization and interpretation of the iterative decoding trajectory, is more intuitive when employing our proposed block-based intermediate code instead of a convolutional intermediate code. Furthermore, in Section III-C, we show that this 3-D EXIT chart can be projected into two dimensions, offering some insight into the proposed scheme's ability to facilitate near-capacity operation.

A. Scenario

In this section, we introduce a typical scenario where the employment of the proposed block-based intermediate code is particularly motivated. More specifically, in the scenario considered, the outer and inner codes cannot be replaced by a pair of MMIA codes, because they are required to perform particular source coding and modulation functions, respectively. Furthermore, both the outer and the inner code are block based, motivating the employment of a block-based intermediate code. Note, however, that the proposed block-based intermediate code can readily be applied in other scenarios, employing different outer and inner codes.

During our investigations, 16-ary symbols were employed for the source sequence $\mathbf{a} = \{a_i\}_{i=1}^{N_{\mathbf{a}}}$ in Fig. 1. We used symbols with values of $a_i \in \{0, 1, 2, \ldots, 15\}$ that occur with unequal probabilities of $\{P(a_i = k)\}_{k=0}^{15} = \{0.00818, 0.02445, 0.04272, 0.06047, 0.07615, 0.08872, 0.09743, 0.1019, 0.1019, 0.09743, 0.08872, 0.07615, 0.06047, 0.04272, 0.02445, 0.00818\}$. These probabilities of occurrence yield an entropy of $E = 3.77$ b per 16-ary symbol and result from the Lloyd–Max quantization [26] of Gaussian distributed source samples. Note that, in the receiver, the unequal probabilities of occurrence can be exploited to assist iterative decoding. This approach motivates the use of a joint source/channel decoder for the outer decoder of the iterative decoding process in Fig. 1 [27].

A fixed-length code (FLC) was selected for the outer joint source/channel code. We elected to employ FLC codewords with lengths of $L_{\mathrm{o}} = 5$ b, which is sufficient for maintaining Hamming distances of at least $d_{\mathrm{o}} = 2$ and to satisfy the first MMIA condition in the Section I. The FLC encoder maps the 16-ary source symbol values $a_i \in \{0, 1, 2, \ldots, 15\}$ to binary FLC codewords, which may be expressed in octal form as {00, 35, 30, 05, 17, 14, 11, 12, 21, 22, 27, 24, 36, 03, 06, 33}. This mapping was designed by considering the source symbol occurrence probabilities to maximize the Hamming distance between the most likely codewords while allowing lower distances between less likely codewords. Note that the described FLC has a coding rate of $R_{\mathrm{o}} = E/L_{\mathrm{o}} = 0.754$.

Furthermore, we elected to employ set-partitioned $M_{\mathrm{i}} = 16$-ary quadrature amplitude modulation (16-QAM) [28] for the inner code in Fig. 1. Because a 16-QAM demodulator is not MMIA, the inclusion of the block-based intermediate code is motivated to facilitate iterative decoding convergence toward the ML symbol error ratio (SER) performance. A coding rate of $R_{\mathrm{c}} = 5/6$ was selected for our block-based intermediate code, because this approach will facilitate a fair comparison with a benchmarker that will employ a convolutional intermediate code in Section IV. The effective throughput of the scheme considered is therefore $\eta = R_{\mathrm{o}} \cdot R_{\mathrm{c}} \cdot \log_2(M_{\mathrm{i}}) = 2.51$ b of source information per channel use, which corresponds to an uncorrelated narrowband Rayleigh fading channel capacity E_b/N_0 bound of 5.57 dB [29].

B. Three-Dimensional EXIT Charts

As described in Section II-C, the outer component in Fig. 1 comprises the outer decoder and the check interleaver π_{c}, which can jointly consider the *a priori* LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}$ to generate the extrinsic LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}}$. As described in [6], the MIs of these LLR sequences are related to each other by the EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}}; \mathbf{e}) = f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}; \mathbf{e})]$, the inverse of which is shown in Fig. 3(a). Similar to an irregular code [22], this relationship depends on the individual EXIT functions of the outer decoder and check interleaver π_{c}, i.e., on $I(\tilde{\mathbf{c}}^{\mathrm{e}}; \mathbf{c}) = f_{\mathrm{o1}}[I(\tilde{\mathbf{c}}^{\mathrm{a}}; \mathbf{c})]$ and $I(\tilde{\mathbf{d}}^{\mathrm{e}}; \mathbf{d}) = f_{\mathrm{o2}}[I(\tilde{\mathbf{d}}^{\mathrm{a}}; \mathbf{d})]$, respectively. More specifically, the EXIT function of the outer component is given by the weighted average of the outer decoder and of the check interleaver's EXIT functions according to

$$f_{\mathrm{o}}\left[I\left(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}; \mathbf{e}\right)\right] = R_{\mathrm{c}} \cdot f_{\mathrm{o1}}\left[I\left(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}; \mathbf{e}\right)\right] + (1 - R_{\mathrm{c}}) \cdot f_{\mathrm{o2}}\left[I\left(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}; \mathbf{e}\right)\right]. \tag{1}$$

Note that, for the case where the *a priori* LLRs in the sequence $\tilde{\mathbf{d}}^{\mathrm{a}}$ are Gaussian distributed, the EXIT function of the check

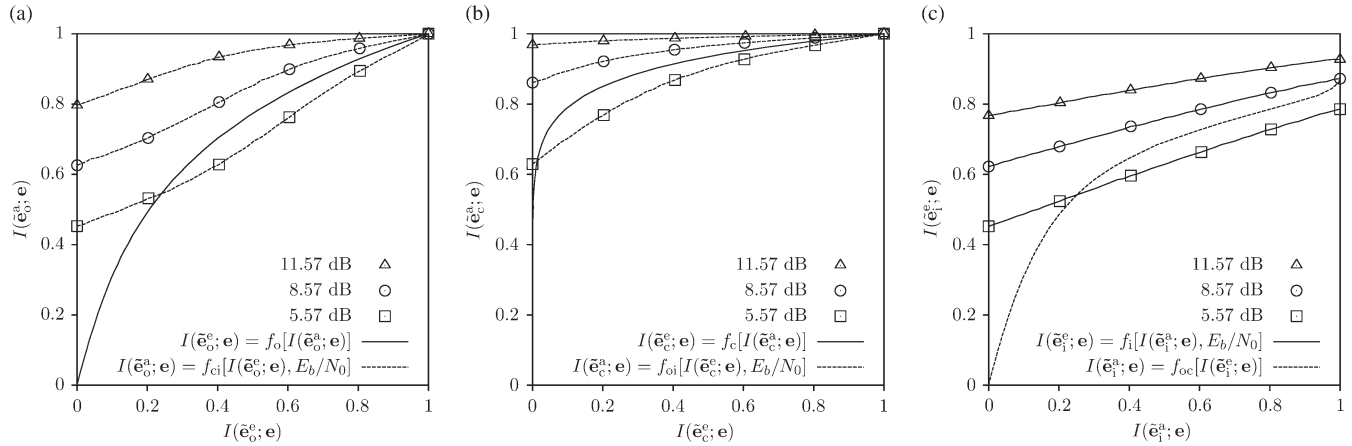


Fig. 3. Two-dimensional EXIT charts [4] for the block-based intermediate code scheme, which are projected in terms of (a) the outer component, (b) the check component, and (c) the inner component. In each case, E_b/N_0 values of 5.57, 8.57, and 11.57 dB are considered for the EXIT function to which the inner component contributes, because its EXIT characteristics depend on the nature of the uncorrelated narrowband Rayleigh fading channel.

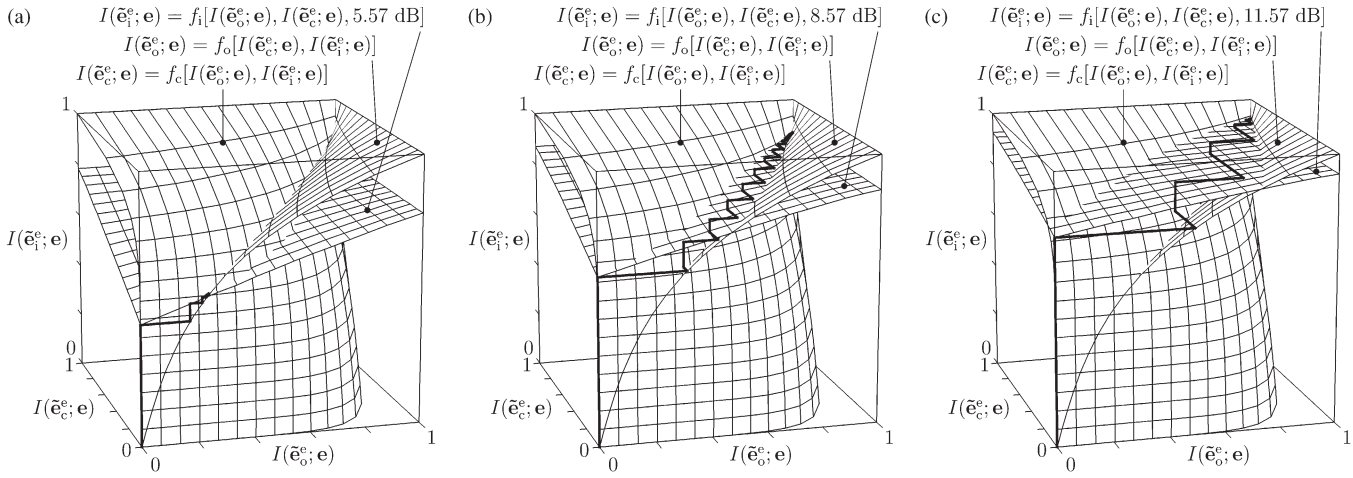


Fig. 4. Three-dimensional EXIT charts [4] for the block-based intermediate code scheme when communicating over an uncorrelated narrowband Rayleigh fading channel with E_b/N_0 values of (a) 5.57, (b) 8.57, and (c) 11.57 dB. Note that, in each case, the iterative decoding trajectory resides *below* the EXIT function of the inner component and *above* the other two EXIT functions.

interleaver π_{c} can analytically be obtained using an analogy to [30, eqs. (4) and (14)]. More specifically, we have

$$f_{\mathrm{o2}}\left[I(\tilde{\mathbf{d}}^{\mathrm{a}};\mathbf{d})\right] = \frac{N_{\mathbf{d}}-3}{N_{\mathbf{d}}} I(\tilde{\mathbf{d}}^{\mathrm{a}};\mathbf{d}) + \frac{3}{N_{\mathbf{d}}} J\left(\sqrt{2J^{-1}\left[I(\tilde{\mathbf{d}}^{\mathrm{a}};\mathbf{d})\right]^2}\right) \tag{2}$$

where $J(\cdot)$ and $J^{-1}(\cdot)$ are defined in [30].

Observe in Fig. 1 that the *a priori* LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}$ is obtained as a sum of the extrinsic LLR sequences $\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}}$ and $\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}}$. Hence, the outer component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}};\mathbf{e})]$ may be expressed as a 3-D function of two arguments, i.e., as $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}), I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})]$. Note that, for the case where the extrinsic LLRs in the sequences $\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}}$ and $\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}}$ are Gaussian distributed, the MI of the *a priori* LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}$ can analytically be obtained using an analogy to [30, eq. (4)], yielding

$$I\left(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}};\mathbf{e}\right) = J\left(\sqrt{J^{-1}\left[I\left(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}\right)\right]^2 + J^{-1}\left[I\left(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e}\right)\right]^2}\right). \tag{3}$$

This approach allows the 3-D EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}), I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})]$ to be obtained from the 2-D EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}};\mathbf{e})]$ using only a relatively low number of simulations.

In a similar manner, the 3-D function $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{c}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}), I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})]$ can be used to express the check component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{c}}[I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{a}};\mathbf{e})]$, the inverse of which is shown in Fig. 3(b). Note that, for the case where the *a priori* LLRs in the sequence $\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{a}}$ are Gaussian distributed, the EXIT function of the check component can analytically be obtained using [30, eq. (9)]. Finally, the inner component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{i}}[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e}), E_b/N_0]$ in Fig. 3(c) can be expressed as $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e}) = f_{\mathrm{i}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}), I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}), E_b/N_0]$. Because all three of the 3-D EXIT functions have arguments and values of $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}), I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e})$ and $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})$, they can all be plotted in a single 3-D EXIT chart with axes labeled with these MIs. This approach is shown in Fig. 4 for the scheme considered when communicating over uncorrelated narrowband Rayleigh fading channels with a range of SNRs per bit E_b/N_0 above the channel capacity bound of 5.57 dB.

As shown in Fig. 4, the MI $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})$ of the extrinsic LLRs generated by the inner component increases as the MIs $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e})$ and $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e})$ of its *a priori* LLRs increase, as well as when

the channel SNR increases, as usual. However, the inner component cannot generate extrinsic LLRs with an MMI of $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})=1$, regardless of how high the *a priori* MIs and the channel SNR are, demonstrating that it is not MMIA, as described in Section I. However, Fig. 4 shows that the outer and check components *can* generate extrinsic LLRs with MMIs of $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e})=1$ and $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e})=1$, because they satisfy the first MMIA condition in the Section I.

As a result, iterative decoding convergence toward the ML SER performance is facilitated. The iterative decoding trajectory [6] in Fig. 4(b) shows that this approach is facilitated when the E_b/N_0 value exceeds a threshold of approximately 8.57 dB, which happens to be 3 dB from the channel capacity bound of 5.57 dB [29]. This result is evident, because the trajectory reaches the specific vertical edge of the 3-D EXIT chart, where we have $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e})=1$ and $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e})=1$. Note that the trajectories in Fig. 4 correspond to a source sequence length of $N_{\mathbf{a}}=2\times10^6$ and a component activation order of $\{\text{inner, outer, check; inner, outer, check; }\ldots\}$, where each of these three decoding processes is associated with trajectory segments with a different orthogonal direction. This decoder activation order was selected, because it is the periodic order that offers the greatest trajectory advancement at the commencement of the iterative decoding process, as shown in Fig. 4. Our future work will investigate adaptive nonperiodic component activation orders to maximize the SER performance that can be achieved at a limited computational complexity. Intuitively, it seems that, at a particular stage in an iterative decoding process, the best decoding component to activate is the component that offers the greatest trajectory advancement, relative to its computational complexity. Naturally, a different component activation order would yield a trajectory that advances in a different sequence of directions.

C. Two-Dimensional EXIT Chart Projections

Note that the 3-D EXIT charts in Fig. 4 can be projected into two dimensions [4], [13], as shown in Fig. 3. For example, in addition to the EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e})=f_{\mathrm{o}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}};\mathbf{e})]$ of the outer component, Fig. 3(a) also provides a second EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}};\mathbf{e})=f_{\mathrm{ci}}[I(\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}};\mathbf{e}),E_b/N_0]$ for E_b/N_0 values of 5.57, 8.57, and 11.57 dB. This EXIT function characterizes the process in which the LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}}$ is provided for the inner and check components. These components are iteratively operated until convergence is achieved, and then, they generate the LLR sequence $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{a}}$, as shown in Fig. 1. As a result, Fig. 3(a) characterizes a component activation order of $\{\text{inner,check,inner, check, }\ldots\text{, inner, check, outer; inner, check, inner, check, }\ldots\text{, inner, check, outer; }\ldots\}$.

Similarly, Fig. 3(b) complements the check component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e})=f_{\mathrm{c}}[I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{a}};\mathbf{e})]$ with the EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{a}};\mathbf{e})=f_{\mathrm{oi}}[I(\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}};\mathbf{e}),E_b/N_0]$, which is provided for E_b/N_0 values of 5.57, 8.57, and 11.57 dB. This EXIT function characterizes the iterative operation of the outer and inner components until convergence is achieved. Finally, the EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e})=f_{\mathrm{oc}}[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})]$ in Fig. 3(c) characterizes the iterative operation of the outer and check components until convergence is achieved. Note that, because the inner component does not contribute to this EXIT function, it is not parameterized by the E_b/N_0 value. Instead, it is the inner component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})=f_{\mathrm{i}}[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e}),E_b/N_0]$ in Fig. 3(c) that is provided for E_b/N_0 values of 5.57, 8.57, and 11.57 dB.

As shown in Fig. 3, the 2-D EXIT chart tunnels become closed at $E_b/N_0=5.57$ dB, similar to the 3-D EXIT chart tunnel in Fig. 4(a). By contrast, in Fig. 3(a) and (b), open 2-D EXIT chart tunnels are obtained for E_b/N_0 values of 8.57 and 11.57 dB, similar to Fig. 4(b) and (c), respectively. Note that, in the case of Fig. 3(c), iterative decoding convergence toward the $[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e}),I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})]=[1,0.87]$ point is facilitated at $E_b/N_0=8.57$ dB, whereas convergence toward the [1,0.93] point is facilitated when $E_b/N_0=11.57$ dB. Although MMI is not achieved for the LLR sequence $\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}}$ in these cases, we consider the corresponding EXIT chart tunnels to be open, because MMI *is* achieved for $\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}}$. As shown in Fig. 1, this case implies that MMI is also achieved for $\tilde{\mathbf{e}}_{\mathrm{o}}^{\mathrm{e}}$ and $\tilde{\mathbf{e}}_{\mathrm{c}}^{\mathrm{e}}$, which is associated with achieving the ML SER performance, as described in the Section I.

As detailed in [31], the area properties of 2-D EXIT charts can be exploited to determine if an iteratively decoded scheme suffers from capacity loss, which prevents near-capacity operation. In Fig. 3(c), we are therefore interested in the area A_{i} beneath the inner component's EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}};\mathbf{e})=f_{\mathrm{i}}[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e}),E_b/N_0]$ and the area A_{oc} beneath the inverse of the other components' EXIT function $I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{a}};\mathbf{e})=f_{\mathrm{oc}}[I(\tilde{\mathbf{e}}_{\mathrm{i}}^{\mathrm{e}},\mathbf{e})]$. As expected in [31], it is shown that $A_{\mathrm{i}}\cdot\log_2(M_{\mathrm{i}})\approx C$, where the channel capacity C is 2.51, 3.04, and 3.43 b per channel use, when $E_b/N_0=5.57$, 8.57, and 11.57 dB, respectively. Furthermore, $A_{\mathrm{oc}}\cdot\log_2(M_{\mathrm{i}})\approx\eta$, where the throughput of our scheme is $\eta=2.51$ b per channel use, as described in Section III-A. This result shows that the proposed scheme does not suffer from capacity loss [31]. Indeed, without changing the areas beneath them, the EXIT functions in Fig. 3(c) could be reshaped to more accurately match each other by using irregular coding [22]. Our future research will employ this technique to facilitate the creation of open EXIT chart tunnels at E_b/N_0 values that are arbitrarily close to the channel capacity bound of 5.57 dB.

IV. Symbol Error Ratio Performance

Let us now consider the SER performance of the scheme introduced in Section III-A, which employs the schematic in Fig. 1. Recall that this scheme comprises an $L_{\mathrm{o}}=5$-bit FLC, an $R_{\mathrm{c}}=5/6$-rate block-based intermediate code, and an $M_{\mathrm{i}}=$ 16-QAM modulator, yielding an effective throughput of $\eta=$ 2.51 b of source information per channel use. We will compare the performance of this "block-based intermediate code" scheme to the performance of two appropriate benchmarkers.

We refer to the first benchmarker as the "convolutional intermediate code" scheme, because it replaces the block-based intermediate code in Fig. 1 with an MMIA convolutional intermediate code. Here, a convolutional intermediate code that employs only a single memory element and a coding rate of $R_{\mathrm{p}}=1$ was selected to minimize its decoding complexity. Because the convolutional intermediate code has a coding rate of $R_{\mathrm{p}}=1$, it is necessary to reduce the FLC coding rate R_{o} to maintain an effective throughput of $\eta=2.51$ and to allow fair

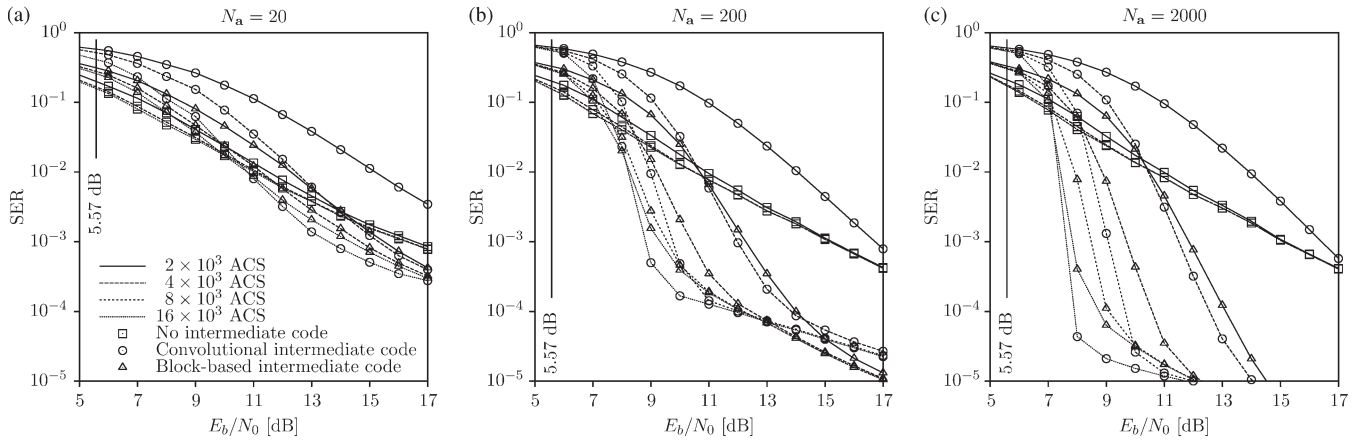


Fig. 5. SER versus uncorrelated narrowband Rayleigh fading channel E_b/N_0 performance of the "no intermediate code," "convolutional intermediate code," and "block-based intermediate code" schemes under various complexity constraints for the case where source sequence lengths $N_{\mathbf{a}}$ of (a) 20, (b) 200, and (c) 2000 are employed.

comparison with the "block-based intermediate code" scheme. This approach may be achieved by employing an $L_{\mathrm{o}} = 6$-bit FLC with the 16 octal codewords {64, 70, 00, 51, 35, 45, 56, 14, 63, 21, 32, 42, 26, 77, 07, 13}. Again, this mapping was designed considering the source symbol occurrence probabilities to maximize the average Hamming distance between codewords and, hence, to achieve the highest possible minimum separation of $d_{\mathrm{o}} = 2$.

To characterize the penalty associated with omitting an intermediate code, we also consider a second benchmarker, which we refer to as the "no intermediate code" scheme. In this arrangement, the bit sequence $\mathbf{e}$ in Fig. 1 is directly provided by $\mathbf{c}$. It is therefore necessary to employ the $L_{\mathrm{o}} = 6$-b FLC of the "convolutional intermediate code" scheme to achieve an effective throughput of $\eta = 2.51$ b of source information per channel use.

The SER performance of the "no intermediate code," "convolutional intermediate code," and "block-based intermediate code" schemes was investigated for transmission over uncorrelated narrowband Rayleigh fading channels with a range of E_b/N_0 values above the channel capacity bound of 5.57 dB. Source sequences $\mathbf{a}$ that comprise $N_{\mathbf{a}} = 20$, 200, and 2000 randomly generated 16-ary symbols were employed, resulting in encoded bit sequences $\mathbf{e}$ that comprise $N_{\mathbf{e}} = 120$, 1200, and 12000 b, respectively, for each of the schemes considered. This range of relatively short encoded bit sequence lengths was considered, because it is typical of the challenging audio, speech, and wireless sensor networking (WSN) scenarios. During each of our simulations, we considered the transmission of as many source sequences as necessary to observe a statistically significant number of decoding errors. For simplicity and to facilitate fair comparisons, the interleavers π_{o} and π_{i} in Fig. 1 employed different random designs for each source sequence. The repercussions of this choice will be discussed as follows.

Because the decoding complexity that can be afforded in audio, speech, and WSN applications is typically limited, we recorded the SERs that can be achieved using a maximum complexity of 2×10^3, 4×10^3, 8×10^3, and 16×10^3 ACS operations per source symbol. In the "block-based intermediate code" scheme, these complexity limits were, respectively, found to facilitate three, five, 10, and 20 iterations of the component activation order described in Section III-B. Similarly, two, four, seven, and 15 iterations of the component activation sequence $\{\text{inner}, \text{intermediate}, \text{outer}; \text{inner}, \text{intermediate}, \text{outer}; \ldots\}$ were facilitated in the "convolutional intermediate code" scheme, respectively. Finally, the aforementioned complexity limits facilitated three, five, 11, and 22 iterations of the component activation sequence $\{\text{inner}, \text{outer}; \text{inner}, \text{outer}; \ldots\}$ in the "no intermediate code" scheme. The recorded SERs are plotted in Fig. 5.

Observe in Fig. 5 that the "no intermediate code" scheme offers little interleaver or iteration gain, yielding similar SERs, regardless of the affordable source sequence length $N_{\mathbf{a}}$ and the decoding complexity. By contrast, the "convolutional intermediate code" and the "block-based intermediate code" schemes exhibit significant interleaver and iteration gains, offering steeper and "nearer capacity" turbo cliffs as the source sequence length $N_{\mathbf{a}}$ and decoding complexity are increased, respectively. At sufficiently high E_b/N_0 values, intermediate coding facilitates iterative decoding convergence toward the ML SER performance, as demonstrated by the error floors in Fig. 5, which are reduced, as the source sequence length $N_{\mathbf{a}}$ is increased.

Fig. 5 shows that, when the affordable decoding complexity is low, the proposed "block-based intermediate code" scheme facilitates operation at lower E_b/N_0 values than the "convolutional intermediate code" scheme. For example, Fig. 5(a) shows that the "block-based intermediate code" arrangement achieves an SER of 10^{-3} at a lower E_b/N_0 value than the "convolutional intermediate code" scheme when the affordable complexity is 2×10^3 or 4×10^3 ACS operations for each of the $N_{\mathbf{a}} = 20$ source symbols. Fig. 5(b) and (c) shows that similar gains of up to 4.57 dB are offered when we have $N_{\mathbf{a}} = 200$ or 2000 and the affordable complexity is 2×10^3, 4×10^3 or 8×10^3 ACS operations per source symbol.

However, Fig. 5 shows that, when a higher decoding complexity of 16×10^3 ACS operations per symbol is affordable, the "convolutional intermediate code" benchmarker offers the lowest SERs in the turbo cliff range of E_b/N_0 values. This range is 11–17 dB for the case of $N_{\mathbf{a}} = 20$, 8–13 dB for $N_{\mathbf{a}} = 200$, and 7–12 dB for $N_{\mathbf{a}} = 2000$, as shown in Fig. 5(a)–(c),

respectively. This result may be attributed to the particular choice of the inner and outer codes, which allows the "convolutional intermediate code" scheme to maintain an open EXIT chart tunnel at lower E_b/N_0 values than the "block-based intermediate code" scheme. As described in Section III-C, our future research effort will consider the use of irregular inner and outer codes [22], [32] to create open EXIT chart tunnels at near-capacity E_b/N_0 values.

Observe in Fig. 5 that the "block-based intermediate code" scheme results in a more rapidly diminishing error floor than the "convolutional intermediate code" scheme, achieving lower SERs at high E_b/N_0 values, even when a decoding complexity of 16×10^3 ACS operations per symbol can be afforded. This case may be attributed to the improved distance spectrum and ML SER performance that is afforded by the block-based intermediate code's systematic design [1], which is described in Section II-B. This case ensures that the legitimate permutations of the output bit sequence $\mathbf{e}$ in Fig. 1 are separated by Hamming distances that are at least as high as the distances that correspond to the block-based intermediate code's input bit sequence $\mathbf{c}$. The legitimate permutations of this sequence are separated by Hamming distances of at least two, because it is obtained by interleaving the bit sequence $\mathbf{b}$ output by the $d_{\mathrm{o}} = 2$ FLC encoder, as described in Section III-A. By contrast, the convolutional intermediate code is nonsystematic, and hence, some permutations of its encoded bit sequence $\mathbf{e}$ were separated by a Hamming distance of just one whenever the interleaver π_{o} was randomly allocated a deficient design during our investigations.

Note, however, that the distance spectrum of the "convolutional intermediate code" scheme can be improved by simply employing S-random designs [33] for the interleaver π_{o} rather than random designs. However, it is also straightforward to design improved interleavers for the "block-based intermediate code" scheme, allowing it to achieve an even better distance spectrum. This case is because the lengths of the block-based intermediate code's error events [20] are limited by the length of its blocks, as depicted in Fig. 2. This condition will be the topic of our future research.

V. Conclusion

In this paper, we have proposed the novel intermediate code design, which facilitates iterative decoding convergence to the ML error ratio performance in serially concatenated schemes. Unlike conventional intermediate codes, our design is block based rather than convolutional. As a result, our approach offers a number of significant advantages when employed in practical scenarios, where short frames are employed, and the affordable complexity is limited.

First, due to its block-based nature, the proposed intermediate code has low implementational complexity, requiring reduced memory and allowing parallel processing, unlike convolutional intermediate codes. Second, we showed that the design and optimization of serially concatenated schemes, as well as the analysis and visualization of iterative decoding convergence, is more intuitive when employing block-based rather than convolutional intermediate codes. Third, the lengths of our block-based intermediate code's error events are limited by its block length, simplifying the task of designing interleavers to improve the distance spectrum. Finally, the proposed block-based intermediate code has lower decoding complexity than the convolutional intermediate code, facilitating improved error ratios in practical scenarios, where the affordable computational complexity is limited. In conclusion, the proposed design procedure is applicable to arbitrary sources and for employment between arbitrary inner and outer codes.

References

[1] I. Sason and S. Shamai, "Performance analysis of linear codes under maximum-likelihood decoding: A tutorial, foundation and trends," *Found. Trends Commun. Inform. Theory*, vol. 3, no. 1/2, pp. 1–225, Jul. 2006.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes (1)," in *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993, vol. 2, pp. 1064–1070.

[3] D. J. C. MacKay and R. M. Neal, "Near-Shannon-limit performance of low-density parity check codes," *Electron. Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[4] M. Tüchler, "Convergence prediction for iterative decoding of threefold concatenated systems," in *Proc. IEEE Global Telecommun. Conf.*, Taipei, Taiwan, Nov. 2002, vol. 2, pp. 1358–1362.

[5] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for 'turbo-like' codes," in *Proc. Allerton Conf. Commun., Control, Comput.*, Urbana, IL, Sep. 1998, pp. 201–210.

[6] S. ten Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.

[7] J. Kliewer, N. Görtz, and A. Mertins, "Iterative source–channel decoding with Markov random field source models," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3688–3701, Oct. 2006.

[8] J. Kliewer, A. Huebner, and D. J. Costello, "On the achievable extrinsic information of inner decoders in serial concatenation," in *Proc. IEEE Int. Symp. Inf. Theory*, Seattle, WA, Jul. 2006, pp. 2680–2684.

[9] R. G. Maunder and L. Hanzo, "Iterative decoding convergence and termination of serially concatenated codes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 216–224, Jan. 2010.

[10] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," in *Proc. IEEE Int. Conf. Commun.*, Vancouver, BC, Canada, Jun. 1999, vol. 2, pp. 858–863.

[11] S. Benedetto and G. Montorsi, "Serial concatenation of block and convolutional codes," *Electron. Lett.*, vol. 32, no. 10, pp. 887–888, May 1996.

[12] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electron. Lett.*, vol. 32, no. 13, pp. 1186–1188, Jun. 1996.

[13] F. Brannstrom, L. K. Rasmussen, and A. J. Grant, "Convergence analysis and optimal scheduling for multiple concatenated codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 9, pp. 3354–3364, Sep. 2005.

[14] R. Y. Tee, R. G. Maunder, J. Wang, and L. Hanzo, "Near-capacity irregular bit-interleaved coded modulation," in *Proc. IEEE Veh. Technol. Conf.*, Singapore, May 2008, pp. 549–553.

[15] N. S. Othman, M. El-Hajjar, A. Q. Pham, O. Alamri, S. X. Ng, and L. Hanzo, "Over-complete source-mapping-aided AMR-WB MIMO transceiver using three-stage iterative detection," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 751–755.

[16] S. X. Ng, J. Wang, and L. Hanzo, "Unveiling near-capacity code design: The realization of Shannon's communication theory for MIMO channels," in *Proc. IEEE Int. Conf. Commun.*, Beijing, China, May 2008, pp. 1415–1419.

[17] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 2, pp. 284–287, Mar. 1974.

[18] H.-A. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

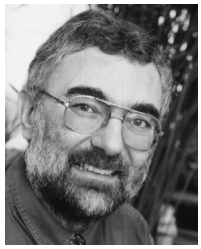
[19] K. Kusume and G. Bauch, "A parallel APP decoding algorithm for accelerating decoding throughput of turbo codes," in *Proc. IEEE Int. Symp. Wireless Commun. Syst.*, Reykjavik, Iceland, Oct. 2008, pp. 528–532.

[20] B. Vucetic, Y. Li, L. C. Perez, and F. Jiang, "Recent advances in turbo code design and theory," *Proc. IEEE*, vol. 95, no. 6, pp. 1323–1344, Jun. 2007.

[21] J. Sun and O. Y. Takeshita, "Interleavers for turbo codes using permutation polynomials over integer rings," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 101–119, Jan. 2005.
[22] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2002, pp. 748–753.
[23] S. ten Brink and G. Kramer, "Design of repeat–accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2764–2772, Nov. 2003.
[24] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. Commun.*, vol. 53, no. 8, pp. 1288–1299, Aug. 2005.
[25] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," in *Proc. IEEE Global Telecommun. Conf.*, San Antonio, TX, Nov. 2001, vol. 2, p. 1036.
[26] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[27] J. Hagenauer and N. Görtz, "The turbo principle in joint source–channel coding," in *Proc. IEEE Inf. Theory Workshop*, Paris, France, Mar. 2003, pp. 275–278.
[28] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 1, pp. 55–67, Jan. 1982.
[29] L. Hanzo, S. X. Ng, T. Keller, and W. Webb, *Quadrature Amplitude Modulation*. Chichester, U.K.: Wiley, 2004.
[30] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
[31] A. Ashikhmin, G. Kramer, and S. ten Brink, "Code rate and the area under extrinsic information transfer curves," in *Proc. IEEE Int. Symp. Inf. Theory*, Lausanne, Switzerland, Jun. 2002, p. 115.
[32] R. G. Maunder and L. Hanzo, "Near-capacity irregular variable-length coding and irregular unity rate coding," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5500–5507, Nov. 2009.
[33] S. Dolinar and D. Divsalar, "Weight distributions for turbo codes using random and nonrandom permutations," *Telecommun. Data Acquisition Progr. Rep.*, vol. 122, pp. 56–65, Apr. 1995.

Robert G. Maunder (M'03) received the B.Eng. (first-class honors) degree in electronic engineering and the Ph.D. degree in wireless communications from the University of Southampton, Southampton, U.K., in July 2003 and December 2007, respectively.

Since December 2007, he has been a Lecturer with the School of Electronics and Computer Science, University of Southampton. His research interests include joint source/channel coding, iterative decoding, irregular coding, and modulation techniques, for which he has published a number of IEEE papers.



Lajos Hanzo (F'04) received the degree in electronics in 1976 and the doctorate degree in 1983. In 2009, he received the honorary doctorate "Doctor Honaris Causa" degree from the Technical University of Budapest, Budapest, Hungary.

During his 35-year career in telecommunications, he has held various research and academic posts in Hungary, Germany and the U.K. Since 1986, he has been with the School of Electronics and Computer Science, University of Southampton, U.K., where he holds the chair in telecommunications. He has co-authored 20 John Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, published about 990 research entries on IEEE Xplore, acted as Technical Program Committee Chair of IEEE conferences, presented keynote lectures, and been awarded a number of distinctions. Currently, he is directing an academic research team, working on a range of research projects in the field of wireless multimedia communications sponsored by industry, the Engineering and Physical Sciences Research Council (EPSRC) U.K., the European IST Programme, and the Mobile Virtual Centre of Excellence (VCE), U.K. He is an enthusiastic supporter of industrial and academic liaison, and he offers a range of industrial courses. He is also a Governor of the IEEE Vehicular Technology Society. Since 2008, he has been the Editor-in-Chief of the IEEE Press and, since 2009, a Chaired Professor, also at Tsinghua University, Beijing, China. For further information on research in progress and associated publications, see http://www-mobile.ecs.soton.ac.uk.