Irregular Trellis for the Near-Capacity Unary Error Correction Coding of Symbol Values From an Infinite Set

Wenbo Zhang, *Student Member, IEEE*, Matthew F. Brejza, Tao Wang, Robert G. Maunder, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Irregular joint source and channel coding (JSCC) scheme is proposed, which we refer to as the irregular unary error correction (IrUEC) code. This code operates on the basis of a single irregular trellis, instead of employing a set of separate regular trellises, as in previous irregular trellis-based codes. Our irregular trellis is designed with consideration of the UEC free distance, which we characterize for the first time in this paper. We conceive the serial concatenation of the proposed IrUEC code with an irregular unity rate code (IrURC) code and propose a new EXtrinsic Information Transfer (EXIT) chart matching algorithm for parametrizing these codes. This facilitates the creation of a narrow EXIT tunnel at a low E_b/N_0 value and provides near-capacity operation. Owing to this, our scheme is found to offer a low symbol error ratio (SER), which is within 0.4 dB of the discrete-input continuous-output memoryless channel (DCMC) capacity bound in a particular practical scenario, where gray-mapped quaternary phase shift keying (QPSK) modulation is employed for transmission over an uncorrelated narrowband Rayleigh-fading channel with an effective throughput of $0.508\text{bit}\,\text{s}^{-1}\,\text{Hz}^{-1}$. Furthermore, the proposed IrUEC–IrURC scheme offers a SER performance gain of 0.8 dB, compared to the best of several regular and irregular separate source and channel coding (SSCC) benchmarkers, which is achieved without any increase in transmission energy, bandwidth, transmit duration, or decoding complexity.**

***Index Terms*—Joint source–channel coding, irregular codecs, channel capacity, iterative decoding.**

I. Introduction

IN MOBILE wireless scenarios, multimedia transmission is required to be bandwidth efficient and resilient to transmission errors, motivating both source and channel coding [1]–[3]. Classic Separate Source and Channel Coding (SSCC) may be achieved by combining a near-entropy source code with a near-capacity channel code. In this scenario, it is theoretically possible to reconstruct the source information with an infinitesimally low probability of error, provided that the transmission rate does not exceed the channel's capacity [4]. However, separate source-channel coding [4] is only capable of approaching the capacity in the general case by imposing both infinite complexity and infinite latency. For example, adaptive arithmetic coding [5] and Lempel-Ziv coding [6] are capable of encoding a sequence of symbols using a near-entropy number of bits per symbol. However, these schemes require both the transmitter and receiver to accurately estimate the occurrence probability of every symbol value that the source produces. In practice, the occurrence probability of rare symbol values can only be accurately estimated, if a sufficiently large number of symbols has been observed, hence potentially imposing an excessive latency.

This motivates the design of universal codes, such as the Elias Gamma (EG) code [7], which facilitate the binary encoding of symbols selected from infinite sets, without requiring any knowledge of the corresponding occurrence probabilities at either the transmitter or receiver. The H.264 video codec [8] employs the EG code and this may be concatenated with classic channel codes, such as a Convolutional Code (CC) to provide a separate error correction capability. Nevertheless, this SSCC typically suffers from a capacity loss, owing to the residual redundancy that is typically retained during EG encoding, which results in an average number of EG-encoded bits per symbol that exceeds the entropy of the symbols.

In order to exploit the residual redundancy and hence to achieve near-capacity operation, the classic SSCC schemes may be replaced by Joint Source and Channel Coding (JSCC) arrangements [9] in many applications. As we have previously demonstrated in [10, Fig. 1], the symbols that are EG encoded in H.264 are approximately zeta probability distributed [11], resulting in most symbols having low values, but some rare symbols having values around 1000. Until recently, the decoding complexity of all previous JSCCs, such as Reversible Variable Length Codes (RVLCs) [12] and Variable Length Error Correction (VLEC) codes [13], increased rapidly with the cardinality of the symbol set, so much so that it became excessive for the H.264 symbol probability distribution and asymptotically tending to infinity, when the cardinality is infinite.

Against this background, a novel JSCC scheme referred to as a Unary Error Correction (UEC) code [10] was proposed as the first JSCC that mitigates the capacity loss and incurs

Manuscript received October 31, 2014; revised April 14, 2015 and July 25, 2015; accepted October 8, 2015. This work was supported in part by the EPSRC, Swindon UK under Grant EP/J015520/1 and Grant EP/L010550/1, in part by the TSB Swindon UK under Grant TS/L009390/1, in part by the RCUK under the India-UK Advanced Technology Centre (IU-ARC), and in part by the EU under the CONCERTO project and in part by the European Research Council's Advanced Fellow grant. The associate editor coordinating the review of this paper and approving it for publication was M. Xiao.

The authors are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: wz4g11@ecs.soton.ac.uk; mfb2g09@ecs.soton.ac.uk; tw08r@ecs.soton.ac.uk; rm@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2015.2493149

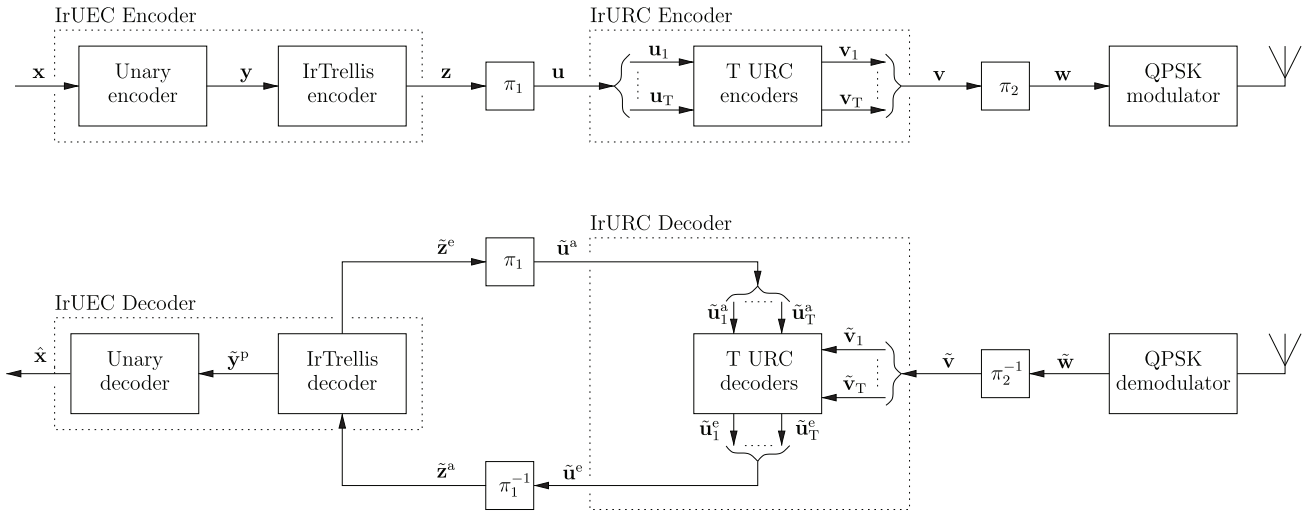


Fig. 1. Schematic of the proposed IrUEC-IrURC scheme, in which an IrUEC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

only a moderate decoding complexity, even when the cardinality of the symbol set is infinite. In a particular practical scenario, an iteratively-decoded serial concatenation of the UEC code with an Irregular Unity Rate Code (IrURC) was shown to offer a 1.3 dB gain compared to a SSCC benchmarker, without incurring an increased transmission energy, duration, bandwidth or decoding complexity. Furthermore, this was achieved within 1.6 dB of the Quaternary Phase Shift Keying (QPSK)-modulated uncorrelated narrow band Rayleigh fading Discrete-input Continuous-output Memoryless Channel (DCMC) capacity bound.

In this paper, we will further exploit the properties of UEC codes in order to facilitate reliable operation even closer to the capacity bound. More specifically, we propose an *Irregular* Unary Error Correction (IrUEC) code, which extends the regular UEC of our previous work [10]. *This IrUEC code employs different UEC parametrizations for the coding of different subsets of each message frame, in analogy with previous irregular codes, such as the IrURC [14], the Irregular Convolutional Code (IrCC) [15] and the Irregular Variable Length Code (IrVLC) [16]. However, these previous irregular codes operate on the basis of a number of separate trellises, each of which has a different but uniform structure and is used for the coding of a different subset of the message frame. By contrast, our new IrUEC code operates on the basis of a single irregular trellis having a novel design. This trellis has a non-uniform structure that applies different UEC parametrizations for different subsets of the frame on a bit-by-bit basis. This allows the irregularity of the proposed IrUEC code to be controlled on a fine-grained bit-by-bit basis, rather than on a symbol-by-symbol basis, hence facilitating nearer-to-capacity operation. More specifically, our results demonstrate that controlling the IrUEC irregularity on a bit-by-bit basis offers gains of up to 0.2 dB over the symbol-by-symbol approach, without imposing any increase in transmission energy, bandwidth, latency or decoding complexity.*

This bit-by-bit IrUEC approach is facilitated by some particular properties of UEC codes, which grant some commonality to all UEC parametrizations. By exploiting this fine-grained control of the IrUEC irregularity, the IrUEC EXtrinsic Information Transfer (EXIT) function may be shaped to create a narrow, but marginally open EXIT chart tunnel. This implies that near-capacity operation is facilitated, according to the theoretical properties of EXIT charts [17].

The rest of this paper is organized as follows. Section II describes a transmitter that serially concatenates the proposed IrUEC encoder with a IrURC encoder, while Section III describes the corresponding iterative receiver. The IrUEC encoder and decoder operate on the basis of our novel irregular trellis structure, which allows bit-level control of the irregular coding fractions. The free distance of UEC codes is quantified for the first time in Section IV, which proposes a novel low-complexity heuristic method conceived for this purpose. This is used for selecting a family of UEC trellis structures having a wide variety of EXIT function shapes. The resultant UEC trellis family maximises the design freedom for the IrUEC EXIT function and therefore has a general applicability for IrUEC codes used in diverse applications. Furthermore, for any particular application of an IrUEC code, we propose a double-sided EXIT chart matching algorithm for selecting the specific fraction of the frame that should be encoded using each IrUEC and IrURC trellis structure. This allows the EXIT functions of IrUEC and IrURC codes to be accurately shaped for closely matching each other, hence creating a narrow but marginally open EXIT chart tunnel. In Section V, the proposed IrUEC-IrURC scheme is compared to an irregular JSCC benchmarker, which is referred to as the EG-IrCC-IrURC scheme. The first version of this benchmarker employs the recursive systematic CCs that were originally recommended as IrCC component codes in [15]. However, we demonstrate that the systematic nature of these IrCC component codes results in a capacity loss. This motivates the employment of the second version of our EG-IrCC-IrURC benchmarker, which employs the recursive non-systematic CCs of [10] as the IrCC component codes. The simulation results of Section V show that in a particular practical scenario, the proposed IrUEC-IrURC scheme provides a

0.8 dB gain over the best SSCC benchmarker, while operating within 0.4 dB of the capacity bound. This is achieved without any increase in transmission energy, bandwidth, latency or decoding complexity. Finally, Section VI concludes the paper.

II. IRUEC-IRURC ENCODER

In this section, we introduce the transmitter of the proposed IrUEC-IrURC scheme of Fig. 1. The IrURC encoder employs T number of component Unity Rate Code (URC) encoders $\{\text{URC}^t\}_{t=1}^{T}$, each having a distinct independent trellis structure. By contrast, the IrUEC employs a unary encoder and a novel *Irregular* Trellis (IrTrellis) encoder with a single irregular trellis. However, in analogy with the IrURC code, we note that this irregular trellis comprises a merging of S component UEC trellis structures $\{\text{UEC}^s\}_{s=1}^{S}$, where UEC^s is the s-th component UEC trellis structure that is defined by the corresponding codeword set $\mathbb{C}_s$, as illustrated in [10, Fig. 3(a)]. In Section II-A and Section II-B, the two components of the IrUEC encoder in Fig. 1, namely the unary encoder and the novel IrTrellis encoder are detailed. The IrURC encoder and the modulator are introduced in Section II-C.

A. Unary Encoder

The IrUEC encoder is designed for conveying a vector $\mathbf{x} = [x_i]_{i=1}^{a}$ comprising a number of symbols, as shown in Fig. 1. The value of each symbol $x_i \in \mathbb{N}_1$ may be modeled by an Independent and Identically Distributed (IID) Random Variable (RV) X_i, which adopts the value x with a probability of $\Pr(X_i = x) = P(x)$, where $\mathbb{N}_1 = \{1, 2, 3, \ldots, \infty\}$ is the infinite-cardinality set comprising all positive integers. Throughout this paper we assume that the symbol values obey a zeta probability distribution [11], since this models the symbols produced by multimedia encoders, as described in Section I. The zeta probability distribution is defined as

$$P(x) = \frac{x^{-s}}{\zeta(s)}, \tag{1}$$

where $\zeta(s) = \sum_{x\in\mathbb{N}_1} x^{-s}$ is the Riemann zeta function, $s > 1$ parametrizes the zeta distribution and $p_1 = \Pr(X_i = 1) = 1/\zeta(s)$ is the probability of occurrence for the most frequently occurring symbol value, namely $x = 1$. Without loss of generality, Table I exemplifies the first ten symbol probabilities $P(x_i)$ for a zeta distribution having the parameter $p_1 = 0.797$, which corresponds to $s = 2.77$ and was found in [10] to allow a fair comparison between unary- and EG-based schemes. Note that other p_1 values of 0.694, 0.8 and 0.9 have been investigated in [18], [19]. In the situation where the symbols obey the zeta probability distribution of (1), the symbol entropy is given by

$$H_X = \sum_{x\in\mathbb{N}_1} H[P(x)] = \frac{\ln(\zeta(s))}{\ln(2)} - \frac{s\zeta'(s)}{\ln(2)\zeta(s)}, \tag{2}$$

where $H[p] = p\log_2(1/p)$ and $\zeta'(s) = -\sum_{x\in\mathbb{N}_1} \ln(x)x^{-s}$ is the derivative of the Riemann zeta function.

As shown in Fig. 1, the IrUEC encoder represents the source vector $\mathbf{x}$ using a unary encoder. More specifically, each symbol x_i in the vector $\mathbf{x}$ is represented by a corresponding codeword $\mathbf{y}_i$ that comprises x_i bits, namely $(x_i - 1)$ binary ones followed by a zero, as exemplified in Table I. When the symbols adopt the zeta distribution of (1), the average unary codeword length l is only finite for $s > 2$ and hence for $p_1 > 0.608$ [10], in which case we have

$$l = \sum_{x\in\mathbb{N}_1} P(x)\cdot x = \frac{\zeta(s-1)}{\zeta(s)}. \tag{3}$$

Note that for $p_1 \leq 0.608$, our Elias Gamma Error Correction (EGEC) code of [19] may be employed in order to achieve a finite average codeword length, albeit at the cost of an increased complexity. In our future work, we will consider a novel Irregular EGEC code, which has a finite codeword length for $p_1 \leq 0.608$. Without loss of generality, in the example scenario of $p_1 = 0.797$, an average codeword length of $l = 1.54$ results. The output of the unary encoder is generated by concatenating the selected codewords $\{\mathbf{y}_i\}_{i=1}^{a}$, in order to form the b-bit vector $\mathbf{y} = [y_j]_{j=1}^{b}$. For example, the source vector $\mathbf{x} = [4, 1, 2]$ of $a = 3$ symbols yields the $b = 7$-bit vector $\mathbf{y} = [1110010]$. Note that the average length of the bit vector $\mathbf{y}$ is given by $(a \cdot l)$.

TABLE I
THE FIRST TEN SYMBOL PROBABILITIES FOR A ZETA DISTRIBUTION HAVING THE PARAMETER $p_1 = 0.797$, AS WELL AS THE CORRESPONDING UNARY AND EG CODEWORDS

x_i	$P(x_i)$	$\mathbf{y}_i$	
		Unary	EG
1	0.797	0	1
2	0.117	10	010
3	0.038	110	011
4	0.017	1110	00100
5	0.009	11110	00101
6	0.006	111110	00110
7	0.004	1111110	00111
8	0.003	11111110	0001000
9	0.002	111111110	0001001
10	0.001	1111111110	0001010

B. IrTrellis Encoder

Following unary encoding, the IrTrellis encoder of Fig. 1 employs a single new *irregular* trellis to encode the bit vector $\mathbf{y}$, rather than using a selection of separate trellis structures, as is necessary for the IrCC [15], IrVLC [16] and IrURC [14] coding schemes. Our novel irregular trellis structure is facilitated by the properties of the generalised trellis structure of [10, Fig. 3(a)], which was the basis of our previous work on regular UEC codes. This trellis structure is parametrized by an even number of states r and by the UEC codeword set $\mathbb{C}$, which comprises $r/2$ binary codewords of a particular length n. Each bit y_j of the unary-encoded bit sequence $\mathbf{y} = [y_j]_{j=1}^{b}$ corresponds to a transition in the UEC trellis from the previous state $m_{j-1} \in \{1, 2, \ldots, r\}$ to the next state $m_j \in \{1, 2, \ldots, r\}$. Each next state m_j is selected from two legitimate alternatives, depending both on the previous state m_{j-1} and on the bit value y_j, according to [18, (3)]. More specifically, regardless of how the UEC trellis is parametrized, a unary-coded bit of $y_j = 1$ causes a transition towards state $m_j = r - 1$ or r of the generalised UEC trellis of [10, Fig. 3(a)], while the $y_j = 0$-valued bit

at the end of each unary codeword causes a transition to state $m_j = 1$ or 2, depending on whether the current symbol x_i has an odd or even index i.

This common feature of all UEC trellises maintains synchronisation with the unary codewords and allows the residual redundancy that remains following unary encoding to be explicated for error correction. Furthermore, this common treatment of the unary-encoded bits in $\mathbf{y}$ between all UEC trellises allows them to merge in order to form our novel irregular trellis. More specifically, our novel irregular trellis can be seen as concatenation of a number of individual UEC trellis structures with different numbers of states r and different codeword sets $\mathbb{C}$. By contrast, CCs, Variable Length Codes (VLCs) and URC codes having different parametrizations do not generally exhibit the required similarity in their trellises. More specifically, the final state of a particular component encoder has no specific relationship with the initial state of the subsequent component encoder, hence preventing their amalgamation into IrCC, IrVLC and IrURC trellises, respectively.

The IrTrellis encoder of Fig. 1 encodes the b-bit unary-encoded bit sequence $\mathbf{y} = [y_j]_{j=1}^{b}$ using an irregular trellis that is obtained by concatenating b number of regular UEC trellis structures. The proposed IrTrellis can be constructed using diverse combinations of component regular UEC trellises, having any parametrization. However, the component regular trellises may be strategically selected in order to carefully shape the EXIT function of the IrUEC code, for the sake of producing a narrow EXIT chart tunnel and for facilitating near-capacity operation, as it will be detailed in Section IV. Without loss of generality, Fig. 1 provides an example of the irregular trellis for the example scenario where we have $b = 7$. Each bit y_j in the vector $\mathbf{y}$ is encoded using the corresponding one of these b trellis structures, which is parametrized by an even number of states r_j and the codeword set $\mathbb{C}_j = \{\mathbf{c}_1^j, \mathbf{c}_2^j, \ldots, \mathbf{c}_{r_j/2-1}^j, \mathbf{c}_{r_j/2}^j\}$, which comprises $r_j/2$ binary codewords of a particular length n_j. Note that successive trellis structures can have different numbers of states, subject to the constraint $r_j \leq r_{j-1} + 2$, as it will be demonstrated in the following discussions. Note that this constraint does not restrict the generality of the IrUEC trellis, since the IrUEC EXIT function shape is independent of the ordering of the component trellis structures.

As in the regular UEC trellis of [10], the encoding process always emerges from the state $m_0 = 1$. The unary-encoded bits of $\mathbf{y}$ are considered in order of increasing index j and each bit y_j causes the novel IrTrellis to traverse from the previous state $m_{j-1} \in \{1, 2, \ldots, r_{j-1}\}$ to the next state $m_j \in \{1, 2, \ldots, r_j\}$, which is selected from two legitimate alternatives. More specifically,

$$m_j = \begin{cases} 1 + \text{odd}(m_{j-1}) & \text{if } y_j = 0 \\ \min\left[m_{j-1} + 2, r_j - \text{odd}\left(m_{j-1}\right)\right] & \text{if } y_j = 1 \end{cases}, \quad (4)$$

where the function odd$(\cdot)$ yields 1 if the operand is odd or 0 if it is even. Note that the next state m_j in the irregular trellis is confined by the number of states r_j in the corresponding trellis structure, rather than by a constant number of states r, as in the regular UEC trellis of [10]. In this way, the bit sequence $\mathbf{y}$ identifies a path through the single irregular trellis, which may be represented by a vector $\mathbf{m} = [m_j]_{j=0}^{b}$ comprising $b + 1$ state values. As in the regular UEC trellis of [10], the transitions of the proposed irregular trellis are synchronous with the unary codewords of Table I. More specifically, just as each symbol x_i in the vector $\mathbf{x}$ corresponds to an x_i-bit codeword $\mathbf{y}_i$ in the vector $\mathbf{y}$, the symbol x_i also corresponds to a section $\mathbf{m}_i$ of the trellis path $\mathbf{m}$ comprising x_i transitions between $(x_i + 1)$ states. Owing to this, the path $\mathbf{m}$ is guaranteed to terminate in the state $m_b = 1$, when the symbol vector $\mathbf{x}$ has an even length a, while $m_b = 2$ is guaranteed when a is odd [10]. Note that the example unary-encoded bit sequence $\mathbf{y} = [1110010]$ corresponds to the path $\mathbf{m} = [1, 3, 5, 3, 2, 1, 1, 2]$ through the irregular UEC trellis of Fig. 2.

The path $\mathbf{m}$ may be modeled as a particular realization of a vector $\mathbf{M} = [M_j]_{j=0}^{b}$ comprising $(b + 1)$ RVs. Note that the probability $\Pr(M_j = m_j, M_{j-1} = m_{j-1}) = P(m_j, m_{j-1})$ of the transition from the previous state m_{j-1} to the next state m_j can be derived by observing the value of each symbol in the vector $\mathbf{x}$ and simultaneously its corresponding index. The state transition $\mathbf{M} = \{M_j\}_{j=0}^{b}$ follows the same rule shown in (4), and all the transitions can be categorised into four types, as illustrated in [10, (8)]. Owing to this, the probability of a transition $P(m_j, m_{j-1})$ in the irregular trellis is associated with the transition probabilities $\Pr(M_j = m, M_{j-1} = m') = P(m, m')$ in (5), shown at the bottom of the page. Note that these

$$P(m_j, m_{j-1}) = \begin{cases} \frac{1}{2l}\left[1 - \sum_{x=1}^{\left\lceil \frac{m_{j-1}}{2} \right\rceil} P(x)\right] & \text{if } m_{j-1} \in \{1, 2, 3, \ldots, r_{j-1} - 2\},\ m_j = m_{j-1} + 2 \\ \frac{1}{2l} P(x)\Big|_{x=\left\lceil \frac{m_{j-1}}{2} \right\rceil} & \text{if } m_{j-1} \in \{1, 2, 3, \ldots, r_{j-1} - 2\},\ m_j = 1 + \text{odd}(m_{j-1}) \\ \frac{1}{2l}\left[1 - \sum_{x=1}^{\frac{r_{j-1}}{2}-1} P(x)\right] & \text{if } m_{j-1} \in \{r_{j-1} - 1, r_{j-1}\},\ m_j = 1 + \text{odd}(m_{j-1}) \\ \frac{1}{2l}\left[l - \frac{r_{j-1}}{2} - \sum_{x=1}^{\frac{r_{j-1}}{2}-1} P(x)\left(x - \frac{r_{j-1}}{2}\right)\right] & \text{if } m_{j-1} \in \{r_{j-1} - 1, r_{j-1}\},\ m_j \in \{r_j - 1, r_j\} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

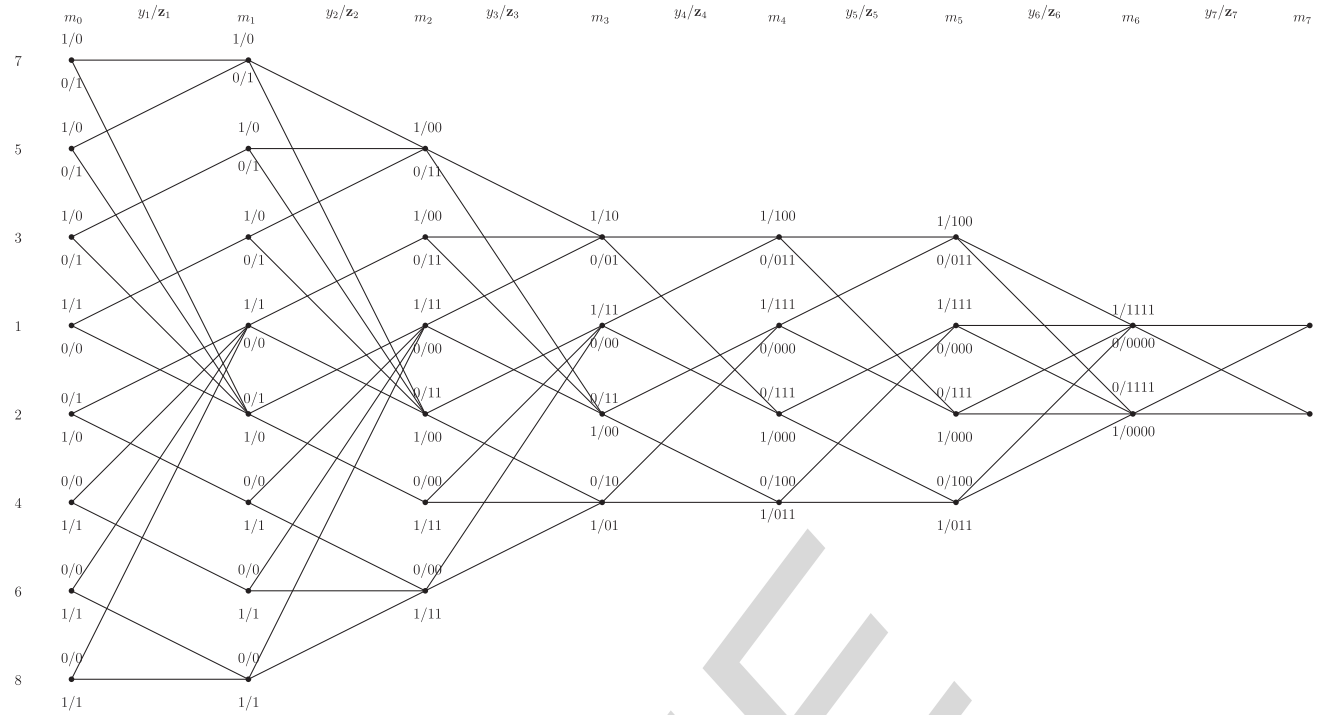


Fig. 2. An example of the proposed irregular UEC trellis, which is obtained by amalgamating seven different UEC trellises. Here, the component UEC codebooks $\mathbb{C}_1 = \{0, 1, 1, 1\}$, $\mathbb{C}_2 = \{0, 1, 1, 1\}$, $\mathbb{C}_3 = \{000, 000, 000\}$, $\mathbb{C}_4 = \{00, 01\}$, $\mathbb{C}_5 = \{000, 011\}$, $\mathbb{C}_6 = \{000, 011\}$ and $\mathbb{C}_7 = \{0000\}$ are employed.

transition probabilities are generalized, allow their application to any IrUEC trellis and to any source probability distribution $P(x)$.

Similar to the regular UEC trellis encoder, the proposed IrTrellis encoder represents each bit y_j in the vector $\mathbf{y}$ by a codeword $\mathbf{z}_j$ comprising n_j bits. This is selected from the corresponding set of $r_j/2$ codewords $\mathbb{C}_j = \{\mathbf{c}^j_1, \mathbf{c}^j_2, \ldots, \mathbf{c}^j_{r_j/2-1}, \mathbf{c}^j_{r_j/2}\}$ or from the complementary set $\overline{\mathbb{C}}_j = \{\overline{\mathbf{c}^j_1}, \overline{\mathbf{c}^j_2}, \ldots, \overline{\mathbf{c}^j_{r_j/2-1}}, \overline{\mathbf{c}^j_{r_j/2}}\}$, which is achieved according to

$$\mathbf{z}_j = \begin{cases} \overline{\mathbf{c}^j_{\lceil m_{j-1}/2 \rceil}} & \text{if } y_j = \text{odd}(m_{j-1}) \\ \mathbf{c}^j_{\lceil m_{j-1}/2 \rceil} & \text{if } y_j \neq \text{odd}(m_{j-1}) \end{cases}. \tag{6}$$

Finally, the selected codewords are concatenated to obtain the bit vector $\mathbf{z} = [z_k]_{k=1}^{b\bar{n}}$ of Fig. 1, where $\bar{n} = \frac{1}{b}\sum_{j=1}^{b} n_j$ is the average codeword length. For example, the path $\mathbf{m} = [1, 3, 5, 3, 2, 1, 1, 2]$ through the irregular UEC trellis of Fig. 2 yields the encoded bit sequence $\mathbf{z} = [1000011111110000]$, which comprises $b\bar{n} = 16$ bits, where we have $\bar{n} = \frac{16}{7}$.

Note that the bit vector $\mathbf{z}$ may be modeled as a specific realization of a vector $\mathbf{Z} = [Z_k]_{k=1}^{b\bar{n}}$ comprising $b\bar{n}$ binary RVs. Observe in Fig. 2 that each of the b component trellis structures in the irregular UEC trellis of the IrTrellis encoder is designed to obey symmetry and to rely on complementary codewords. Hence, bits of the encoded bit vector $\mathbf{Z}$ have equiprobable values, where $\Pr(Z_k = 0) = \Pr(Z_k = 1) = 0.5$, and the bit entropy obeys $H_{Z_k} = H[\Pr(Z_k = 0)] + H[\Pr(Z_k = 1)] = 1$. Owing to this, in contrast to some of the benchmarkers to be considered in Section V, the proposed IrUEC scheme of Fig. 1 does not suffer from additional capacity loss.

We assume that each of the b trellis structures in the proposed irregular UEC trellis is selected from a set of S component UEC trellis structures $\{\text{UEC}^s\}_{s=1}^{S}$, corresponding to a set of S component codebooks $\{\mathbb{C}_s\}_{s=1}^{S}$. More specifically, we assume that each codebook $\mathbb{C}_s$ is employed for generating a particular fraction α_s of the bits in $\mathbf{z}$, where we have $\sum_{s=1}^{S} \alpha_s = 1$. Here, the number of bits generated using the codebook $\mathbb{C}_s$ is given by $b\bar{n} \cdot \alpha_s$. We will in Section IV show that the fractions $\alpha = \{\alpha_s\}_{s=1}^{S}$ may be designed in order to appropriately shape the IrUEC EXIT function. Moreover, the IrUEC coding rate is given by $R_{\text{IrUEC}} = \sum_{s=1}^{S} \alpha_s \cdot R_{\text{UEC}^s}$, where the corresponding coding rate R_{UEC^s} of the regular UEC^s code depends on the codebook $\mathbb{C}_s$ and is given by [10, Eq. (11)].

C. IrURC Encoder and Modulator

As shown in Fig. 1, the IrUEC-encoded bit sequence $\mathbf{z}$ is interleaved in the block π_1 in order to obtain the bit vector $\mathbf{v}$, which is encoded by an IrURC encoder [14], [20] comprising T component URC codes $\{\text{URC}^t\}_{t=1}^{T}$. Unlike our IrUEC code, each component URC code URC^t of the IrURC code employs a separate trellis structure. This is necessary, since the final state of each component URC code has no relation to the initial state of the subsequent component URC code, as described in Section II-B. Therefore, the interleaved IrURC-encoded bit vector $\mathbf{u}$ is decomposed into T sub-vectors $\{\mathbf{u}_t\}_{t=1}^{T}$, each having a length given by $b\bar{n} \cdot \beta_t$, where β_t represents the specific fraction of the bits in $\mathbf{v}$ that are encoded by the component URC^t code, which obeys $\sum_{t=1}^{T} \beta_t = 1$. In Section IV, we also show that the fractions $\beta = \{\beta_t\}_{t=1}^{T}$ may be designed in order to shape the IrURC EXIT function.

In common with each of its T number of component URC codes, the IrURC code has a coding rate of $R_{\text{IrURC}} = 1$, regardless of the particular irregular code design. Owing to this, each of the T number of binary sub-vectors $\{\mathbf{v}_t\}_{t=1}^{T}$ that result from IrURC encoding has the same length as the corresponding sub-vector $\mathbf{u}_t$. The set of these sub-vectors $\{\mathbf{v}_t\}_{t=1}^{T}$ are concatenated to obtain the bit-vector $\mathbf{v}$, which comprises $b\bar{n}$ bits.

Finally, the IrURC-encoded bit vector $\mathbf{v}$ is interleaved by π_2 in order to obtain the bit vector $\mathbf{w}$, which is modulated onto the uncorrelated non-dispersive Rayleigh fading channel using Gray-mapped QPSK. The overall effective throughput of the proposed scheme is given by $\eta = R_{\text{IrUEC}} \cdot R_{\text{IrURC}} \cdot \log_2(M)$, where we have $M = 4$ for QPSK.

III. IrUEC-IrURC Decoder

In this section, we introduce the receiver of the proposed IrUEC-IrURC scheme shown in Fig. 1. In analogy with the IrURC encoder, the IrURC decoder employs T number of component URC decoders $\{\text{URC}^t\}_{t=1}^{T}$, each having a distinct independent trellis structure. By contrast, the IrUEC employs a unary decoder and a novel IrTrellis decoder relying on a single irregular trellis. In Section III-A, the demodulator and the iterative operation of the IrURC and IrUEC decoders will be discussed, while in Sections III-B and III-C we will detail the internal operation of two components of the IrUEC decoder, namely of the IrTrellis decoder and of the unary decoder, respectively.

A. Demodulator and Iterative Decoding

As shown in Fig. 1, QPSK demodulation is employed by the receiver in order to obtain the vector $\tilde{\mathbf{w}}$ of Logarithmic Likelihood Ratios (LLRs), which pertain to the bits in the vector $\mathbf{w}$. This vector is deinterleaved by π_2^{-1} for the sake of obtaining the LLR vector $\tilde{\mathbf{v}}$, which is decomposed into the T sub-vectors $\{\tilde{\mathbf{v}}_t\}_{t=1}^{T}$ that have the same lengths as the corresponding sub-vectors of $\{\mathbf{v}_t\}_{t=1}^{T}$. Here, we assume that a small amount of side information is used for reliably conveying the lengths of all vectors in the IrUEC-IrURC transmitter to the receiver. The sub-vectors $\{\tilde{\mathbf{v}}_t\}_{t=1}^{T}$ are then input to the corresponding component URC decoders $\{\text{URC}^t\}_{t=1}^{T}$ of the IrURC decoder.

Following this, iterative exchanges of the vectors of extrinsic LLRs [21] commences between the Soft-Input Soft-Output (SISO) IrUEC and IrURC decoders. In Fig. 1, the notation $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{z}}$ represent vectors of LLRs pertaining to the bit vectors $\mathbf{u}$ and $\mathbf{z}$, which are related to the inner IrURC decoder and the outer IrUEC decoder, respectively. Additionally, a subscript of this notation denotes the dedicated role of the LLRs, with a, e and p indicating *a priori*, *extrinsic* and *a posteriori* LLRs, respectively.

At the beginning of iterative decoding, the *a priori* LLR vector $\tilde{\mathbf{u}}^{\text{a}}$ is initialised with a vector of zeros, having the same length as the corresponding bit vector $\mathbf{u}$. As shown in the IrURC decoder of Fig. 1, the vector $\tilde{\mathbf{u}}^{\text{a}}$ is decomposed into the T sub-vectors $\{\tilde{\mathbf{u}}_t^{\text{a}}\}_{t=1}^{T}$, which have the same lengths as the corresponding sub-vectors of $\{\mathbf{u}_t\}_{t=1}^{T}$. Together with $\{\tilde{\mathbf{v}}_t^{\text{a}}\}_{t=1}^{T}$, the sub-vectors $\{\tilde{\mathbf{u}}_t^{\text{a}}\}_{t=1}^{T}$ are fed to the corresponding URC decoder URC^t, which then outputs the resulting extrinsic LLR vectors $\{\tilde{\mathbf{u}}_t^{\text{e}}\}_{t=1}^{T}$ by employing the logarithmic Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [22]. These vectors are combined for forming the extrinsic LLR vector $\tilde{\mathbf{u}}^{\text{e}}$ that pertains to the vector $\mathbf{u}$, which is sequentially deinterleaved by the block π_1^{-1} in order to obtain the *a priori* LLR vector $\tilde{\mathbf{z}}^{\text{a}}$ that pertains to the bit vector $\mathbf{z}$. Similarly, the IrTrellis decoder is provided with the *a priori* LLR vector $\tilde{\mathbf{z}}^{\text{a}}$ and generates the vector of extrinsic LLRs $\tilde{\mathbf{z}}^{\text{e}}$, which are interleaved in the block π_1 to obtain the *a priori* LLR vector $\tilde{\mathbf{u}}^{\text{a}}$ that is provided for the next iteration of the IrURC decoder.

B. IrTrellis Decoder

As discussed in Section II, our IrUEC code employs a novel bit-based irregular trellis, while the IrURC code employs a selection of independent trellises. The novel IrTrellis decoder within the IrUEC decoder applies the BCJR algorithm to the irregular trellis. The synchronization between the novel irregular trellis and the unary codewords is exploited during the BCJR algorithm's γ_t calculation of [22, (9)]. This employs the conditional transition probability $\Pr(M_j = m_j | M_{j-1} = m_{j-1})$, where we have

$$P(m_j|m_{j-1}) = \frac{P(m_j, m_{j-1})}{\sum_{\check{m}=1}^{r_j} P(\check{m}, m_{j-1})} \tag{7}$$

and $P(m_j, m_{j-1})$ is given in (5).

Note that the IrUEC decoder will have an EXIT function [23] that reaches the (1, 1) point of perfect convergence to an infinitesimally low Symbol Error Ratio (SER), provided that all component codebooks in the set $\{\mathbb{C}_s\}_{s=1}^{S}$ have a free distance of at least 2 [24], as characterised in Section IV. Since the combination of the IrURC decoder and demodulator will also have an EXIT curve that reaches the (1, 1) point in the top right corner of the EXIT chart, iterative decoding convergence towards the Maximum Likelihood (ML) performance is facilitated [25]. At this point, the IrTrellis decoder may invoke the BCJR algorithm for generating the vector of *a posteriori* LLRs $\tilde{\mathbf{y}}^{\text{p}}$ that pertain to the corresponding bits in the vector $\mathbf{y}$.

C. Unary Decoder

As described in [10], the unary decoder of Fig. 1 sorts the values in the LLR vector $\tilde{\mathbf{y}}^{\text{p}}$ in order to identify the a number of bits in the vector $\mathbf{y}$ that are most likely to have values of zero. A hard decision vector $\hat{\mathbf{y}}$ is then obtained by setting the value of these bits to zero and the value of all other bits to one. Finally, the bit vector $\hat{\mathbf{y}}$ can be unary decoded in order to obtain the symbol vector $\hat{\mathbf{x}}$ of Fig. 1, which is guaranteed to comprise a number of symbols.

IV. Algorithm for the Parametrization of the IrUEC-IrURC Scheme

The performance of the IrUEC-IrURC scheme depends on how well it is parametrized. A good parametrization is one that

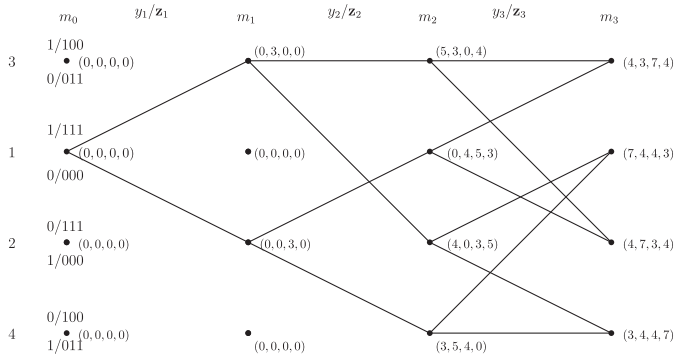


Fig. 3. The legitimate paths through the first three stages in UEC trellis having the codewords $\mathbb{C} = \{000, 011\}$.

results in a narrow but still open EXIT chart tunnel, although achieving this requires a high degree of design freedom, when shaping the IrUEC and IrURC EXIT functions. Therefore, we begin in Section IV-A by characterising the free distance property of the UEC codes and selecting a set of UEC component codes having a wide variety of different inverted EXIT function shapes. This maximises the degree of freedom that is afforded, when matching the IrUEC EXIT function to that of the IrURC code. In Section IV-B, we propose a novel extension to the double-sided EXIT chart matching algorithm of [14], which we employ for jointly matching the EXIT functions of the IrUEC and the IrURC codes. However, in contrast to the algorithm of [14], which does not allow a particular coding rate to be targeted for the IrUEC-IrURC scheme, our algorithm designs both the fractions α and β to achieve a particular target coding rate. In Section V, this will be exploited to facilitate a fair comparison with benchmarkers having particular coding rates.

A. Design of UEC Component Codes

Since an r-state n-bit UEC code is parametrized by a codebook set $\mathbb{C}$ comprising $r/2$ number of codewords each having n bits, there are a total of $2^{n \cdot r/2}$ number of candidates for $\mathbb{C}$. It is neither possible nor necessary to employ all these $2^{n \cdot r/2}$ codebooks as the component codes in our IrUEC code, because some of the codebooks will have identical or similar inverted EXIT function shapes, offering no additional degree of freedom, when performing EXIT chart matching. Therefore, it is desirable to eliminate these candidate codebooks.

The generalised UEC trellis structure associated with the codebook $\mathbb{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r/2-1}, \mathbf{c}_{r/2}\}$ is depicted in [10, Fig. 3(a)]. Note that the upper half and the lower half of the trellis is symmetrical in terms of the output codewords z_j generated in response to a given input bit value y_j, as shown in (6). More specifically, for the states in the upper half of the trellis, the output codewords z_j are selected from the codebook $\mathbb{C}$ when $y_j = 0$, while the codewords from its complementary codebook $\overline{\mathbb{C}} = \{\overline{\mathbf{c}_1}, \overline{\mathbf{c}_2}, \ldots, \overline{\mathbf{c}_{r/2-1}}, \overline{\mathbf{c}_{r/2}}\}$ are selected when $y_j = 1$. For the states in the lower half of the trellis, the output codewords z_j are selected from the codebook $\mathbb{C}$ when $y_j = 1$ and from the complementary codebook $\overline{\mathbb{C}}$ when $y_j = 0$. Intuitively, if any particular subset of the n bits at the same positions within each codeword of $\mathbb{C}$ are inverted, this would not change the distance properties of the output bit vector $\mathbf{z}$, hence resulting in an identical inverted EXIT function. For example, inverting the first bit of each codeword in the codebook $\mathbb{C}_0 = \{00, 01\}$ will give a new codebook $\mathbb{C}_1 = \{10, 11\}$ having an identical EXIT function. Likewise, inverting both bits of the codewords in $\mathbb{C}_0$ will give $\mathbb{C}_2 = \{11, 10\}$, which also has an identical EXIT function. Similarly, swapping any pair of the n bits at the same positions between each pair of codewords will not affect the distance properties or the shape of the inverted EXIT function either. For example, swapping the two bits in the codebook $\mathbb{C}_0$ results in a new codebook $\mathbb{C}_3 = \{00, 10\}$, having an identical inverted UEC EXIT function shape. Therefore, each of these four codebooks, $\mathbb{C}_0, \mathbb{C}_1, \mathbb{C}_2$ and $\mathbb{C}_3$, as well as their conversions created by bit-inversion and swapping, have identical inverted EXIT functions. Consequently, all but one of these codebooks can be eliminated as candidates for the sake of reducing the complexity of EXIT chart matching.

The number of candidate UEC codebooks may be further reduced by characterising their free distance properties. Since no analytic method has been developed for calculating the free distance d_f of a UEC code, we propose a heuristic method for obtaining an approximate measure of d_f. The free distance represents the minimum distance between any pair of encoded bit vectors produced by different paths through the trellis. The total number of possible pairings of paths emerging from a particular state in a UEC trellis of length b is given by $2^{b-1}(2^b - 1)$, which grows exponentially. However, considering the symmetry of a regular UEC trellis, it is possible to use a step-by-step directed search for determining the free distance, rather than using a brute force exhaustive search. Note that in the regular UEC trellis as generalised in [10, Fig. 3(a)], a bit vector $\mathbf{y} = [y_j]_{j=1}^{b}$ identifies a unique path $\mathbf{m} = [m_j]_{j=0}^{b}$ that emerges from state 1 and terminates at either state 1 or 2, hence accordingly identifying a corresponding output bit sequence $\mathbf{z} = [z_k]_{k=1}^{b\bar{n}}$. By exploiting this observation, the free distance d_f can be obtained by computing the Hamming Distance(HD) between each pair of paths and then selecting the pair having the minimum HD, whenever two paths merge at a particular state in the trellis.

When the bit sequence length considered satisfies $b > r/2$, the paths form complete trellis stages, as exemplified in Fig. 3. Therefore, in order to reduce the search complexity, we consider all permutations of the b-bit unary-encoded vector $\mathbf{y}$ bit-by-bit, considering all paths that emerge from state $m_0 = 1$ and terminate at each particular state $m_b = 1, 2, \ldots, r$, on a step-by-step basis. For a pair of states $m_j, m'_j \in \{1, 2, 3, \ldots, r\}$, we define $d^j_{m_j, m'_j}$ as the minimum HD between the set of all paths that terminate at state m_j and the set that ends at state m'_j, given the input bit sequence $[y_1, y_2, \ldots, y_j]$, where $j \in \{0, 1, \ldots, b\}$. Each state m_j is labelled as $(d^j_{m_j,1}, d^j_{m_j,2}, d^j_{m_j,3}, \ldots, d^j_{m_j,r})$, where we have $d^j_{m_j,m'_j} = d^j_{m'_j,m_j}$. For each state $m_0 \in \{1, 2, 3, \ldots, r\}$, the minimum HDs are initialized to 0s. Therefore, the distance $d^j_{m_j,m'_j}$ can be calculated by

$$d^j_{m_j,m'_j} = \min_{m_{j-1},m'_{j-1}} \left[d^{j-1}_{m_{j-1},m'_{j-1}} + h(\mathbf{z}_{m_{j-1},m_j}, \mathbf{z}_{m'_{j-1},m'_j}) \right]. \quad (8)$$

Here, $\mathbf{z}_{m_{j-1},m_j}$ is the codeword in the set $\mathbb{C}$ or in the complementary set $\overline{\mathbb{C}}$ that is generated by the transition from state m_{j-1} to state m_j, while the function $h(\cdot,\cdot)$ denotes the HD between the two operands. Owing to this, our method conceived for determining the free distance of a UEC code has a complexity order of $O[b \cdot r(r-1)]$, where r is the number of states in the trellis and b is the length of the bit vector $\mathbf{y}$ considered. Let $\mathbb{Y}_{b_1}$ be the bit sequence set associated with the set of all paths $\mathbb{M}_{b_1}$ having a length of b_1, while $\mathbb{Y}_{b_2}$ is the bit sequence set associated with the path set $\mathbb{M}_{b_2}$ having a length of b_2. Therefore, all sequences in $\mathbb{Y}_{b_1}$ are prefix of sequences in $\mathbb{Y}_{b_2}$, when we have $b_1 < b_2$. For example, when $b_1 = 2$ and $b_2 = 3$, the bit sequence $\mathbf{y}_2 = \{111011\}$ is a prefix of the bit sequence $\mathbf{y}_3 = \{111011111\}$, where $\mathbf{y}_2$ is associated with the path vector $\mathbf{m}_2 = \{1, 3, 2\}$ and $\mathbf{y}_2$ is associated with the path vector $\mathbf{m}_2 = \{1, 3, 2, 1\}$, respectively. Note that according to [26, Lemma 1], the minimum HD $d_f(\mathbb{Y}_{b_1})$ among all bit sequences in $\mathbb{Y}_{b_1}$ is an upper bound on the minimum HD $d_f(\mathbb{Y}_{b_2})$ of $\mathbb{Y}_{b_2}$, when we have $b_1 < b_2$. Owing to this, the approximate free distance d_f calculated using our method converges to the true free distance, as the lengths of the paths considered are extended towards infinity. In our experiments, we considered bit vector lengths of up to $b = 10r$. In all cases, we found that the free distance has converged before that point, regardless of how the UEC code is parametrised, owing to the common features of all UEC codes described in Section II-B."

For example, Fig. 3 shows all of the legitimate paths through an $r = 4$-state trellis employing the codebook $\mathbb{C} = \{000, 011\}$ that may be caused by the first three bits in a bit vector $\mathbf{y} = \{y_j\}_{j=1}^{b}$, having a length $b > 3$. Particularly, the minimum HD $d_{2,3}^1$ between states $m_1 = 2$ and $m_1' = 3$ is given by $d_{2,3}^1 = d_{1,1}^0 + h(111, 000) = 3$. Since there are no legitimate paths leading to the states $m_1 = 1$ or $m_1 = 4$, we do not update the associated distances, as shown in Fig. 3. Similarly, we have $d_{1,2}^2 = d_{2,3}^1 + h(111, 011) = 4$, and $d_{1,2}^3 = \min(d_{1,2}^2 + h(000, 111), d_{1,4}^2 + h(000, 100), d_{2,3}^2 + h(111, 011), d_{3,4}^2 + h(011, 100)) = 4$. Once the forward recursion has considered a sufficient number of trellis stages for $\min(d_{1,1}^j, d_{1,2}^j, d_{2,2}^j) = \min(d_{1,1}^{j-1}, d_{1,2}^{j-1}, d_{2,2}^{j-1})$, then the approximate free distance becomes $d_f = \min(d_{1,1}^j, d_{1,2}^j, d_{2,2}^j)$.

Our set of candidate component UEC codes was further reduced by considering their free distances. More specifically, in order to achieve a wide variety of EXIT function shapes, we retained only UEC codebooks having the maximal or minimal free distances for each combination of $n \in \{2, 3, 4\}$ and $r \in \{2, 4\}$, where a free distance of 3 is the minimal value that facilitates convergence to the $(1, 1)$ point [24] and avoids an error floor. We drew the EXIT functions for all remaining candidate component UEC codes and selected the five codebooks offering the largest variety of EXIT function shapes, as listed in Table II. Our experiments revealed that only insignificant EXIT function shape variations are obtained, when considering more than $r = 4$ states. Without loss of generality, our irregular trellis example of Fig. 2 is constructed by concatenating the five UEC codebooks of Table II. In the following simulations, we will consider irregular trellises that are constructed using these codebooks. However, the number of states r employed by our five UEC component codes can be optionally and independently increased in the receiver, in order to facilitate nearer-to-capacity operation at the cost of an increased decoding complexity [10]. This is achieved by repeating the last element in the codebook. For example, while the transmitter may use the codebook $\mathbb{C} = \{00, 01\}$, the receiver may extend this to the $r = 10$-state codebook $\mathbb{C} = \{00, 01, 01, 01, 01\}$. Fig. 4 plots the inverted EXIT functions of the component UEC codes $\{\text{UEC}^s\}_{s=1}^5$, when extended to $r = 10$ states. Note that, similar to the IrURC EXIT function, the composite IrUEC EXIT function f_{IrUEC} is given as a weighted average of the component EXIT functions $\{f_{\text{UEC}^s}\}_{s=1}^5$, where we have

$$f_{\text{IrUEC}} = \sum_{s=1}^{5} \alpha_s \cdot f_{\text{UEC}^s}. \tag{9}$$

TABLE II
AFTER INVERTING AND SWAPPING, WE SELECT THE IRUEC COMPONENT UEC CODEBOOKS $\{\mathbb{C}_s\}_{s=1}^5$ WITH n BITS AND r STATES BOTH UP TO 4. ALL THE CODEBOOKS ARE IN THE FORMAT $(\mathbb{C}_s, d_f)$, WHERE d_f IS THE APPROXIMATE FREE DISTANCE

n	$r = 2$	$r = 4$
2		({00, 01}, 3)
3	({000}, 3)	({000, 011}, 4)
4	({0000}, 4)	({0000, 0111}, 5)

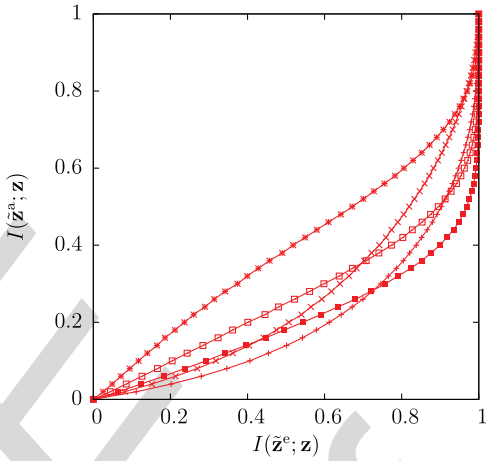


Fig. 4. Inverted EXIT functions for the $S = 5$ component UEC codes $\{\text{UEC}^s\}_{s=1}^5$ of Table II, when extended to $r = 10$ states codebooks, and when the symbol values obey a zeta probability distribution having the parameter value $p_1 = 0.797$.

B. Double-Sided EXIT Chart Matching Algorithm

The sixth column of Table III provides the specific E_b/N_0 values, where the DCMC capacity becomes equal to the throughput η of each scheme considered. These E_b/N_0 values represent the *capacity bound*, above which it is theoretically possible to achieve reliable communication. Note that the capacity bound is a function of the overall effective throughput η of the proposed IrUEC scheme, as described in Section II-C. In turn, the overall effective throughput η depends on the

TABLE III
CHARACTERISTICS OF THE VARIOUS SCHEMES CONSIDERED, INCLUDING OUTER CODING RATE $R_{\rm o}$, INNER CODING RATE $R_{\rm I}$ AND EFFECTIVE THROUGHPUT η. $E_{\rm B}/N_0$ BOUNDS ARE GIVEN FOR THE CASE OF GRAY-CODED QPSK TRANSMISSION OVER AN UNCORRELATED NARROWBAND RAYLEIGH FADING CHANNEL. COMPLEXITY IS QUANTIFIED BY THE AVERAGE NUMBER OF ACS OPERATIONS INCURRED PER DECODING ITERATION AND PER BIT IN THE VECTOR $\mathbf{z}$

1	2	3	4	5	6	7	8	9
Scheme	Codebooks	$R_{\rm o}$	$R_{\rm i}$	η	$E_{\rm b}/N_0$ [dB] capacity bound	$E_{\rm b}/N_0$ [dB] area bound	$E_{\rm b}/N_0$ [dB] tunnel bound	Complexity
IrUEC-IrURC	$\{\text{UEC}^s\}_{s=1}^{5}$	0.254	1	0.508	-0.05	0.21	0.3	258
IrUEC(med)-IrURC						0.30	0.6	192
IrUEC(low)-IrURC						1.14	1.2	157
UEC-IrURC	{000, 011}					0.49	1.7	120
EG-IrCC-IrURC	$\{\mathbf{CC}_{\rm sys}^s\}_{s=1}^{13}$	0.667	0.576			1.72	2.0	341
	$\{\mathbf{CC}_{\rm ns}^s\}_{s=1}^{11}$	0.254	1			1.02	1.1	146
EG-CC-IrURC	([4,7,7],6,6)					1.00	2.2	132

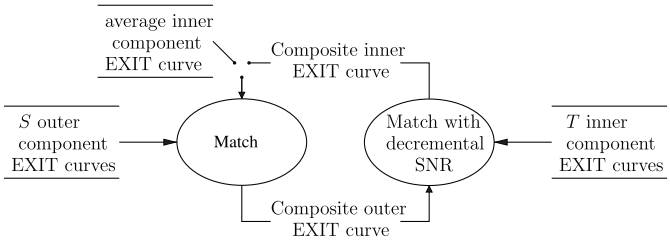


Fig. 5. Data-flow diagram of the proposed double-sided EXIT chart matching algorithm.

IrUEC coding rate $R_{\rm IrUEC}$, which depends on the entropy of the zeta distribution H_X, as described in Section II-A. In order to facilitate the creation of an open EXIT chart tunnel, it is necessary, but not sufficient, for the area $A_{\rm o}$ beneath the inverted outer EXIT function to exceed the area $A_{\rm i}$ beneath the inner EXIT function [17]. Therefore, the *area bound* provides the $E_{\rm b}/N_0$ values where we have $A_{\rm o} = A_{\rm i}$, which would theoretically allow the creation of an open EXIT chart tunnel [27], if the outer and inner EXIT functions were shaped to match each other. Here, $A_{\rm o}$ and $A_{\rm i}$ are the areas beneath the outer and inner EXIT functions, respectively. Depending on how well the EXIT functions match each other, a narrow but open EXIT chart tunnel can only be created at a specific $E_{\rm b}/N_0$ value, which we refer to as the *tunnel bound*. Based on these observations, the $E_{\rm b}/N_0$ difference between the capacity bound and the area bound quantifies the capacity loss that is mitigated by JSCC, while the difference between the area bound and the tunnel bound quantifies the capacity loss that is mitigated by irregular coding [28]. Based on this observation, our double-sided EXIT chart matching algorithm may be iteratively applied in order to match a pair of composite outer and inner EXIT functions, which are formed as a combination of S component UEC EXIT functions and T constituent URC EXIT functions, where the latter depend on the $E_{\rm b}/N_0$ value of the channel. In this way, a narrow but open EXIT chart tunnel between the inverted IrUEC EXIT function and the inner IrURC EXIT function may be created at $E_{\rm b}/N_0$ values that approach the capacity and area bounds, hence avoiding capacity loss and facilitating near-capacity operation.

As depicted in the data-flow diagram of Fig. 5, the algorithm commences by selecting the fractions α, in order to yield an IrUEC code design having a particular coding rate $R_{\rm IrUEC}$ and a composite IrUEC EXIT function that is shaped to match the average of T URC EXIT functions that correspond to a particular $E_{\rm b}/N_0$ value. The technique of [14] may be employed for selecting the fractions β, in order to yield a composite IrURC EXIT function that is shaped to match that of the IrUEC code. Following this, the algorithm alternates between the matching of the composite IrUEC EXIT function to the composite IrURC EXIT function and vice versa, as shown in Fig. 5. In order to facilitate near-capacity operation, we use a 0.1 dB $E_{\rm b}/N_0$ decrement per iteration for the component URC EXIT functions, when designing the fractions β for the IrURC code, until we find the lowest $E_{\rm b}/N_0$ value that achieves a marginally open EXIT tunnel. Note that the double-sided EXIT chart matching algorithm allows the design of an IrUEC code having a specific coding rate $R_{\rm IrUEC}$. This enables us to design the IrUEC code to have a coding rate of $R_{\rm IrUEC} = 0.254$, which provides a fair performance comparison with the regular UEC-IrURC scheme of [10] and with other benchmarkers, as detailed in Section V. More specifically, this results in the same overall effective throughput of $\eta = R_{\rm IrUEC} \cdot R_{\rm IrURC} \cdot \log_2(M) = 0.508$ bit/s/Hz, as listed in Table III.

For the IrURC encoder, we employ the $T = 10$-component URC codes $\{\text{URC}^t\}_{t=1}^{10}$ of [20], [29]. After running the double-sided EXIT chart matching algorithm of Fig. 5 until the $E_{\rm b}/N_0$ value cannot be reduced any further without closing the EXIT chart tunnel, the composite EXIT functions of the IrUEC and IrURC schemes are obtained, as depicted in Fig. 6(a). Here, the $E_{\rm b}/N_0$ value is 0.3 dB, which is 0.35 dB away from the DCMC capacity bound of -0.05 dB and was found to be the lowest one that creates an open EXIT chart tunnel. More specifically, the fractions of the bit vector $\mathbf{z}$ that are generated by the constituent UEC codes $\{\text{UEC}^s\}_{s=1}^{5}$ of the IrUEC encoder are $\alpha = [0\ 0.7240\ 0.0924\ 0\ 0.1836]$, respectively. Similarly, the fractions of the bit vector $\mathbf{u}$ that encoded by the constituent URC codes $\{\text{URC}^t\}_{t=1}^{10}$ of the IrURC encoder are $\beta = [0.1767\ 0\ 0.8233\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively.

V. BENCHMARKERS AND SIMULATIONS

In this section, we compare the SER performance of the proposed IrUEC-IrURC scheme of Fig. 1 to that of various SSCC and JSCC benchmarkers. As mentioned in Section IV, the proposed IrUEC-IrURC scheme and all benchmarkers are designed to have the same effective overall throughput of

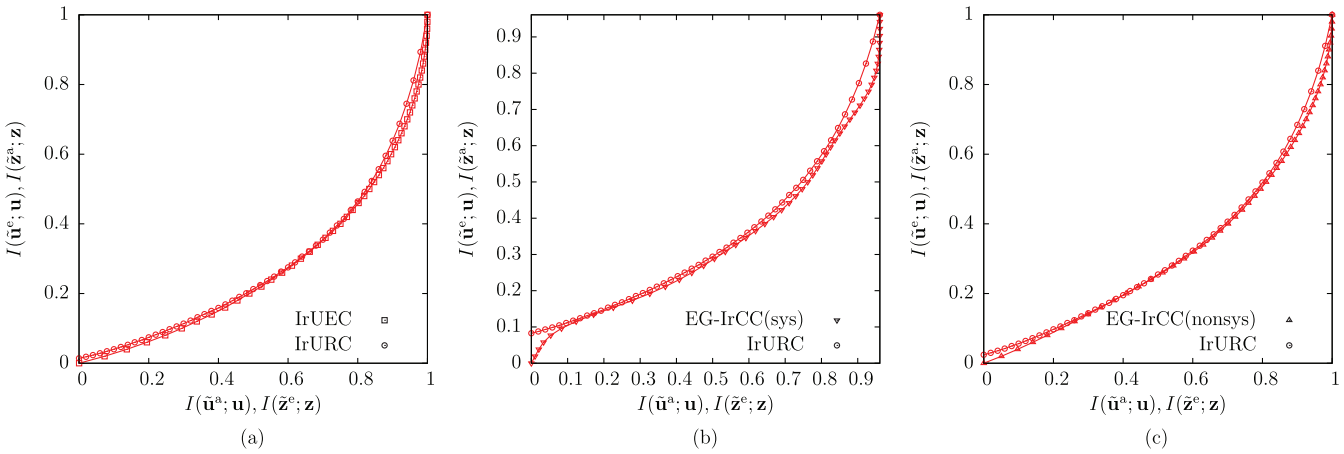


Fig. 6. Composite EXIT functions of (a) the IrUEC decoder employing $S = 5$ component UEC codes $\{\text{UEC}^s\}_{s=1}^{5}$, (b) the EG-IrCC decoder employing the $S = 13$ component recursive systematic CC codes $\{\mathbf{CC}_{\text{sys}}^s\}_{s=1}^{13}$ and (c) the EG-IrCC scheme employing the $S = 11$ component non-systematic CC codes $\{\mathbf{CC}_{\text{ns}}^s\}_{s=1}^{11}$, and the IrURC scheme employing the $T = 10$ component URC codes $\{\text{URC}^t\}_{t=1}^{10}$, when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel. The EXIT chart tunnel is marginally open when $E_b/N_0 = 0.3$, 2.0 and 1.1 dB, respectively.

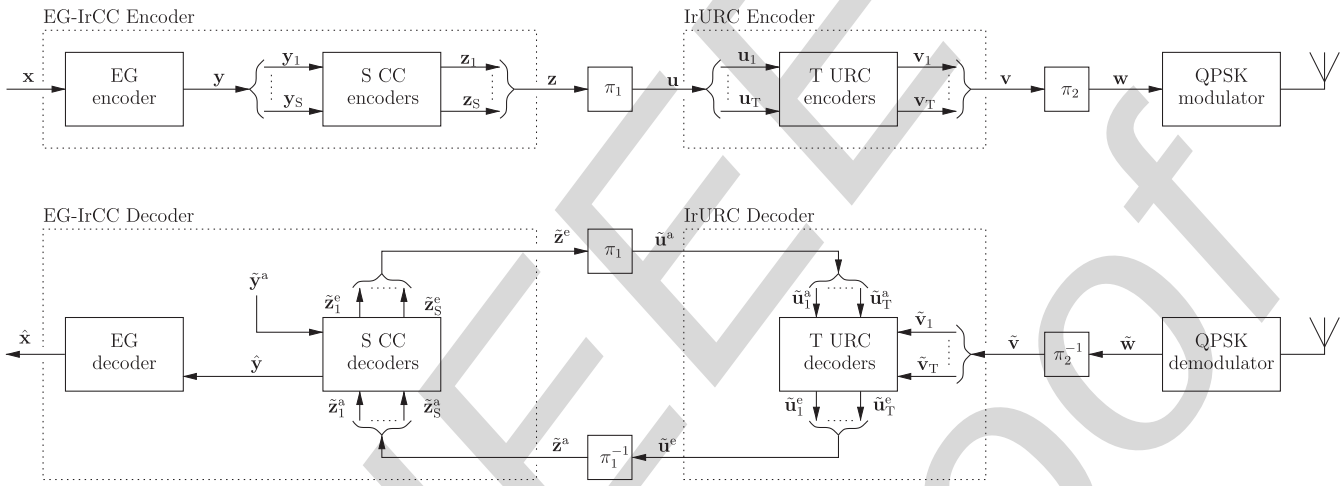


Fig. 7. Schematic of the EG-IrCC-IrURC benchmarker, in which an EG-IrCC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

$\eta = 0.508$ bit/s/Hz, for the sake of fair comparison. A pair of benchmarkers are constituted by the UEC-IrURC and EG-CC-IrURC schemes of our previous work [10]. Furthermore, a new benchmarker is created by replacing the unary encoder and the IrTrellis encoder in the transmitter of Fig. 1 with an EG encoder and an IrCC encoder, respectively. This results in the SSCC benchmarker of Fig. 7, which we refer to as the EG-IrCC-IrURC scheme. Table I shows the first ten codewords of the EG code, which are used for encoding the symbol vector $\mathbf{x}$.

As in the IrUEC-IrURC scheme, the bit vector $\mathbf{y}$ output by the EG encoder may be modeled as a realization of vector $\mathbf{Y} = [Y_j]_{j=1}^{b}$ having binary RVs. However, as observed in [10], these RVs do not adopt equiprobable values $\Pr(Y_j = 0) \neq \Pr(Y_j = 1)$, hence giving a less than unity value for the corresponding bit entropy H_{Y_j}. Similarly, the bit vector $\mathbf{z}$ of Fig. 7 may be modeled as a particular realization of a vector $\mathbf{Z} = [Z_k]_{k=1}^{b\bar{n}}$ comprising $b\bar{n}$ binary RVs. Each binary RV Z_k adopts the values 0 and 1 with the probabilities $\Pr(Z_k = 0)$ and $\Pr(Z_k = 1)$ respectively, corresponding to a bit entropy of H_{Z_k}. In the case where the IrCC code employs systematic component codes, the bits of $\mathbf{y}$ having the entropy $H_{Y_j} < 1$ will appear in $\mathbf{z}$, resulting in a bit entropy of $H_{Z_k} < 1$. However, a bit entropy of $H_{Z_k} < 1$ is associated with a capacity loss, as described in [10].

Hence, for the sake of avoiding any capacity loss, it is necessary to use non-systematic recursive component codes, so that the bits in the resultant encoded vector $\mathbf{z}$ have equiprobable values [10]. In order to demonstrate this, we introduce two versions of the EG-IrCC-IrURC benchmarker. Firstly, the $N = 13$ recursive systematic component CC codes [15] $\{\mathbf{CC}_{\text{sys}}^s\}_{s=1}^{13}$ that were originally proposed for IrCC encoding are adopted in the EG-IrCC-IrURC encoder, as it will be described in Section V-A. Secondly, Section V-B employs the $S = 11$ non-systematic recursive CC codebooks $\{\mathbf{CC}_{\text{ns}}^s\}_{s=1}^{11}$ proposed in [20], in order to offer an improved version of the EG-IrCC benchmarker. Meanwhile, the 10 component URC codebooks $\{\text{URC}^t\}_{t=1}^{10}$ employed by the IrURC encoder in both versions of the benchmarker of Fig. 7 are identical to those in the IrURC encoder of Fig. 1.

A. Recursive Systematic Component CC Codes

The recursive systematic CC codes $\{\mathbf{CC}^s_{\text{sys}}\}^{13}_{s=1}$ employed in [15] were designed to have coding rates of $R_{\text{CC}^s_{\text{sys}}} \in \{0.1, 0.15, \ldots, 0.65, 0.7\}$. However, since the EG-encoded bits in the vector $\mathbf{y}$ are not equiprobable, none of the systematic bits in the bit vector $\mathbf{z}$ will be equiprobable either. As a result, the coding rate $R_{\text{CC}^s_{\text{sys}}} = \frac{H_{Y_j}}{n_{\text{CC}^s_{\text{sys}}} \cdot H_{Z_k}^{\text{CC}^s_{\text{sys}}}}$ of each systematic CC will be lower than the above-mentioned values. Since each CC code $\mathbf{CC}^s_{\text{sys}}$ produces a different number of systematic bits, each will have a different bit entropy $H_{Z_k}^{\text{CC}^s_{\text{sys}}}$, and the EXIT function of each CC code will converge to a different point $(H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}, H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}})$ in the EXIT chart [30]. The composite IrCC EXIT function will converge to a point $(H_{Z_k}^{\text{IrCC}}, H_{Z_k}^{\text{IrCC}})$, where $H_{Z_k}^{\text{IrCC}}$ is given by a weighted average of $\{H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}\}^{13}_{s=1}$, according to

$$H_{Z_k}^{\text{IrCC}} = \sum_{s=1}^{13} \alpha_s \cdot H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}. \tag{10}$$

Since the vector $\mathbf{z}$ is interleaved to generate the bit vector $\mathbf{u}$ as the input of the IrURC encoder, the IrURC EXIT function will also converge to $(H_{Z_k}^{\text{IrCC}}, H_{Z_k}^{\text{IrCC}})$. However, this presents a particular challenge, when parametrizing the fractions α and β of the EG-IrCC(sys)-IrURC scheme. More specifically, the fractions α vary as our double-sided EXIT chart matching algorithm progresses, causing the entropy $H_{Z_k}^{\text{IrCC}}$ to vary as well. This in turn causes the IrURC EXIT function to vary, creating a cyclical dependency that cannot be readily resolved. More specifically, the fractions α must be selected to shape the EG-IrCC EXIT function so that it matches the IrURC EXIT function, but the IrURC EXIT function depends on the fractions α selected for the EG-IrCC EXIT function.

Owing to this, we design the fractions α and β by assuming that the bits of $\mathbf{y}$ are equiprobable and by plotting the inverted EXIT functions for the $S = 13$ recursive systematic CC codes accordingly, giving convergence to the $(1, 1)$ point in Fig. 6(b). Then we invoke our double-sided EXIT matching algorithm to design the fractions α and β for the IrCC(sys) and IrURC codes, which we apply to the EG-IrCC(sys)-IrURC scheme. For the case where the bits of the vector $\mathbf{y}$ have the non-equiprobable values that result from EG encoding, the composite EXIT functions are shown in Fig. 6(b). Here, the effective throughput is $\eta = 0.508$ bit/s/Hz and the E_b/N_0 value is 2.0 dB, which is the lowest value for which an open EXIT chart tunnel can be created. This E_b/N_0 tunnel bound is 2.05 dB away from the DCMC capacity bound of -0.05 dB, owing to the above-mentioned capacity loss. Furthermore, the EG-IrCC(sys)-IrURC scheme has an area bound of 1.72 dB, which corresponds to a capacity loss of 1.77 dB, relative to the capacity bound. The designed fractions for the EG-IrCC scheme are $\alpha = [0.0620\ 0.2997\ 0.0497\ 0.0004\ 0.1943\ 0\ 0.0984\ 0.1285\ 0\ 0\ 0\ 0.0002\ 0.1668]$, while the fractions for the IrURC code are $\beta = [0.6548\ 0\ 0.3452\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively.

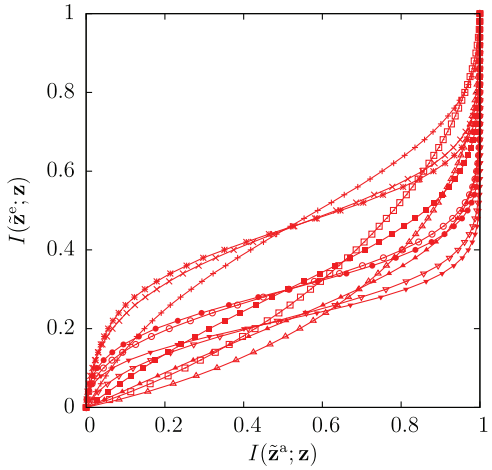


Fig. 8. Inverted EXIT functions for EG-CC code, for the case where the $S = 11$ component recursive non-systematic CC codes $\{\mathbf{CC}^s_{\text{ns}}\}^{11}_{s=1}$ are employed, and the symbol values obey a zeta probability distribution having the parameter value $p_1 = 0.797$.

B. Recursive Non-Systematic Component CC Codes

In order to avoid the capacity loss introduced by the recursive systematic CC codes, we advocate the recursive non-systematic CC codebooks $\{\mathbf{CC}^s_{\text{ns}}\}^{11}_{s=1}$, which are described by the generator and feedback polynomials provided in [10, Table II]. More specifically, of the 12 codes presented in [10, Table II], we use all but the $r = 2$, $n = 2$ code, for the sake of avoiding an error floor. These recursive non-systematic CC codes attain the optimal distance properties [31] subject to the constraint of producing equiprobable bits $\Pr(Z_j = 0) = \Pr(Z_j = 1)$, which is necessary for avoiding any capacity loss. The inverted EXIT functions are plotted in Fig. 8.

For the sake of a fair comparison, we apply the double-sided EXIT chart matching algorithm of Fig. 5 again to design the EG-IrCC(nonsys)-IrURC scheme having a coding rate of $R_{\text{EG-IrCC}} = 0.254$ and an effective throughput of $\eta = 0.508$ bit/s/Hz. The composite EXIT functions of the EG-IrCC(nonsys) and IrURC schemes are shown in Fig. 6(c). Here, the fractions of the EG-IrCC scheme are $\alpha = [0.8101\ 0\ 0.0643\ 0\ 0\ 0\ 0\ 0.1256\ 0\ 0\ 0]$, while the fractions of the IrURC code are $\beta = [0.2386\ 0\ 0.7614\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively. The EXIT chart of Fig. 8 is provided for an E_b/N_0 value of 1.1 dB, which is the lowest value for which an open EXIT chart tunnel is created. As shown in Table III, this E_b/N_0 tunnel bound is just 1.15 dB away from the DCMC capacity bound of -0.05 dB. This improvement relative to the EG-IrCC(sys)-IrURC scheme may be attributed to the non-systematic nature of the EG-IrCC(nonsys)-IrURC scheme, which has reduced the capacity loss to 1.07 dB, as quantified by considering the difference between the E_b/N_0 area bound of 1.02 dB and the capacity bound.

C. Parallel Component UEC Codes

In order to make a comprehensive comparison, we also consider a Parallel IrUEC-IrURC scheme. As shown in Fig. 9, this scheme employs a parallel concatenation of S number

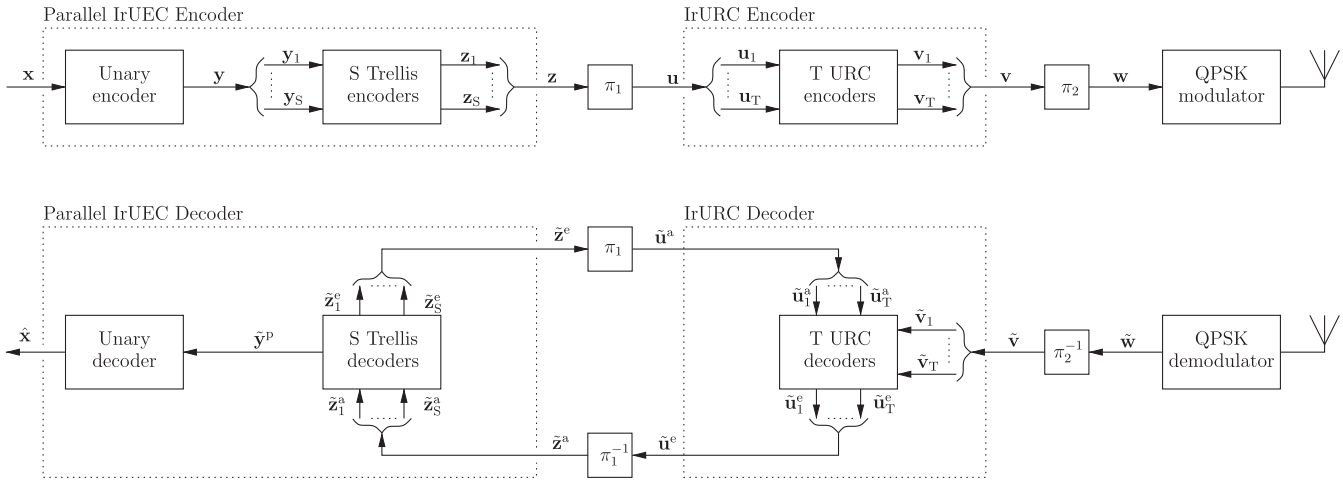


Fig. 9. Schematic of the Parallel IrUEC-IrURC benchmarker, in which a parallel IrUEC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

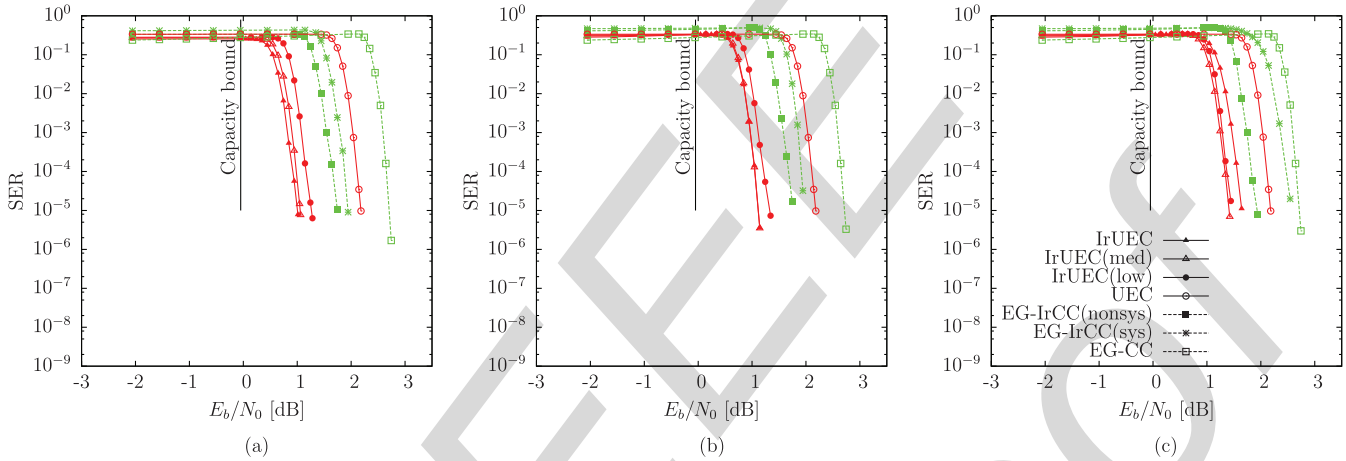


Fig. 10. SER performance for various arrangements of the proposed IrUEC-IrURC scheme of Fig. 1, the EG-IrCC-IrURC of Fig. 7, the Parallel IrUEC-IrURC scheme of Fig. 9, as well as the UEC-IrURC and the EG-IrURC schemes of [10], when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values. A complexity limit of (a) unlimited, (b) 10,000 and (c) 5,000 ACS operations per decoding iteration is imposed for decoding each of the bits in $\mathbf{z}$.

of separate UEC trellis encoders to encode the bit vector $\mathbf{y}$, in analogy with the structure of the EG-IrCC scheme. More specifically, the component UEC codes of the Parallel IrUEC encoder are selected from the five constituent codes provided in Table II, while the component UEC codes of the Parallel IrUEC decoder are extended to $r = 10$ states. The irregular fractions employed by the Parallel IrUEC scheme are the same as those used in our proposed IrUEC scheme. However, in order for each component UEC trellis encoder to remain synchronized with the unary codewords in the bit vector $\mathbf{y}$, it is necessary for each component trellis to commence its encoding action from state $m_0 = 1$ and end at state $m_b = 1$ or $m_b = 2$. Owing to this, the subvectors of $\mathbf{y}$ input to each component UEC must comprise an integer number of complete unary codewords. The irregular coding fractions can only be controlled at the symbol level in the case of the parallel IrUEC scheme, rather than at the bit level, as in the proposed IrUEC scheme. Therefore, the corresponding EXIT chart of the parallel IrUEC scheme is not guaranteed to have an open tunnel, when the E_b/N_0 value approaches the tunnel bound of Table III, hence resulting in a degraded SER performance. However, if the frame length a was orders of magnitude higher, the difference between the symbol-based and bit-based segmentations of the bit vector $\mathbf{y}$ would become insignificantly small. As a result, a similar SER performance may be expected for the parallel IrUEC scheme in this case. In the following section, we will compare the performances of the Parallel IrUEC and the proposed IrUEC schemes, using different values for the frame length a.

D. SER Results

The SER performance of the IrUEC-IrURC, the EG-IrCC(sys)-IrURC and the EG-IrCC(nonsys)-IrURC, UEC-IrURC and EG-CC-IrURC schemes is characterised in Fig. 10. In each case, the source symbol sequence $\mathbf{x}$ comprises $a = 10^4$ symbols, the values of which obey a zeta distribution having a parameter value of $p_1 = 0.797$. As shown above, the parametrizations of the irregular codes in each scheme are

designed to achieve the closest possible matching of EXIT charts, while giving the same overall effective throughput of $\eta = 0.508$ bit/s/Hz. Transmission is performed over a Gray-coded QPSK-modulated uncorrelated narrowband Rayleigh fading channel, resulting in the DCMC capacity bound of -0.05 dB. We select two parametrizations of the schemes in [10] to create two of our four benchmarkers, namely the $r = 4$-state UEC-IrURC and the $r = 4$-state EG-CC-IrURC schemes. Note that the $r = 4$-state EG-CC-IrURC scheme was found to outperform other parametrizations of the same scheme having higher number of states, owing to its superior EXIT chart matching accordingly. With the same effective throughput η, a fair comparison is provided between our proposed IrUEC-IrURC scheme and the four benchmarkers.

Note that the practical implementation of the time-variant IrTrellis used in our IrUEC-IrURC scheme follows the same principles as the parallel time-invariant trellises of the benchmarker schemes, such as the EG-IrCC-IrURC scheme and the regular UEC-IrURC scheme. Once the irregular coding fractions have been determined, the specific portions of message that should be encoded and decoded by the corresponding trellises are also determined. In both time-variant and parallel time-invariant trellises, the hardware is required to support different trellis structures, which may be implemented by appropriately changing the connections among the states of a single hardware implementation of a trellis. Although the proposed time-invariant trellis has some peculiarities at the interface between its different sections, these can also be implemented using the same hardware at either side of the interface. As an example platform for hardware implementation, the computation unit of [32] performs one ACS arithmetic operation per clock cycle, which are the fundamental operations used in BCJR decoders [18]. Therefore, the implementational complexity depends only on the computational complexity, as quantified per decoding iteration in Table III. Since a common computational complexity limit is used in our comparisons of the various schemes, they can be deemed to have the same implementational complexity. Although the routing and control of the proposed IrTrellis may be expected to be more complicated than in the parallel time-invariant trellises of the benchmarkers, it may be expected that the associated overhead is negligible compared to the overall implementational complexity.

As shown in Table III, our IrUEC-IrURC scheme imposes a complexity of 258 ACS operations per iteration per bit, when employing $r = 10$ states for each component UEC code in the IrTrellis decoder. We also consider alternative parametrizations of our IrUEC-IrURC scheme, which employ an IrTrellis having fewer states, in order to achieve lower complexities. The IrUEC(med)-IrURC scheme relies on $r = 6$ trellis states for different stages of the IrTrellis, which results in a total complexity of 192 ACS operations per iteration per bit. This matches that of the UEC-IrURC benchmarker. At the same time, the IrUEC(low)-IrURC scheme employs the minimal number of states for each stage of the IrTrellis, namely either $r = 4$ states, as listed in Table II, hence resulting in a complexity of 157 ACS operations per iteration per bit.

During the simulation of each scheme, we recorded both the SER and the complexity incurred after each decoding iteration, resulting in a 3D plot of SER versus E_b/N_0 and versus complexity. Fig. 10 presents 2D plots of SER versus E_b/N_0 relationship, which were obtained by slicing through these 3D plots at a particular complexity. More specifically, we select the complexity limits of $10,000$ and $5,000$ ACS operations per iteration per bit in Fig. 10(b) and (c), respectively. Meanwhile, Fig.10 (a) characterizes the SER performance achieved after iterative decoding convergence, regardless of the complexity.

As shown in Table III, the proposed IrUEC-IrURC scheme has an area bound of 0.21 dB, which is the E_b/N_0 value where the area A_o beneath the inverted IrUEC EXIT function equals that beneath the IrURC EXIT function. Although the UEC-IrURC benchmarker has a similar area bound of $E_b/N_0 = 0.49$ dB, it has an inferior EXIT chart matching capability owing to its employment of regular UEC constituent codes. By contrast, the employment of two irregular codes in the proposed IrUEC-IrURC scheme facilitates an open EXIT chart tunnel at an E_b/N_0 value of 0.3 dB, which is 1.4 dB lower than the open tunnel bound of the UEC-IrURC benchmarker. Note that the area and tunnel bounds are degraded in the context of the lower complexity versions of the proposed IrUEC-IrURC scheme, which have fewer states in the IrTrellis. This may be explained by the increased capacity loss encountered when the number of UEC states is reduced [10]. Note however that even with a reduced complexity, the proposed IrUEC-IrURC scheme tends to exhibit superior area and tunnel bounds, when compared to the EG-IrCC-IrURC and EG-CC-IrURC benchmarkers, as shown in Table III. This may be attributed to the large capacity loss that is associated with SSCC scheme [10].

Fig. 10 demonstrates that our proposed IrUEC-IrURC scheme has a superior SER performance compared to all other benchmarkers, regardless of which complexity limit is selected in this particular scenario. For example, as shown in Fig. 10(a), our IrUEC-IrURC scheme facilitates operation within 0.4 dB of the capacity bound, offering a 0.8 dB gain compared to the EG-IrCC(nonsys)-IrURC scheme, which is the best-performing of the SSCC benchmarkers. This is achieved without any increase in transmission energy, bandwidth, transmit duration or decoding complexity. Note that the EG-IrCC(nonsys)-IrURC benchmarker offers a 0.9 dB gain over the EG-IrCC(sys)-IrURC benchmarker, which is owing to the capacity loss that is associated with systematic IrCC component codes. As expected, the reduced complexity versions of the proposed IrUEC-IrURC scheme exhibit a degraded SER performance. However, the IrUEC(low)-IrURC scheme can be seen to offer up to 0.5 dB gain over the UEC-IrURC benchmarker, which has a close decoding complexity per bit per iteration. Since the Parallel IrUEC-IrURC scheme can only provide a symbol-level control of the irregular coding fractions, the EXIT chart tunnel is not guaranteed to be open at low E_b/N_0 values. As a result, Fig. 11 shows that the Parallel IrUEC-IrURC scheme of Fig. 9 performs relatively poorly compared to the proposed IrUEC-IrURC scheme, particularly when the frame length has values of $a = 10^2$ and $a = 10^3$ symbols. Note that this performance gain offered by the proposed scheme is obtained without imposing any additional decoding complexity and without requiring any additional transmission-energy, -bandwidth, or -duration. In analogy with Fig. 10(a), an additional set of SER results

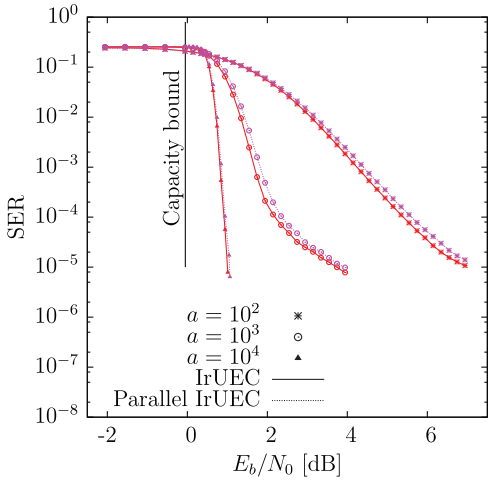


Fig. 11. SER performance for various frame lengths $a \in \{10^2, 10^3, 10^4\}$ of the proposed IrUEC-IrURC scheme of Fig. 1 and the Parallel IrUEC-IrURC scheme of Fig. 9, when conveying symbols obeying a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values.

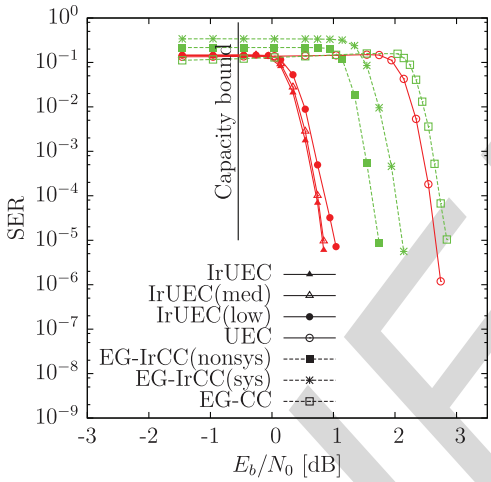


Fig. 12. SER performance for various arrangements of the proposed IrUEC-IrURC scheme of Fig. 1, the EG-IrCC-IrURC of Fig. 7, the Parallel IrUEC-IrURC scheme of Fig. 9, as well as the UEC-IrURC and the EG-IrURC schemes of [10], when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.9$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values. The complexity is unlimited for decoding each of the bits in $\mathbf{z}$.

is provided in Fig. 12 for the various schemes considered, where the source symbols obey a zeta distribution having the parameter $p_1 = 0.9$, where the complexity is potentially unlimited. It can be seen that the proposed IrUEC-IrURC scheme also outperforms all other benchmarkers in this situation, offering a 1 dB gain compared to the EG-IrCC(nonsys)-IrURC scheme, which is the best-performing one of the set of SSCC benchmarkers.

Note that the performance gain of the proposed IrUEC-IrURC scheme is obtained by elaborately designing the IrUEC EXIT function, in order to create a narrow but marginally open EXIT chart tunnel at a low E_b/N_0 value that is close to the area bound and capacity bound, as discussed in Section IV-B. Since the benchmarker schemes suffer from capacity loss which separates their tunnel, area and capacity bounds, the performance gain of the proposed IrUEC-IrURC scheme depicted in Fig. 10 and 12 may be expected in the general case, regardless of the specific source probability distribution and the parametrization of the scheme. As an additional benefit of the proposed IrUEC-IrURC scheme, a single bit error within a particular codeword can only result in splitting it into two codewords, or into merging it with the next codeword, since every unary codeword contains only a single 0. Fortunately, the decoding of the other unary codewords will be unaffected. Owing to this, a single bit error in the IrUEC-IrURC scheme can only cause a Levenshtein distance [33] of 2, hence preventing error propagation. By contrast, in the EG-based benchmarkers, a single bit error can cause error propagation, resulting in a Levenshtein distance that is bounded only by the length of the message.

VI. Conclusions

In this paper, we have proposed a novel near-capacity JSCC scheme, which we refer to as the IrUEC code. Like the regular UEC code of [10], this employs a unary code, but replaces the UEC's trellis code with a novel IrTrellis code. Unlike a conventional irregular code, the IrTrellis code operates on the basis of a single amalgamated irregular trellis, rather than a number of separate trellises. Our results demonstrated that this single amalgamated trellis offers gains of up to 0.2 dB over the use of separate trellises, without imposing any increase in transmission energy, bandwidth, latency or decoding complexity. By characterizing the free distance property of the UEC trellis, we have selected a suite of UEC codes having a wide variety of EXIT chart shapes for the component codes of our IrUEC code. We concatenated the proposed IrUEC code with an IrURC code in Fig. 1 and introduced a new double-sided EXIT chart matching algorithm. On the one hand, the component UEC codes having a wide variety of EXIT chart shapes provide a great design freedom of the IrUEC EXIT chart. On the other hand, the novel double-sided EXIT chart matching algorithm utilize this design freedom sufficiently, in order to parametrize the IrUEC-IrURC scheme for creating a narrow but marginally open EXIT chart tunnel at a low E_b/N_0 value that is close the area bound and the capacity bound. As a result, near-capacity operation is facilitated at E_b/N_0 values that are within 0.4 dB of the DCMC capacity bound, when achieving an effective throughput of $\eta = 0.508$ bit/s/Hz and employing (QPSK) for transmission over an uncorrelated narrowband Rayleigh fading channel. This corresponds to a gain of 0.8 dB compared to the best of several SSCC benchmarkers, which is achieved without any increase in transmission energy, bandwidth, transmit duration or decoding complexity.

References

[1] J. Zou, H. Xiong, C. Li, R. Zhang, and Z. He, "Lifetime and distortion optimization with joint source/channel rate adaptation and network coding-based error control in wireless video sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1182–1194, Mar. 2011.

[2] Y. Huo, C. Zhu, and L. Hanzo, "Spatio-temporal iterative source-channel decoding aided video transmission," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1597–1609, May 2013.

[3] N. S. Othman, M. El-Hajjar, O. Alamri, S. X. Ng, and L. Hanzo, "Iterative AMR-WB source and channel decoding using differential space-time spreading-assisted sphere-packing modulation," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 484–490, Jan. 2009.

[4] C. E. Shannon, "The mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.

[5] B. Ryabko and J. Rissanen, "Fast adaptive arithmetic code for large alphabet sources with asymmetrical distributions," in *Proc. IEEE Int. Symp. Inf. Theory*, 2002, p. 319.

[6] J. Ziv and A. Lempel, "Compression of individual sequences via variable rate coding," *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, Sep. 1978.

[7] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 194–203, Mar. 1975.

[8] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Std. H.264, Mar. 2005.

[9] J. L. Massey, "Joint source and channel coding," in *Proc. Commun. Syst. Random Process Theory*, Dec. 1978, pp. 279–293.

[10] R. G. Maunder, W. Zhang, T. Wang, and L. Hanzo, "A unary error correction code for the near-capacity joint source and channel coding of symbol values from an infinite set," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1977–1987, May 2013.

[11] N. L. Johnson, A. W. Kemp, and S. Kotz, *Univariate Discrete Distributions*. Hoboken, NJ, USA: Wiley, 2005.

[12] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 158–162, Feb. 1995.

[13] V. Buttigieg and P. G. Farrell, "Variable-length error-correcting codes," *IEE Proc. Commun.*, vol. 147, no. 4, pp. 211–215, Aug. 2000.

[14] 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.

[15] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 209–218, Feb. 2004.

[16] R. G. Maunder and L. Hanzo, "Genetic algorithm aided design of component codes for irregular variable length coding," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1290–1297, May 2009.

[17] 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.

[18] W. Zhang, R. G. Maunder, and L. Hanzo, "On the complexity of unary error correction codes for the near-capacity transmission of symbol values from an infinite set," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2013, pp. 2795–2800.

[19] T. Wang, W. Zhang, R. G. Maunder, and L. Hanzo, "Near-capacity joint source and channel coding of symbol values from an infinite source set using Elias gamma error correction codes," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 280–292, Jan. 2014.

[20] L. Hanzo, R. G. Maunder, J. Wang, and L.-L. Yang, *Near-Capacity Variable Length Coding*. Hoboken, NJ, USA: Wiley, 2010.

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

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

[23] 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.

[24] J. Kliewer, N. Goertz, 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.

[25] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo like codes," in *Proc. 36th Allerton Conf. Commun. Control Comput.*, Allerton House, IA, USA, Sep. 1998, pp. 201–210.

[26] A. Diallo, C. Weidmann, and M. Kieffer, "Efficient computation and optimization of the free distance of variable-length finite-state joint source-channel codes," *IEEE Trans. Commun.*, vol. 59, no. 4, pp. 1043–1052, Apr. 2011.

[27] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[28] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2002, pp. 748–753.

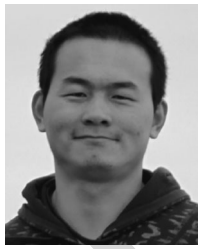
[29] L. Hanzo, T. H. Liew, B. L. Yeap, R. Y. S. Tee, and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart Aided Near-Capacity Designs for Wireless Channels*. Hoboken, NJ, USA: Wiley, 2010.

[30] 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, USA, Jul. 2006, pp. 2680–2684.

[31] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Commun. Lett.*, vol. 3, no. 11, pp. 317–319, Nov. 1999.

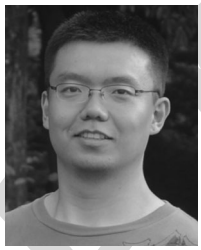
[32] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 14–22, Jan. 2013.

[33] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Reading, MA, USA: Addison-Wesley, 1983.

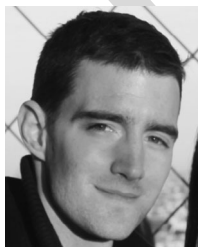


Wenbo Zhang (S'14) received the M.E. degree in information and communication engineering from the University of Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011. Currently, he is pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton, Southampton, U.K. His research interests include joint source/channel coding and variable length coding.

Matthew F. Brejza received the B.Eng. degree (first class Hons.) in electronic engineering from the University of Southampton, Southampton, U.K., in 2012, and is currently pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton. His research interests include flexible hardware implementation, channel coding and their applications in low power data communications.



Tao Wang received the B.S. degree in information engineering from the University of Science and Technology of Beijing (USTB), Beijing, China, and the M.Sc. degree in communication from University of Southampton, Southampton, U.K., in 2006 and 2008, respectively, and is currently pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton. His research interests include joint source/channel coding and distributed video coding.



Robert G. Maunder (SM'12) received the B.Eng. degree (first class Hons.) in electronic engineering and the Ph.D. degree in wireless communications, in 2003 and 2007, respectively. He has authored a number of IEEE papers in his research interest areas. His research interests include joint source/channel coding, iterative decoding, irregular coding, and modulation techniques. He received a lectureship in December 2007.

Lajos Hanzo (F'04) received the degree in electronics and the doctorate degree, in 1976 and 1983, respectively. During his 38-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, Southampton, U.K., where he holds the Chair in telecommunications. He has coauthored 20 Wiley/IEEE Press books on mobile radio communications totalling in excess of 10000 pages, authored more than 1500 research entries at the IEEE Xplore, acted both as TPC and General Chair of the IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. Currently, he is directing a 60-strong 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 Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. He is a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE Press and also a Chaired Professor at Tsinghua University, Beijing, China. His research is funded by the European Research Council's Senior Research Fellow Grant. Lajos has more than 22 000 citations. In 2009, he was was the recipient of an Honorary Doctorate by the Technical University of Budapest, while in 2015, by the University of Edinburgh.

Irregular Trellis for the Near-Capacity Unary Error Correction Coding of Symbol Values From an Infinite Set

Wenbo Zhang, *Student Member, IEEE*, Matthew F. Brejza, Tao Wang, Robert G. Maunder, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—Irregular joint source and channel coding (JSCC) scheme is proposed, which we refer to as the irregular unary error correction (IrUEC) code. This code operates on the basis of a single irregular trellis, instead of employing a set of separate regular trellises, as in previous irregular trellis-based codes. Our irregular trellis is designed with consideration of the UEC free distance, which we characterize for the first time in this paper. We conceive the serial concatenation of the proposed IrUEC code with an irregular unity rate code (IrURC) code and propose a new EXtrinsic Information Transfer (EXIT) chart matching algorithm for parametrizing these codes. This facilitates the creation of a narrow EXIT tunnel at a low E_b/N_0 value and provides near-capacity operation. Owing to this, our scheme is found to offer a low symbol error ratio (SER), which is within 0.4 dB of the discrete-input continuous-output memoryless channel (DCMC) capacity bound in a particular practical scenario, where gray-mapped quaternary phase shift keying (QPSK) modulation is employed for transmission over an uncorrelated narrowband Rayleigh-fading channel with an effective throughput of $0.508\text{bit}\,\text{s}^{-1}\,\text{Hz}^{-1}$. Furthermore, the proposed IrUEC–IrURC scheme offers a SER performance gain of 0.8 dB, compared to the best of several regular and irregular separate source and channel coding (SSCC) benchmarkers, which is achieved without any increase in transmission energy, bandwidth, transmit duration, or decoding complexity.**

***Index Terms*—Joint source–channel coding, irregular codecs, channel capacity, iterative decoding.**

I. Introduction

IN MOBILE wireless scenarios, multimedia transmission is required to be bandwidth efficient and resilient to transmission errors, motivating both source and channel coding [1]–[3]. Classic Separate Source and Channel Coding (SSCC) may be achieved by combining a near-entropy source code with a near-capacity channel code. In this scenario, it is theoretically possible to reconstruct the source information with an infinitesimally low probability of error, provided that the transmission rate does not exceed the channel's capacity [4]. However, separate source-channel coding [4] is only capable of approaching the capacity in the general case by imposing both infinite complexity and infinite latency. For example, adaptive arithmetic coding [5] and Lempel-Ziv coding [6] are capable of encoding a sequence of symbols using a near-entropy number of bits per symbol. However, these schemes require both the transmitter and receiver to accurately estimate the occurrence probability of every symbol value that the source produces. In practice, the occurrence probability of rare symbol values can only be accurately estimated, if a sufficiently large number of symbols has been observed, hence potentially imposing an excessive latency.

This motivates the design of universal codes, such as the Elias Gamma (EG) code [7], which facilitate the binary encoding of symbols selected from infinite sets, without requiring any knowledge of the corresponding occurrence probabilities at either the transmitter or receiver. The H.264 video codec [8] employs the EG code and this may be concatenated with classic channel codes, such as a Convolutional Code (CC) to provide a separate error correction capability. Nevertheless, this SSCC typically suffers from a capacity loss, owing to the residual redundancy that is typically retained during EG encoding, which results in an average number of EG-encoded bits per symbol that exceeds the entropy of the symbols.

In order to exploit the residual redundancy and hence to achieve near-capacity operation, the classic SSCC schemes may be replaced by Joint Source and Channel Coding (JSCC) arrangements [9] in many applications. As we have previously demonstrated in [10, Fig. 1], the symbols that are EG encoded in H.264 are approximately zeta probability distributed [11], resulting in most symbols having low values, but some rare symbols having values around 1000. Until recently, the decoding complexity of all previous JSCCs, such as Reversible Variable Length Codes (RVLCs) [12] and Variable Length Error Correction (VLEC) codes [13], increased rapidly with the cardinality of the symbol set, so much so that it became excessive for the H.264 symbol probability distribution and asymptotically tending to infinity, when the cardinality is infinite.

Against this background, a novel JSCC scheme referred to as a Unary Error Correction (UEC) code [10] was proposed as the first JSCC that mitigates the capacity loss and incurs

Manuscript received October 31, 2014; revised April 14, 2015 and July 25, 2015; accepted October 8, 2015. This work was supported in part by the EPSRC, Swindon UK under Grant EP/J015520/1 and Grant EP/L010550/1, in part by the TSB Swindon UK under Grant TS/L009390/1, in part by the RCUK under the India-UK Advanced Technology Centre (IU-ARC), and in part by the EU under the CONCERTO project and in part by the European Research Council's Advanced Fellow grant. The associate editor coordinating the review of this paper and approving it for publication was M. Xiao.

The authors are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: wz4g11@ecs.soton.ac.uk; mfb2g09@ecs.soton.ac.uk; tw08r@ecs.soton.ac.uk; rm@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCOMM.2015.2493149

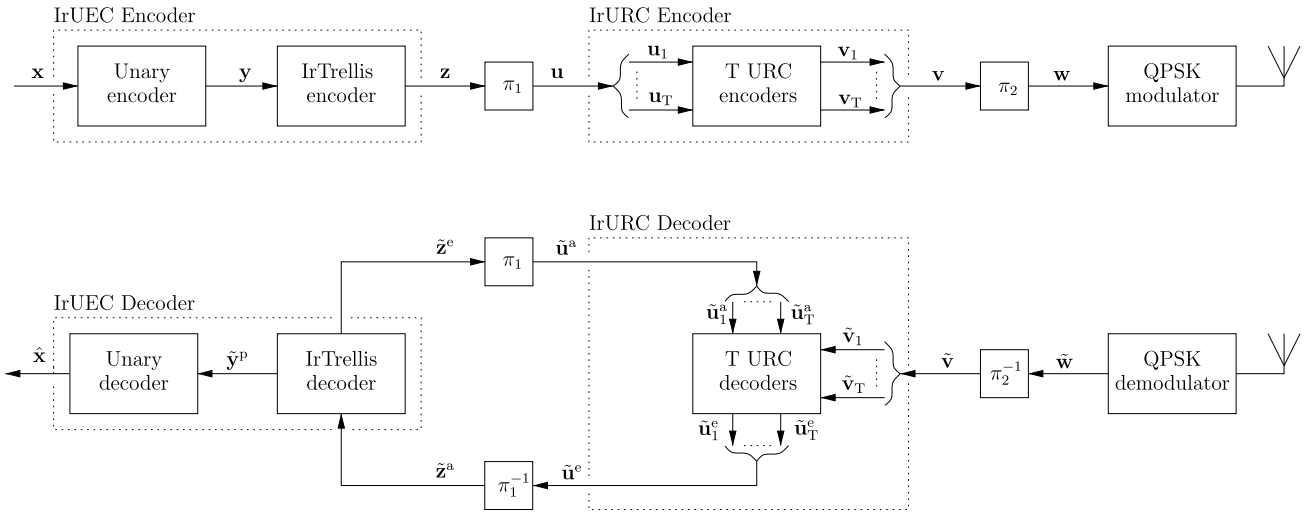


Fig. 1. Schematic of the proposed IrUEC-IrURC scheme, in which an IrUEC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

only a moderate decoding complexity, even when the cardinality of the symbol set is infinite. In a particular practical scenario, an iteratively-decoded serial concatenation of the UEC code with an Irregular Unity Rate Code (IrURC) was shown to offer a 1.3 dB gain compared to a SSCC benchmarker, without incurring an increased transmission energy, duration, bandwidth or decoding complexity. Furthermore, this was achieved within 1.6 dB of the Quaternary Phase Shift Keying (QPSK)-modulated uncorrelated narrow band Rayleigh fading Discrete-input Continuous-output Memoryless Channel (DCMC) capacity bound.

In this paper, we will further exploit the properties of UEC codes in order to facilitate reliable operation even closer to the capacity bound. More specifically, we propose an *Irregular* Unary Error Correction (IrUEC) code, which extends the regular UEC of our previous work [10]. *This IrUEC code employs different UEC parametrizations for the coding of different subsets of each message frame, in analogy with previous irregular codes, such as the IrURC [14], the Irregular Convolutional Code (IrCC) [15] and the Irregular Variable Length Code (IrVLC) [16]. However, these previous irregular codes operate on the basis of a number of separate trellises, each of which has a different but uniform structure and is used for the coding of a different subset of the message frame. By contrast, our new IrUEC code operates on the basis of a single irregular trellis having a novel design. This trellis has a non-uniform structure that applies different UEC parametrizations for different subsets of the frame on a bit-by-bit basis. This allows the irregularity of the proposed IrUEC code to be controlled on a fine-grained bit-by-bit basis, rather than on a symbol-by-symbol basis, hence facilitating nearer-to-capacity operation. More specifically, our results demonstrate that controlling the IrUEC irregularity on a bit-by-bit basis offers gains of up to 0.2 dB over the symbol-by-symbol approach, without imposing any increase in transmission energy, bandwidth, latency or decoding complexity.*

This bit-by-bit IrUEC approach is facilitated by some particular properties of UEC codes, which grant some commonality to all UEC parametrizations. By exploiting this fine-grained control of the IrUEC irregularity, the IrUEC EXtrinsic Information Transfer (EXIT) function may be shaped to create a narrow, but marginally open EXIT chart tunnel. This implies that near-capacity operation is facilitated, according to the theoretical properties of EXIT charts [17].

The rest of this paper is organized as follows. Section II describes a transmitter that serially concatenates the proposed IrUEC encoder with a IrURC encoder, while Section III describes the corresponding iterative receiver. The IrUEC encoder and decoder operate on the basis of our novel irregular trellis structure, which allows bit-level control of the irregular coding fractions. The free distance of UEC codes is quantified for the first time in Section IV, which proposes a novel low-complexity heuristic method conceived for this purpose. This is used for selecting a family of UEC trellis structures having a wide variety of EXIT function shapes. The resultant UEC trellis family maximises the design freedom for the IrUEC EXIT function and therefore has a general applicability for IrUEC codes used in diverse applications. Furthermore, for any particular application of an IrUEC code, we propose a double-sided EXIT chart matching algorithm for selecting the specific fraction of the frame that should be encoded using each IrUEC and IrURC trellis structure. This allows the EXIT functions of IrUEC and IrURC codes to be accurately shaped for closely matching each other, hence creating a narrow but marginally open EXIT chart tunnel. In Section V, the proposed IrUEC-IrURC scheme is compared to an irregular JSCC benchmarker, which is referred to as the EG-IrCC-IrURC scheme. The first version of this benchmarker employs the recursive systematic CCs that were originally recommended as IrCC component codes in [15]. However, we demonstrate that the systematic nature of these IrCC component codes results in a capacity loss. This motivates the employment of the second version of our EG-IrCC-IrURC benchmarker, which employs the recursive non-systematic CCs of [10] as the IrCC component codes. The simulation results of Section V show that in a particular practical scenario, the proposed IrUEC-IrURC scheme provides a

0.8 dB gain over the best SSCC benchmarker, while operating within 0.4 dB of the capacity bound. This is achieved without any increase in transmission energy, bandwidth, latency or decoding complexity. Finally, Section VI concludes the paper.

II. IRUEC-IRURC ENCODER

In this section, we introduce the transmitter of the proposed IrUEC-IrURC scheme of Fig. 1. The IrURC encoder employs T number of component Unity Rate Code (URC) encoders $\{\text{URC}^t\}_{t=1}^{T}$, each having a distinct independent trellis structure. By contrast, the IrUEC employs a unary encoder and a novel *Irregular* Trellis (IrTrellis) encoder with a single irregular trellis. However, in analogy with the IrURC code, we note that this irregular trellis comprises a merging of S component UEC trellis structures $\{\text{UEC}^s\}_{s=1}^{S}$, where UEC^s is the s-th component UEC trellis structure that is defined by the corresponding codeword set $\mathbb{C}_s$, as illustrated in [10, Fig. 3(a)]. In Section II-A and Section II-B, the two components of the IrUEC encoder in Fig. 1, namely the unary encoder and the novel IrTrellis encoder are detailed. The IrURC encoder and the modulator are introduced in Section II-C.

A. Unary Encoder

The IrUEC encoder is designed for conveying a vector $\mathbf{x} = [x_i]_{i=1}^{a}$ comprising a number of symbols, as shown in Fig. 1. The value of each symbol $x_i \in \mathbb{N}_1$ may be modeled by an Independent and Identically Distributed (IID) Random Variable (RV) X_i, which adopts the value x with a probability of $\Pr(X_i = x) = P(x)$, where $\mathbb{N}_1 = \{1, 2, 3, \ldots, \infty\}$ is the infinite-cardinality set comprising all positive integers. Throughout this paper we assume that the symbol values obey a zeta probability distribution [11], since this models the symbols produced by multimedia encoders, as described in Section I. The zeta probability distribution is defined as

$$P(x) = \frac{x^{-s}}{\zeta(s)}, \tag{1}$$

where $\zeta(s) = \sum_{x\in\mathbb{N}_1} x^{-s}$ is the Riemann zeta function, $s > 1$ parametrizes the zeta distribution and $p_1 = \Pr(X_i = 1) = 1/\zeta(s)$ is the probability of occurrence for the most frequently occurring symbol value, namely $x = 1$. Without loss of generality, Table I exemplifies the first ten symbol probabilities $P(x_i)$ for a zeta distribution having the parameter $p_1 = 0.797$, which corresponds to $s = 2.77$ and was found in [10] to allow a fair comparison between unary- and EG-based schemes. Note that other p_1 values of 0.694, 0.8 and 0.9 have been investigated in [18], [19]. In the situation where the symbols obey the zeta probability distribution of (1), the symbol entropy is given by

$$H_X = \sum_{x\in\mathbb{N}_1} H[P(x)] = \frac{\ln(\zeta(s))}{\ln(2)} - \frac{s\zeta'(s)}{\ln(2)\zeta(s)}, \tag{2}$$

where $H[p] = p\log_2(1/p)$ and $\zeta'(s) = -\sum_{x\in\mathbb{N}_1} \ln(x)x^{-s}$ is the derivative of the Riemann zeta function.

As shown in Fig. 1, the IrUEC encoder represents the source vector $\mathbf{x}$ using a unary encoder. More specifically, each symbol x_i in the vector $\mathbf{x}$ is represented by a corresponding codeword $\mathbf{y}_i$ that comprises x_i bits, namely $(x_i - 1)$ binary ones followed by a zero, as exemplified in Table I. When the symbols adopt the zeta distribution of (1), the average unary codeword length l is only finite for $s > 2$ and hence for $p_1 > 0.608$ [10], in which case we have

$$l = \sum_{x\in\mathbb{N}_1} P(x)\cdot x = \frac{\zeta(s-1)}{\zeta(s)}. \tag{3}$$

Note that for $p_1 \leq 0.608$, our Elias Gamma Error Correction (EGEC) code of [19] may be employed in order to achieve a finite average codeword length, albeit at the cost of an increased complexity. In our future work, we will consider a novel Irregular EGEC code, which has a finite codeword length for $p_1 \leq 0.608$. Without loss of generality, in the example scenario of $p_1 = 0.797$, an average codeword length of $l = 1.54$ results. The output of the unary encoder is generated by concatenating the selected codewords $\{\mathbf{y}_i\}_{i=1}^{a}$, in order to form the b-bit vector $\mathbf{y} = [y_j]_{j=1}^{b}$. For example, the source vector $\mathbf{x} = [4, 1, 2]$ of $a = 3$ symbols yields the $b = 7$-bit vector $\mathbf{y} = [1110010]$. Note that the average length of the bit vector $\mathbf{y}$ is given by $(a \cdot l)$.

TABLE I
THE FIRST TEN SYMBOL PROBABILITIES FOR A ZETA DISTRIBUTION HAVING THE PARAMETER $p_1 = 0.797$, AS WELL AS THE CORRESPONDING UNARY AND EG CODEWORDS

x_i	$P(x_i)$	$\mathbf{y}_i$	
		Unary	EG
1	0.797	0	1
2	0.117	10	010
3	0.038	110	011
4	0.017	1110	00100
5	0.009	11110	00101
6	0.006	111110	00110
7	0.004	1111110	00111
8	0.003	11111110	0001000
9	0.002	111111110	0001001
10	0.001	1111111110	0001010

B. IrTrellis Encoder

Following unary encoding, the IrTrellis encoder of Fig. 1 employs a single new *irregular* trellis to encode the bit vector $\mathbf{y}$, rather than using a selection of separate trellis structures, as is necessary for the IrCC [15], IrVLC [16] and IrURC [14] coding schemes. Our novel irregular trellis structure is facilitated by the properties of the generalised trellis structure of [10, Fig. 3(a)], which was the basis of our previous work on regular UEC codes. This trellis structure is parametrized by an even number of states r and by the UEC codeword set $\mathbb{C}$, which comprises $r/2$ binary codewords of a particular length n. Each bit y_j of the unary-encoded bit sequence $\mathbf{y} = [y_j]_{j=1}^{b}$ corresponds to a transition in the UEC trellis from the previous state $m_{j-1} \in \{1, 2, \ldots, r\}$ to the next state $m_j \in \{1, 2, \ldots, r\}$. Each next state m_j is selected from two legitimate alternatives, depending both on the previous state m_{j-1} and on the bit value y_j, according to [18, (3)]. More specifically, regardless of how the UEC trellis is parametrized, a unary-coded bit of $y_j = 1$ causes a transition towards state $m_j = r - 1$ or r of the generalised UEC trellis of [10, Fig. 3(a)], while the $y_j = 0$-valued bit

at the end of each unary codeword causes a transition to state $m_j = 1$ or 2, depending on whether the current symbol x_i has an odd or even index i.

This common feature of all UEC trellises maintains synchronisation with the unary codewords and allows the residual redundancy that remains following unary encoding to be explicated for error correction. Furthermore, this common treatment of the unary-encoded bits in $\mathbf{y}$ between all UEC trellises allows them to merge in order to form our novel irregular trellis. More specifically, our novel irregular trellis can be seen as concatenation of a number of individual UEC trellis structures with different numbers of states r and different codeword sets $\mathbb{C}$. By contrast, CCs, Variable Length Codes (VLCs) and URC codes having different parametrizations do not generally exhibit the required similarity in their trellises. More specifically, the final state of a particular component encoder has no specific relationship with the initial state of the subsequent component encoder, hence preventing their amalgamation into IrCC, IrVLC and IrURC trellises, respectively.

The IrTrellis encoder of Fig. 1 encodes the b-bit unary-encoded bit sequence $\mathbf{y} = [y_j]_{j=1}^{b}$ using an irregular trellis that is obtained by concatenating b number of regular UEC trellis structures. The proposed IrTrellis can be constructed using diverse combinations of component regular UEC trellises, having any parametrization. However, the component regular trellises may be strategically selected in order to carefully shape the EXIT function of the IrUEC code, for the sake of producing a narrow EXIT chart tunnel and for facilitating near-capacity operation, as it will be detailed in Section IV. Without loss of generality, Fig. 1 provides an example of the irregular trellis for the example scenario where we have $b = 7$. Each bit y_j in the vector $\mathbf{y}$ is encoded using the corresponding one of these b trellis structures, which is parametrized by an even number of states r_j and the codeword set $\mathbb{C}_j = \{\mathbf{c}_1^j, \mathbf{c}_2^j, \ldots, \mathbf{c}_{r_j/2-1}^j, \mathbf{c}_{r_j/2}^j\}$, which comprises $r_j/2$ binary codewords of a particular length n_j. Note that successive trellis structures can have different numbers of states, subject to the constraint $r_j \leq r_{j-1} + 2$, as it will be demonstrated in the following discussions. Note that this constraint does not restrict the generality of the IrUEC trellis, since the IrUEC EXIT function shape is independent of the ordering of the component trellis structures.

As in the regular UEC trellis of [10], the encoding process always emerges from the state $m_0 = 1$. The unary-encoded bits of $\mathbf{y}$ are considered in order of increasing index j and each bit y_j causes the novel IrTrellis to traverse from the previous state $m_{j-1} \in \{1, 2, \ldots, r_{j-1}\}$ to the next state $m_j \in \{1, 2, \ldots, r_j\}$, which is selected from two legitimate alternatives. More specifically,

$$m_j = \begin{cases} 1 + \text{odd}\big(m_{j-1}\big) & \text{if } y_j = 0 \\ \min\big[m_{j-1} + 2, r_j - \text{odd}\left(m_{j-1}\right)\big] & \text{if } y_j = 1 \end{cases}, \quad (4)$$

where the function odd$(\cdot)$ yields 1 if the operand is odd or 0 if it is even. Note that the next state m_j in the irregular trellis is confined by the number of states r_j in the corresponding trellis structure, rather than by a constant number of states r, as in the regular UEC trellis of [10]. In this way, the bit sequence $\mathbf{y}$ identifies a path through the single irregular trellis, which may be represented by a vector $\mathbf{m} = [m_j]_{j=0}^{b}$ comprising $b + 1$ state values. As in the regular UEC trellis of [10], the transitions of the proposed irregular trellis are synchronous with the unary codewords of Table I. More specifically, just as each symbol x_i in the vector $\mathbf{x}$ corresponds to an x_i-bit codeword $\mathbf{y}_i$ in the vector $\mathbf{y}$, the symbol x_i also corresponds to a section $\mathbf{m}_i$ of the trellis path $\mathbf{m}$ comprising x_i transitions between $(x_i + 1)$ states. Owing to this, the path $\mathbf{m}$ is guaranteed to terminate in the state $m_b = 1$, when the symbol vector $\mathbf{x}$ has an even length a, while $m_b = 2$ is guaranteed when a is odd [10]. Note that the example unary-encoded bit sequence $\mathbf{y} = [1110010]$ corresponds to the path $\mathbf{m} = [1, 3, 5, 3, 2, 1, 1, 2]$ through the irregular UEC trellis of Fig. 2.

The path $\mathbf{m}$ may be modeled as a particular realization of a vector $\mathbf{M} = [M_j]_{j=0}^{b}$ comprising $(b + 1)$ RVs. Note that the probability $\Pr(M_j = m_j, M_{j-1} = m_{j-1}) = P(m_j, m_{j-1})$ of the transition from the previous state m_{j-1} to the next state m_j can be derived by observing the value of each symbol in the vector $\mathbf{x}$ and simultaneously its corresponding index. The state transition $\mathbf{M} = \{M_j\}_{j=0}^{b}$ follows the same rule shown in (4), and all the transitions can be categorised into four types, as illustrated in [10, (8)]. Owing to this, the probability of a transition $P(m_j, m_{j-1})$ in the irregular trellis is associated with the transition probabilities $\Pr(M_j = m, M_{j-1} = m') = P(m, m')$ in (5), shown at the bottom of the page. Note that these

$$P(m_j, m_{j-1}) = \begin{cases} \dfrac{1}{2l}\left[1 - \sum_{x=1}^{\left\lceil \frac{m_{j-1}}{2} \right\rceil} P(x)\right] & \text{if } m_{j-1} \in \{1, 2, 3, \ldots, r_{j-1} - 2\},\ m_j = m_{j-1} + 2 \\ \dfrac{1}{2l} P(x)\Big|_{x=\left\lceil \frac{m_{j-1}}{2} \right\rceil} & \text{if } m_{j-1} \in \{1, 2, 3, \ldots, r_{j-1} - 2\},\ m_j = 1 + \text{odd}(m_{j-1}) \\ \dfrac{1}{2l}\left[1 - \sum_{x=1}^{\frac{r_{j-1}}{2}-1} P(x)\right] & \text{if } m_{j-1} \in \{r_{j-1} - 1, r_{j-1}\},\ m_j = 1 + \text{odd}(m_{j-1}) \\ \dfrac{1}{2l}\left[l - \frac{r_{j-1}}{2} - \sum_{x=1}^{\frac{r_{j-1}}{2}-1} P(x)\left(x - \frac{r_{j-1}}{2}\right)\right] & \text{if } m_{j-1} \in \{r_{j-1} - 1, r_{j-1}\},\ m_j \in \{r_j - 1, r_j\} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

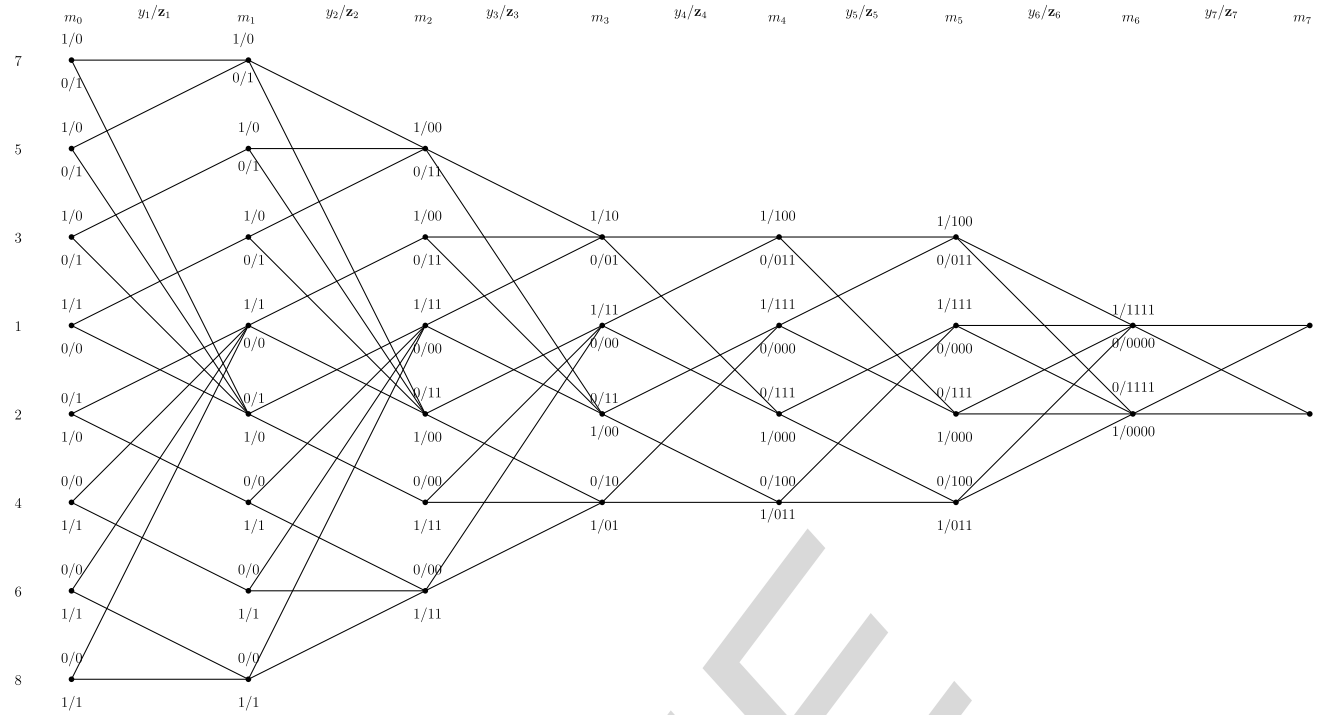


Fig. 2. An example of the proposed irregular UEC trellis, which is obtained by amalgamating seven different UEC trellises. Here, the component UEC codebooks $\mathbb{C}_1 = \{0, 1, 1, 1\}$, $\mathbb{C}_2 = \{0, 1, 1, 1\}$, $\mathbb{C}_3 = \{000, 000, 000\}$, $\mathbb{C}_4 = \{00, 01\}$, $\mathbb{C}_5 = \{000, 011\}$, $\mathbb{C}_6 = \{000, 011\}$ and $\mathbb{C}_7 = \{0000\}$ are employed.

transition probabilities are generalized, allow their application to any IrUEC trellis and to any source probability distribution $P(x)$.

Similar to the regular UEC trellis encoder, the proposed IrTrellis encoder represents each bit y_j in the vector $\mathbf{y}$ by a codeword $\mathbf{z}_j$ comprising n_j bits. This is selected from the corresponding set of $r_j/2$ codewords $\mathbb{C}_j = \{\mathbf{c}^j_1, \mathbf{c}^j_2, \ldots, \mathbf{c}^j_{r_j/2-1}, \mathbf{c}^j_{r_j/2}\}$ or from the complementary set $\overline{\mathbb{C}}_j = \{\overline{\mathbf{c}^j_1}, \overline{\mathbf{c}^j_2}, \ldots, \overline{\mathbf{c}^j_{r_j/2-1}}, \overline{\mathbf{c}^j_{r_j/2}}\}$, which is achieved according to

$$\mathbf{z}_j = \begin{cases} \overline{\mathbf{c}^j_{\lceil m_{j-1}/2 \rceil}} & \text{if } y_j = \text{odd}(m_{j-1}) \\ \mathbf{c}^j_{\lceil m_{j-1}/2 \rceil} & \text{if } y_j \neq \text{odd}(m_{j-1}) \end{cases}. \tag{6}$$

Finally, the selected codewords are concatenated to obtain the bit vector $\mathbf{z} = [z_k]_{k=1}^{b\bar{n}}$ of Fig. 1, where $\bar{n} = \frac{1}{b}\sum_{j=1}^{b} n_j$ is the average codeword length. For example, the path $\mathbf{m} = [1, 3, 5, 3, 2, 1, 1, 2]$ through the irregular UEC trellis of Fig. 2 yields the encoded bit sequence $\mathbf{z} = [1000011111110000]$, which comprises $b\bar{n} = 16$ bits, where we have $\bar{n} = \frac{16}{7}$.

Note that the bit vector $\mathbf{z}$ may be modeled as a specific realization of a vector $\mathbf{Z} = [Z_k]_{k=1}^{b\bar{n}}$ comprising $b\bar{n}$ binary RVs. Observe in Fig. 2 that each of the b component trellis structures in the irregular UEC trellis of the IrTrellis encoder is designed to obey symmetry and to rely on complementary codewords. Hence, bits of the encoded bit vector $\mathbf{Z}$ have equiprobable values, where $\Pr(Z_k = 0) = \Pr(Z_k = 1) = 0.5$, and the bit entropy obeys $H_{Z_k} = H[\Pr(Z_k = 0)] + H[\Pr(Z_k = 1)] = 1$. Owing to this, in contrast to some of the benchmarkers to be considered in Section V, the proposed IrUEC scheme of Fig. 1 does not suffer from additional capacity loss.

We assume that each of the b trellis structures in the proposed irregular UEC trellis is selected from a set of S component UEC trellis structures $\{\text{UEC}^s\}_{s=1}^{S}$, corresponding to a set of S component codebooks $\{\mathbb{C}_s\}_{s=1}^{S}$. More specifically, we assume that each codebook $\mathbb{C}_s$ is employed for generating a particular fraction α_s of the bits in $\mathbf{z}$, where we have $\sum_{s=1}^{S} \alpha_s = 1$. Here, the number of bits generated using the codebook $\mathbb{C}_s$ is given by $b\bar{n} \cdot \alpha_s$. We will in Section IV show that the fractions $\alpha = \{\alpha_s\}_{s=1}^{S}$ may be designed in order to appropriately shape the IrUEC EXIT function. Moreover, the IrUEC coding rate is given by $R_{\text{IrUEC}} = \sum_{s=1}^{S} \alpha_s \cdot R_{\text{UEC}^s}$, where the corresponding coding rate R_{UEC^s} of the regular UEC^s code depends on the codebook $\mathbb{C}_s$ and is given by [10, Eq. (11)].

C. IrURC Encoder and Modulator

As shown in Fig. 1, the IrUEC-encoded bit sequence $\mathbf{z}$ is interleaved in the block π_1 in order to obtain the bit vector $\mathbf{v}$, which is encoded by an IrURC encoder [14], [20] comprising T component URC codes $\{\text{URC}^t\}_{t=1}^{T}$. Unlike our IrUEC code, each component URC code URC^t of the IrURC code employs a separate trellis structure. This is necessary, since the final state of each component URC code has no relation to the initial state of the subsequent component URC code, as described in Section II-B. Therefore, the interleaved IrURC-encoded bit vector $\mathbf{u}$ is decomposed into T sub-vectors $\{\mathbf{u}_t\}_{t=1}^{T}$, each having a length given by $b\bar{n} \cdot \beta_t$, where β_t represents the specific fraction of the bits in $\mathbf{v}$ that are encoded by the component URC^t code, which obeys $\sum_{t=1}^{T} \beta_t = 1$. In Section IV, we also show that the fractions $\beta = \{\beta_t\}_{t=1}^{T}$ may be designed in order to shape the IrURC EXIT function.

In common with each of its T number of component URC codes, the IrURC code has a coding rate of $R_{\text{IrURC}} = 1$, regardless of the particular irregular code design. Owing to this, each of the T number of binary sub-vectors $\{\mathbf{v}_t\}_{t=1}^{T}$ that result from IrURC encoding has the same length as the corresponding sub-vector $\mathbf{u}_t$. The set of these sub-vectors $\{\mathbf{v}_t\}_{t=1}^{T}$ are concatenated to obtain the bit-vector $\mathbf{v}$, which comprises $b\bar{n}$ bits.

Finally, the IrURC-encoded bit vector $\mathbf{v}$ is interleaved by π_2 in order to obtain the bit vector $\mathbf{w}$, which is modulated onto the uncorrelated non-dispersive Rayleigh fading channel using Gray-mapped QPSK. The overall effective throughput of the proposed scheme is given by $\eta = R_{\text{IrUEC}} \cdot R_{\text{IrURC}} \cdot \log_2(M)$, where we have $M = 4$ for QPSK.

III. IrUEC-IrURC Decoder

In this section, we introduce the receiver of the proposed IrUEC-IrURC scheme shown in Fig. 1. In analogy with the IrURC encoder, the IrURC decoder employs T number of component URC decoders $\{\text{URC}^t\}_{t=1}^{T}$, each having a distinct independent trellis structure. By contrast, the IrUEC employs a unary decoder and a novel IrTrellis decoder relying on a single irregular trellis. In Section III-A, the demodulator and the iterative operation of the IrURC and IrUEC decoders will be discussed, while in Sections III-B and III-C we will detail the internal operation of two components of the IrUEC decoder, namely of the IrTrellis decoder and of the unary decoder, respectively.

A. Demodulator and Iterative Decoding

As shown in Fig. 1, QPSK demodulation is employed by the receiver in order to obtain the vector $\tilde{\mathbf{w}}$ of Logarithmic Likelihood Ratios (LLRs), which pertain to the bits in the vector $\mathbf{w}$. This vector is deinterleaved by π_2^{-1} for the sake of obtaining the LLR vector $\tilde{\mathbf{v}}$, which is decomposed into the T sub-vectors $\{\tilde{\mathbf{v}}_t\}_{t=1}^{T}$ that have the same lengths as the corresponding sub-vectors of $\{\mathbf{v}_t\}_{t=1}^{T}$. Here, we assume that a small amount of side information is used for reliably conveying the lengths of all vectors in the IrUEC-IrURC transmitter to the receiver. The sub-vectors $\{\tilde{\mathbf{v}}_t\}_{t=1}^{T}$ are then input to the corresponding component URC decoders $\{\text{URC}^t\}_{t=1}^{T}$ of the IrURC decoder.

Following this, iterative exchanges of the vectors of extrinsic LLRs [21] commences between the Soft-Input Soft-Output (SISO) IrUEC and IrURC decoders. In Fig. 1, the notation $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{z}}$ represent vectors of LLRs pertaining to the bit vectors $\mathbf{u}$ and $\mathbf{z}$, which are related to the inner IrURC decoder and the outer IrUEC decoder, respectively. Additionally, a subscript of this notation denotes the dedicated role of the LLRs, with a, e and p indicating *a priori*, *extrinsic* and *a posteriori* LLRs, respectively.

At the beginning of iterative decoding, the *a priori* LLR vector $\tilde{\mathbf{u}}^{\text{a}}$ is initialised with a vector of zeros, having the same length as the corresponding bit vector $\mathbf{u}$. As shown in the IrURC decoder of Fig. 1, the vector $\tilde{\mathbf{u}}^{\text{a}}$ is decomposed into the T sub-vectors $\{\tilde{\mathbf{u}}_t^{\text{a}}\}_{t=1}^{T}$, which have the same lengths as the corresponding sub-vectors of $\{\mathbf{u}_t\}_{t=1}^{T}$. Together with $\{\tilde{\mathbf{v}}_t^{\text{a}}\}_{t=1}^{T}$, the sub-vectors $\{\tilde{\mathbf{u}}_t^{\text{a}}\}_{t=1}^{T}$ are fed to the corresponding URC decoder URC^t, which then outputs the resulting extrinsic LLR vectors $\{\tilde{\mathbf{u}}_t^{\text{e}}\}_{t=1}^{T}$ by employing the logarithmic Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [22]. These vectors are combined for forming the extrinsic LLR vector $\tilde{\mathbf{u}}^{\text{e}}$ that pertains to the vector $\mathbf{u}$, which is sequentially deinterleaved by the block π_1^{-1} in order to obtain the *a priori* LLR vector $\tilde{\mathbf{z}}^{\text{a}}$ that pertains to the bit vector $\mathbf{z}$. Similarly, the IrTrellis decoder is provided with the *a priori* LLR vector $\tilde{\mathbf{z}}^{\text{a}}$ and generates the vector of extrinsic LLRs $\tilde{\mathbf{z}}^{\text{e}}$, which are interleaved in the block π_1 to obtain the *a priori* LLR vector $\tilde{\mathbf{u}}^{\text{a}}$ that is provided for the next iteration of the IrURC decoder.

B. IrTrellis Decoder

As discussed in Section II, our IrUEC code employs a novel bit-based irregular trellis, while the IrURC code employs a selection of independent trellises. The novel IrTrellis decoder within the IrUEC decoder applies the BCJR algorithm to the irregular trellis. The synchronization between the novel irregular trellis and the unary codewords is exploited during the BCJR algorithm's γ_t calculation of [22, (9)]. This employs the conditional transition probability $\Pr(M_j = m_j | M_{j-1} = m_{j-1})$, where we have

$$P(m_j|m_{j-1}) = \frac{P(m_j, m_{j-1})}{\sum_{\check{m}=1}^{r_j} P(\check{m}, m_{j-1})} \tag{7}$$

and $P(m_j, m_{j-1})$ is given in (5).

Note that the IrUEC decoder will have an EXIT function [23] that reaches the (1, 1) point of perfect convergence to an infinitesimally low Symbol Error Ratio (SER), provided that all component codebooks in the set $\{\mathbb{C}_s\}_{s=1}^{S}$ have a free distance of at least 2 [24], as characterised in Section IV. Since the combination of the IrURC decoder and demodulator will also have an EXIT curve that reaches the (1, 1) point in the top right corner of the EXIT chart, iterative decoding convergence towards the Maximum Likelihood (ML) performance is facilitated [25]. At this point, the IrTrellis decoder may invoke the BCJR algorithm for generating the vector of *a posteriori* LLRs $\tilde{\mathbf{y}}^{\text{p}}$ that pertain to the corresponding bits in the vector $\mathbf{y}$.

C. Unary Decoder

As described in [10], the unary decoder of Fig. 1 sorts the values in the LLR vector $\tilde{\mathbf{y}}^{\text{p}}$ in order to identify the a number of bits in the vector $\mathbf{y}$ that are most likely to have values of zero. A hard decision vector $\hat{\mathbf{y}}$ is then obtained by setting the value of these bits to zero and the value of all other bits to one. Finally, the bit vector $\hat{\mathbf{y}}$ can be unary decoded in order to obtain the symbol vector $\hat{\mathbf{x}}$ of Fig. 1, which is guaranteed to comprise a number of symbols.

IV. Algorithm for the Parametrization of the IrUEC-IrURC Scheme

The performance of the IrUEC-IrURC scheme depends on how well it is parametrized. A good parametrization is one that

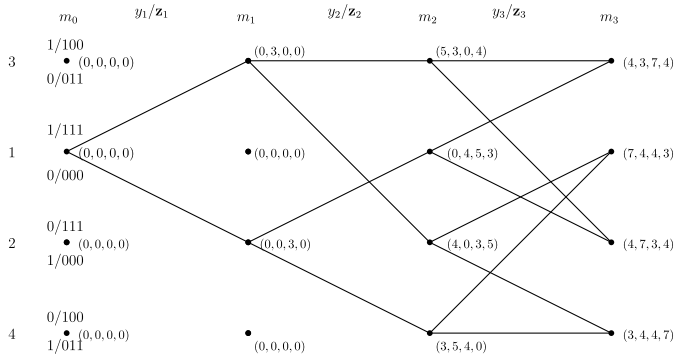


Fig. 3. The legitimate paths through the first three stages in UEC trellis having the codewords $\mathbb{C} = \{000, 011\}$.

results in a narrow but still open EXIT chart tunnel, although achieving this requires a high degree of design freedom, when shaping the IrUEC and IrURC EXIT functions. Therefore, we begin in Section IV-A by characterising the free distance property of the UEC codes and selecting a set of UEC component codes having a wide variety of different inverted EXIT function shapes. This maximises the degree of freedom that is afforded, when matching the IrUEC EXIT function to that of the IrURC code. In Section IV-B, we propose a novel extension to the double-sided EXIT chart matching algorithm of [14], which we employ for jointly matching the EXIT functions of the IrUEC and the IrURC codes. However, in contrast to the algorithm of [14], which does not allow a particular coding rate to be targeted for the IrUEC-IrURC scheme, our algorithm designs both the fractions α and β to achieve a particular target coding rate. In Section V, this will be exploited to facilitate a fair comparison with benchmarkers having particular coding rates.

A. Design of UEC Component Codes

Since an r-state n-bit UEC code is parametrized by a codebook set $\mathbb{C}$ comprising $r/2$ number of codewords each having n bits, there are a total of $2^{n \cdot r/2}$ number of candidates for $\mathbb{C}$. It is neither possible nor necessary to employ all these $2^{n \cdot r/2}$ codebooks as the component codes in our IrUEC code, because some of the codebooks will have identical or similar inverted EXIT function shapes, offering no additional degree of freedom, when performing EXIT chart matching. Therefore, it is desirable to eliminate these candidate codebooks.

The generalised UEC trellis structure associated with the codebook $\mathbb{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r/2-1}, \mathbf{c}_{r/2}\}$ is depicted in [10, Fig. 3(a)]. Note that the upper half and the lower half of the trellis is symmetrical in terms of the output codewords z_j generated in response to a given input bit value y_j, as shown in (6). More specifically, for the states in the upper half of the trellis, the output codewords z_j are selected from the codebook $\mathbb{C}$ when $y_j = 0$, while the codewords from its complementary codebook $\overline{\mathbb{C}} = \{\overline{\mathbf{c}_1}, \overline{\mathbf{c}_2}, \ldots, \overline{\mathbf{c}_{r/2-1}}, \overline{\mathbf{c}_{r/2}}\}$ are selected when $y_j = 1$. For the states in the lower half of the trellis, the output codewords z_j are selected from the codebook $\mathbb{C}$ when $y_j = 1$ and from the complementary codebook $\overline{\mathbb{C}}$ when $y_j = 0$. Intuitively, if any particular subset of the n bits at the same positions within each codeword of $\mathbb{C}$ are inverted, this would not change the distance properties of the output bit vector $\mathbf{z}$, hence resulting in an identical inverted EXIT function. For example, inverting the first bit of each codeword in the codebook $\mathbb{C}_0 = \{00, 01\}$ will give a new codebook $\mathbb{C}_1 = \{10, 11\}$ having an identical EXIT function. Likewise, inverting both bits of the codewords in $\mathbb{C}_0$ will give $\mathbb{C}_2 = \{11, 10\}$, which also has an identical EXIT function. Similarly, swapping any pair of the n bits at the same positions between each pair of codewords will not affect the distance properties or the shape of the inverted EXIT function either. For example, swapping the two bits in the codebook $\mathbb{C}_0$ results in a new codebook $\mathbb{C}_3 = \{00, 10\}$, having an identical inverted UEC EXIT function shape. Therefore, each of these four codebooks, $\mathbb{C}_0, \mathbb{C}_1, \mathbb{C}_2$ and $\mathbb{C}_3$, as well as their conversions created by bit-inversion and swapping, have identical inverted EXIT functions. Consequently, all but one of these codebooks can be eliminated as candidates for the sake of reducing the complexity of EXIT chart matching.

The number of candidate UEC codebooks may be further reduced by characterising their free distance properties. Since no analytic method has been developed for calculating the free distance d_f of a UEC code, we propose a heuristic method for obtaining an approximate measure of d_f. The free distance represents the minimum distance between any pair of encoded bit vectors produced by different paths through the trellis. The total number of possible pairings of paths emerging from a particular state in a UEC trellis of length b is given by $2^{b-1}(2^b - 1)$, which grows exponentially. However, considering the symmetry of a regular UEC trellis, it is possible to use a step-by-step directed search for determining the free distance, rather than using a brute force exhaustive search. Note that in the regular UEC trellis as generalised in [10, Fig. 3(a)], a bit vector $\mathbf{y} = [y_j]_{j=1}^{b}$ identifies a unique path $\mathbf{m} = [m_j]_{j=0}^{b}$ that emerges from state 1 and terminates at either state 1 or 2, hence accordingly identifying a corresponding output bit sequence $\mathbf{z} = [z_k]_{k=1}^{b\bar{n}}$. By exploiting this observation, the free distance d_f can be obtained by computing the Hamming Distance(HD) between each pair of paths and then selecting the pair having the minimum HD, whenever two paths merge at a particular state in the trellis.

When the bit sequence length considered satisfies $b > r/2$, the paths form complete trellis stages, as exemplified in Fig. 3. Therefore, in order to reduce the search complexity, we consider all permutations of the b-bit unary-encoded vector $\mathbf{y}$ bit-by-bit, considering all paths that emerge from state $m_0 = 1$ and terminate at each particular state $m_b = 1, 2, \ldots, r$, on a step-by-step basis. For a pair of states $m_j, m'_j \in \{1, 2, 3, \ldots, r\}$, we define $d^j_{m_j, m'_j}$ as the minimum HD between the set of all paths that terminate at state m_j and the set that ends at state m'_j, given the input bit sequence $[y_1, y_2, \ldots, y_j]$, where $j \in \{0, 1, \ldots, b\}$. Each state m_j is labelled as $(d^j_{m_j,1}, d^j_{m_j,2}, d^j_{m_j,3}, \ldots, d^j_{m_j,r})$, where we have $d^j_{m_j,m'_j} = d^j_{m'_j,m_j}$. For each state $m_0 \in \{1, 2, 3, \ldots, r\}$, the minimum HDs are initialized to 0s. Therefore, the distance $d^j_{m_j,m'_j}$ can be calculated by

$$d^j_{m_j,m'_j} = \min_{m_{j-1},m'_{j-1}} \left[d^{j-1}_{m_{j-1},m'_{j-1}} + h(\mathbf{z}_{m_{j-1},m_j}, \mathbf{z}_{m'_{j-1},m'_j}) \right]. \quad (8)$$

Here, $\mathbf{z}_{m_{j-1},m_j}$ is the codeword in the set $\mathbb{C}$ or in the complementary set $\overline{\mathbb{C}}$ that is generated by the transition from state m_{j-1} to state m_j, while the function $h(\cdot,\cdot)$ denotes the HD between the two operands. Owing to this, our method conceived for determining the free distance of a UEC code has a complexity order of $O[b \cdot r(r-1)]$, where r is the number of states in the trellis and b is the length of the bit vector $\mathbf{y}$ considered. Let $\mathbb{Y}_{b_1}$ be the bit sequence set associated with the set of all paths $\mathbb{M}_{b_1}$ having a length of b_1, while $\mathbb{Y}_{b_2}$ is the bit sequence set associated with the path set $\mathbb{M}_{b_2}$ having a length of b_2. Therefore, all sequences in $\mathbb{Y}_{b_1}$ are prefix of sequences in $\mathbb{Y}_{b_2}$, when we have $b_1 < b_2$. For example, when $b_1 = 2$ and $b_2 = 3$, the bit sequence $\mathbf{y}_2 = \{111011\}$ is a prefix of the bit sequence $\mathbf{y}_3 = \{111011111\}$, where $\mathbf{y}_2$ is associated with the path vector $\mathbf{m}_2 = \{1, 3, 2\}$ and $\mathbf{y}_2$ is associated with the path vector $\mathbf{m}_2 = \{1, 3, 2, 1\}$, respectively. Note that according to [26, Lemma 1], the minimum HD $d_f(\mathbb{Y}_{b_1})$ among all bit sequences in $\mathbb{Y}_{b_1}$ is an upper bound on the minimum HD $d_f(\mathbb{Y}_{b_2})$ of $\mathbb{Y}_{b_2}$, when we have $b_1 < b_2$. Owing to this, the approximate free distance d_f calculated using our method converges to the true free distance, as the lengths of the paths considered are extended towards infinity. In our experiments, we considered bit vector lengths of up to $b = 10r$. In all cases, we found that the free distance has converged before that point, regardless of how the UEC code is parametrised, owing to the common features of all UEC codes described in Section II-B."

For example, Fig. 3 shows all of the legitimate paths through an $r = 4$-state trellis employing the codebook $\mathbb{C} = \{000, 011\}$ that may be caused by the first three bits in a bit vector $\mathbf{y} = \{y_j\}_{j=1}^{b}$, having a length $b > 3$. Particularly, the minimum HD $d_{2,3}^1$ between states $m_1 = 2$ and $m_1' = 3$ is given by $d_{2,3}^1 = d_{1,1}^0 + h(111, 000) = 3$. Since there are no legitimate paths leading to the states $m_1 = 1$ or $m_1 = 4$, we do not update the associated distances, as shown in Fig. 3. Similarly, we have $d_{1,2}^2 = d_{2,3}^1 + h(111, 011) = 4$, and $d_{1,2}^3 = \min(d_{1,2}^2 + h(000, 111), d_{1,4}^2 + h(000, 100), d_{2,3}^2 + h(111, 011), d_{3,4}^2 + h(011, 100)) = 4$. Once the forward recursion has considered a sufficient number of trellis stages for $\min(d_{1,1}^j, d_{1,2}^j, d_{2,2}^j) = \min(d_{1,1}^{j-1}, d_{1,2}^{j-1}, d_{2,2}^{j-1})$, then the approximate free distance becomes $d_f = \min(d_{1,1}^j, d_{1,2}^j, d_{2,2}^j)$.

Our set of candidate component UEC codes was further reduced by considering their free distances. More specifically, in order to achieve a wide variety of EXIT function shapes, we retained only UEC codebooks having the maximal or minimal free distances for each combination of $n \in \{2, 3, 4\}$ and $r \in \{2, 4\}$, where a free distance of 3 is the minimal value that facilitates convergence to the $(1, 1)$ point [24] and avoids an error floor. We drew the EXIT functions for all remaining candidate component UEC codes and selected the five codebooks offering the largest variety of EXIT function shapes, as listed in Table II. Our experiments revealed that only insignificant EXIT function shape variations are obtained, when considering more than $r = 4$ states. Without loss of generality, our irregular trellis example of Fig. 2 is constructed by concatenating the five UEC codebooks of Table II. In the following simulations, we will consider irregular trellises that are constructed using these codebooks. However, the number of states r employed by our five UEC component codes can be optionally and independently increased in the receiver, in order to facilitate nearer-to-capacity operation at the cost of an increased decoding complexity [10]. This is achieved by repeating the last element in the codebook. For example, while the transmitter may use the codebook $\mathbb{C} = \{00, 01\}$, the receiver may extend this to the $r = 10$-state codebook $\mathbb{C} = \{00, 01, 01, 01, 01\}$. Fig. 4 plots the inverted EXIT functions of the component UEC codes $\{\text{UEC}^s\}_{s=1}^5$, when extended to $r = 10$ states. Note that, similar to the IrURC EXIT function, the composite IrUEC EXIT function f_{IrUEC} is given as a weighted average of the component EXIT functions $\{f_{\text{UEC}^s}\}_{s=1}^5$, where we have

$$f_{\text{IrUEC}} = \sum_{s=1}^{5} \alpha_s \cdot f_{\text{UEC}^s}. \tag{9}$$

TABLE II
AFTER INVERTING AND SWAPPING, WE SELECT THE IRUEC COMPONENT UEC CODEBOOKS $\{\mathbb{C}_s\}_{s=1}^5$ WITH n BITS AND r STATES BOTH UP TO 4. ALL THE CODEBOOKS ARE IN THE FORMAT $(\mathbb{C}_s, d_f)$, WHERE d_f IS THE APPROXIMATE FREE DISTANCE

n	$r = 2$	$r = 4$
2		({00, 01}, 3)
3	({000}, 3)	({000, 011}, 4)
4	({0000}, 4)	({0000, 0111}, 5)

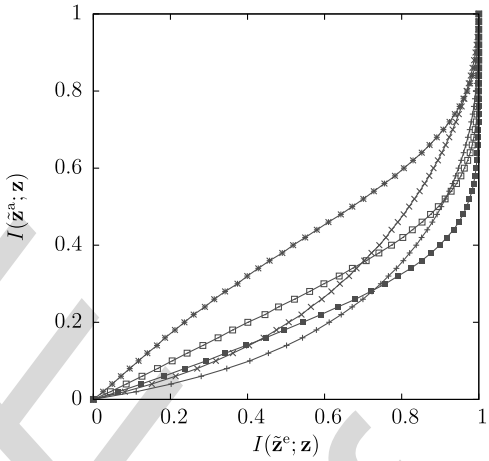


Fig. 4. Inverted EXIT functions for the $S = 5$ component UEC codes $\{\text{UEC}^s\}_{s=1}^5$ of Table II, when extended to $r = 10$ states codebooks, and when the symbol values obey a zeta probability distribution having the parameter value $p_1 = 0.797$.

B. Double-Sided EXIT Chart Matching Algorithm

The sixth column of Table III provides the specific E_b/N_0 values, where the DCMC capacity becomes equal to the throughput η of each scheme considered. These E_b/N_0 values represent the *capacity bound*, above which it is theoretically possible to achieve reliable communication. Note that the capacity bound is a function of the overall effective throughput η of the proposed IrUEC scheme, as described in Section II-C. In turn, the overall effective throughput η depends on the

TABLE III
CHARACTERISTICS OF THE VARIOUS SCHEMES CONSIDERED, INCLUDING OUTER CODING RATE $R_{\rm o}$, INNER CODING RATE $R_{\rm I}$ AND EFFECTIVE THROUGHPUT η. $E_{\rm B}/N_0$ BOUNDS ARE GIVEN FOR THE CASE OF GRAY-CODED QPSK TRANSMISSION OVER AN UNCORRELATED NARROWBAND RAYLEIGH FADING CHANNEL. COMPLEXITY IS QUANTIFIED BY THE AVERAGE NUMBER OF ACS OPERATIONS INCURRED PER DECODING ITERATION AND PER BIT IN THE VECTOR $\mathbf{z}$

1	2	3	4	5	6	7	8	9
Scheme	Codebooks	$R_{\rm o}$	$R_{\rm i}$	η	$E_{\rm b}/N_0$ [dB] capacity bound	$E_{\rm b}/N_0$ [dB] area bound	$E_{\rm b}/N_0$ [dB] tunnel bound	Complexity
IrUEC-IrURC	$\{\text{UEC}^s\}_{s=1}^{5}$	0.254	1	0.508	-0.05	0.21	0.3	258
IrUEC(med)-IrURC						0.30	0.6	192
IrUEC(low)-IrURC						1.14	1.2	157
UEC-IrURC	{000, 011}					0.49	1.7	120
EG-IrCC-IrURC	$\{\mathbf{CC}_{\rm sys}^s\}_{s=1}^{13}$	0.667	0.576			1.72	2.0	341
	$\{\mathbf{CC}_{\rm ns}^s\}_{s=1}^{11}$	0.254	1			1.02	1.1	146
EG-CC-IrURC	([4,7,7],6,6)					1.00	2.2	132

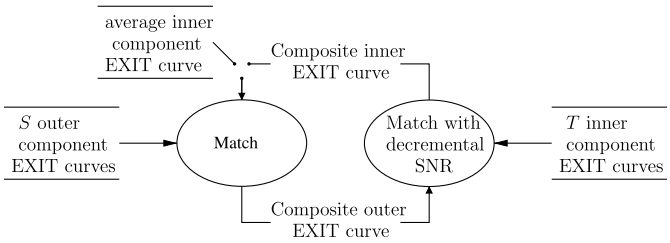


Fig. 5. Data-flow diagram of the proposed double-sided EXIT chart matching algorithm.

IrUEC coding rate $R_{\rm IrUEC}$, which depends on the entropy of the zeta distribution H_X, as described in Section II-A. In order to facilitate the creation of an open EXIT chart tunnel, it is necessary, but not sufficient, for the area $A_{\rm o}$ beneath the inverted outer EXIT function to exceed the area $A_{\rm i}$ beneath the inner EXIT function [17]. Therefore, the *area bound* provides the $E_{\rm b}/N_0$ values where we have $A_{\rm o} = A_{\rm i}$, which would theoretically allow the creation of an open EXIT chart tunnel [27], if the outer and inner EXIT functions were shaped to match each other. Here, $A_{\rm o}$ and $A_{\rm i}$ are the areas beneath the outer and inner EXIT functions, respectively. Depending on how well the EXIT functions match each other, a narrow but open EXIT chart tunnel can only be created at a specific $E_{\rm b}/N_0$ value, which we refer to as the *tunnel bound*. Based on these observations, the $E_{\rm b}/N_0$ difference between the capacity bound and the area bound quantifies the capacity loss that is mitigated by JSCC, while the difference between the area bound and the tunnel bound quantifies the capacity loss that is mitigated by irregular coding [28]. Based on this observation, our double-sided EXIT chart matching algorithm may be iteratively applied in order to match a pair of composite outer and inner EXIT functions, which are formed as a combination of S component UEC EXIT functions and T constituent URC EXIT functions, where the latter depend on the $E_{\rm b}/N_0$ value of the channel. In this way, a narrow but open EXIT chart tunnel between the inverted IrUEC EXIT function and the inner IrURC EXIT function may be created at $E_{\rm b}/N_0$ values that approach the capacity and area bounds, hence avoiding capacity loss and facilitating near-capacity operation.

As depicted in the data-flow diagram of Fig. 5, the algorithm commences by selecting the fractions α, in order to yield an IrUEC code design having a particular coding rate $R_{\rm IrUEC}$ and a composite IrUEC EXIT function that is shaped to match the average of T URC EXIT functions that correspond to a particular $E_{\rm b}/N_0$ value. The technique of [14] may be employed for selecting the fractions β, in order to yield a composite IrURC EXIT function that is shaped to match that of the IrUEC code. Following this, the algorithm alternates between the matching of the composite IrUEC EXIT function to the composite IrURC EXIT function and vice versa, as shown in Fig. 5. In order to facilitate near-capacity operation, we use a 0.1 dB $E_{\rm b}/N_0$ decrement per iteration for the component URC EXIT functions, when designing the fractions β for the IrURC code, until we find the lowest $E_{\rm b}/N_0$ value that achieves a marginally open EXIT tunnel. Note that the double-sided EXIT chart matching algorithm allows the design of an IrUEC code having a specific coding rate $R_{\rm IrUEC}$. This enables us to design the IrUEC code to have a coding rate of $R_{\rm IrUEC} = 0.254$, which provides a fair performance comparison with the regular UEC-IrURC scheme of [10] and with other benchmarkers, as detailed in Section V. More specifically, this results in the same overall effective throughput of $\eta = R_{\rm IrUEC} \cdot R_{\rm IrURC} \cdot \log_2(M) = 0.508$ bit/s/Hz, as listed in Table III.

For the IrURC encoder, we employ the $T = 10$-component URC codes $\{\text{URC}^t\}_{t=1}^{10}$ of [20], [29]. After running the double-sided EXIT chart matching algorithm of Fig. 5 until the $E_{\rm b}/N_0$ value cannot be reduced any further without closing the EXIT chart tunnel, the composite EXIT functions of the IrUEC and IrURC schemes are obtained, as depicted in Fig. 6(a). Here, the $E_{\rm b}/N_0$ value is 0.3 dB, which is 0.35 dB away from the DCMC capacity bound of -0.05 dB and was found to be the lowest one that creates an open EXIT chart tunnel. More specifically, the fractions of the bit vector $\mathbf{z}$ that are generated by the constituent UEC codes $\{\text{UEC}^s\}_{s=1}^{5}$ of the IrUEC encoder are $\alpha = [0\ 0.7240\ 0.0924\ 0\ 0.1836]$, respectively. Similarly, the fractions of the bit vector $\mathbf{u}$ that encoded by the constituent URC codes $\{\text{URC}^t\}_{t=1}^{10}$ of the IrURC encoder are $\beta = [0.1767\ 0\ 0.8233\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively.

V. BENCHMARKERS AND SIMULATIONS

In this section, we compare the SER performance of the proposed IrUEC-IrURC scheme of Fig. 1 to that of various SSCC and JSCC benchmarkers. As mentioned in Section IV, the proposed IrUEC-IrURC scheme and all benchmarkers are designed to have the same effective overall throughput of

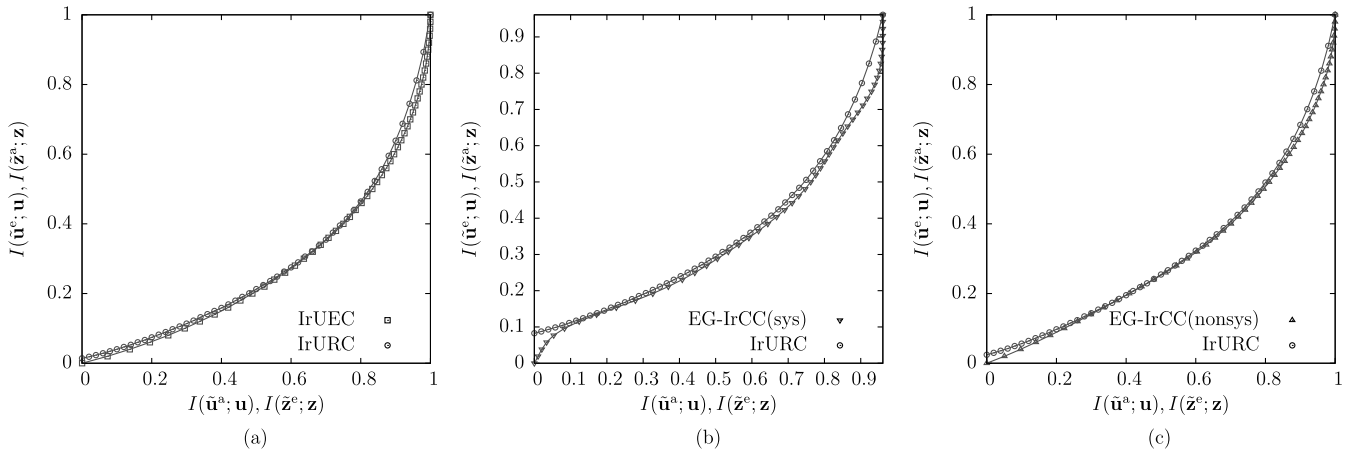


Fig. 6. Composite EXIT functions of (a) the IrUEC decoder employing $S = 5$ component UEC codes $\{\text{UEC}^s\}_{s=1}^{5}$, (b) the EG-IrCC decoder employing the $S = 13$ component recursive systematic CC codes $\{\mathbf{CC}_{\text{sys}}^s\}_{s=1}^{13}$ and (c) the EG-IrCC scheme employing the $S = 11$ component non-systematic CC codes $\{\mathbf{CC}_{\text{ns}}^s\}_{s=1}^{11}$, and the IrURC scheme employing the $T = 10$ component URC codes $\{\text{URC}^t\}_{t=1}^{10}$, when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel. The EXIT chart tunnel is marginally open when $E_b/N_0 = 0.3$, 2.0 and 1.1 dB, respectively.

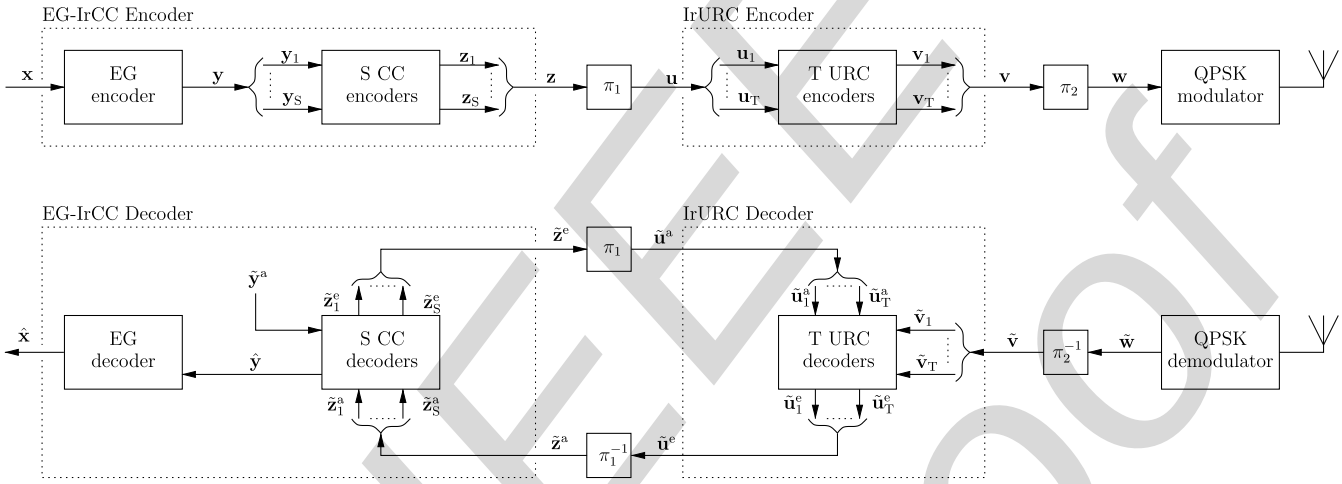


Fig. 7. Schematic of the EG-IrCC-IrURC benchmarker, in which an EG-IrCC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

$\eta = 0.508$ bit/s/Hz, for the sake of fair comparison. A pair of benchmarkers are constituted by the UEC-IrURC and EG-CC-IrURC schemes of our previous work [10]. Furthermore, a new benchmarker is created by replacing the unary encoder and the IrTrellis encoder in the transmitter of Fig. 1 with an EG encoder and an IrCC encoder, respectively. This results in the SSCC benchmarker of Fig. 7, which we refer to as the EG-IrCC-IrURC scheme. Table I shows the first ten codewords of the EG code, which are used for encoding the symbol vector $\mathbf{x}$.

As in the IrUEC-IrURC scheme, the bit vector $\mathbf{y}$ output by the EG encoder may be modeled as a realization of vector $\mathbf{Y} = [Y_j]_{j=1}^{b}$ having binary RVs. However, as observed in [10], these RVs do not adopt equiprobable values $\Pr(Y_j = 0) \neq \Pr(Y_j = 1)$, hence giving a less than unity value for the corresponding bit entropy H_{Y_j}. Similarly, the bit vector $\mathbf{z}$ of Fig. 7 may be modeled as a particular realization of a vector $\mathbf{Z} = [Z_k]_{k=1}^{b\bar{n}}$ comprising $b\bar{n}$ binary RVs. Each binary RV Z_k adopts the values 0 and 1 with the probabilities $\Pr(Z_k = 0)$ and $\Pr(Z_k = 1)$ respectively, corresponding to a bit entropy of H_{Z_k}. In the case where the IrCC code employs systematic component codes, the bits of $\mathbf{y}$ having the entropy $H_{Y_j} < 1$ will appear in $\mathbf{z}$, resulting in a bit entropy of $H_{Z_k} < 1$. However, a bit entropy of $H_{Z_k} < 1$ is associated with a capacity loss, as described in [10].

Hence, for the sake of avoiding any capacity loss, it is necessary to use non-systematic recursive component codes, so that the bits in the resultant encoded vector $\mathbf{z}$ have equiprobable values [10]. In order to demonstrate this, we introduce two versions of the EG-IrCC-IrURC benchmarker. Firstly, the $N = 13$ recursive systematic component CC codes [15] $\{\mathbf{CC}_{\text{sys}}^s\}_{s=1}^{13}$ that were originally proposed for IrCC encoding are adopted in the EG-IrCC-IrURC encoder, as it will be described in Section V-A. Secondly, Section V-B employs the $S = 11$ non-systematic recursive CC codebooks $\{\mathbf{CC}_{\text{ns}}^s\}_{s=1}^{11}$ proposed in [20], in order to offer an improved version of the EG-IrCC benchmarker. Meanwhile, the 10 component URC codebooks $\{\text{URC}^t\}_{t=1}^{10}$ employed by the IrURC encoder in both versions of the benchmarker of Fig. 7 are identical to those in the IrURC encoder of Fig. 1.

A. Recursive Systematic Component CC Codes

The recursive systematic CC codes $\{\mathbf{CC}^s_{\text{sys}}\}^{13}_{s=1}$ employed in [15] were designed to have coding rates of $R_{\text{CC}^s_{\text{sys}}} \in \{0.1, 0.15, \ldots, 0.65, 0.7\}$. However, since the EG-encoded bits in the vector $\mathbf{y}$ are not equiprobable, none of the systematic bits in the bit vector $\mathbf{z}$ will be equiprobable either. As a result, the coding rate $R_{\text{CC}^s_{\text{sys}}} = \frac{H_{Y_j}}{n_{\text{CC}^s_{\text{sys}}} \cdot H_{Z_k}^{\text{CC}^s_{\text{sys}}}}$ of each systematic CC will be lower than the above-mentioned values. Since each CC code $\mathbf{CC}^s_{\text{sys}}$ produces a different number of systematic bits, each will have a different bit entropy $H_{Z_k}^{\text{CC}^s_{\text{sys}}}$, and the EXIT function of each CC code will converge to a different point $(H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}, H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}})$ in the EXIT chart [30]. The composite IrCC EXIT function will converge to a point $(H_{Z_k}^{\text{IrCC}}, H_{Z_k}^{\text{IrCC}})$, where $H_{Z_k}^{\text{IrCC}}$ is given by a weighted average of $\{H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}\}^{13}_{s=1}$, according to

$$H_{Z_k}^{\text{IrCC}} = \sum_{s=1}^{13} \alpha_s \cdot H_{Z_k}^{\mathbf{CC}^s_{\text{sys}}}. \tag{10}$$

Since the vector $\mathbf{z}$ is interleaved to generate the bit vector $\mathbf{u}$ as the input of the IrURC encoder, the IrURC EXIT function will also converge to $(H_{Z_k}^{\text{IrCC}}, H_{Z_k}^{\text{IrCC}})$. However, this presents a particular challenge, when parametrizing the fractions α and β of the EG-IrCC(sys)-IrURC scheme. More specifically, the fractions α vary as our double-sided EXIT chart matching algorithm progresses, causing the entropy $H_{Z_k}^{\text{IrCC}}$ to vary as well. This in turn causes the IrURC EXIT function to vary, creating a cyclical dependency that cannot be readily resolved. More specifically, the fractions α must be selected to shape the EG-IrCC EXIT function so that it matches the IrURC EXIT function, but the IrURC EXIT function depends on the fractions α selected for the EG-IrCC EXIT function.

Owing to this, we design the fractions α and β by assuming that the bits of $\mathbf{y}$ are equiprobable and by plotting the inverted EXIT functions for the $S = 13$ recursive systematic CC codes accordingly, giving convergence to the $(1, 1)$ point in Fig. 6(b). Then we invoke our double-sided EXIT matching algorithm to design the fractions α and β for the IrCC(sys) and IrURC codes, which we apply to the EG-IrCC(sys)-IrURC scheme. For the case where the bits of the vector $\mathbf{y}$ have the non-equiprobable values that result from EG encoding, the composite EXIT functions are shown in Fig. 6(b). Here, the effective throughput is $\eta = 0.508$ bit/s/Hz and the E_b/N_0 value is 2.0 dB, which is the lowest value for which an open EXIT chart tunnel can be created. This E_b/N_0 tunnel bound is 2.05 dB away from the DCMC capacity bound of -0.05 dB, owing to the above-mentioned capacity loss. Furthermore, the EG-IrCC(sys)-IrURC scheme has an area bound of 1.72 dB, which corresponds to a capacity loss of 1.77 dB, relative to the capacity bound. The designed fractions for the EG-IrCC scheme are $\alpha = [0.0620\ 0.2997\ 0.0497\ 0.0004\ 0.1943\ 0\ 0.0984\ 0.1285\ 0\ 0\ 0\ 0.0002\ 0.1668]$, while the fractions for the IrURC code are $\beta = [0.6548\ 0\ 0.3452\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively.

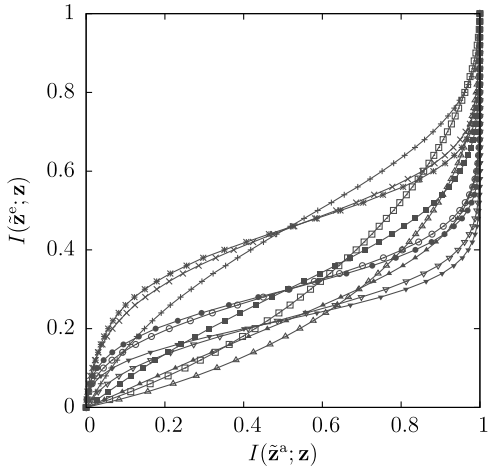


Fig. 8. Inverted EXIT functions for EG-CC code, for the case where the $S = 11$ component recursive non-systematic CC codes $\{\mathbf{CC}^s_{\text{ns}}\}^{11}_{s=1}$ are employed, and the symbol values obey a zeta probability distribution having the parameter value $p_1 = 0.797$.

B. Recursive Non-Systematic Component CC Codes

In order to avoid the capacity loss introduced by the recursive systematic CC codes, we advocate the recursive non-systematic CC codebooks $\{\mathbf{CC}^s_{\text{ns}}\}^{11}_{s=1}$, which are described by the generator and feedback polynomials provided in [10, Table II]. More specifically, of the 12 codes presented in [10, Table II], we use all but the $r = 2$, $n = 2$ code, for the sake of avoiding an error floor. These recursive non-systematic CC codes attain the optimal distance properties [31] subject to the constraint of producing equiprobable bits $\Pr(Z_j = 0) = \Pr(Z_j = 1)$, which is necessary for avoiding any capacity loss. The inverted EXIT functions are plotted in Fig. 8.

For the sake of a fair comparison, we apply the double-sided EXIT chart matching algorithm of Fig. 5 again to design the EG-IrCC(nonsys)-IrURC scheme having a coding rate of $R_{\text{EG-IrCC}} = 0.254$ and an effective throughput of $\eta = 0.508$ bit/s/Hz. The composite EXIT functions of the EG-IrCC(nonsys) and IrURC schemes are shown in Fig. 6(c). Here, the fractions of the EG-IrCC scheme are $\alpha = [0.8101\ 0\ 0.0643\ 0\ 0\ 0\ 0\ 0.1256\ 0\ 0\ 0]$, while the fractions of the IrURC code are $\beta = [0.2386\ 0\ 0.7614\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, respectively. The EXIT chart of Fig. 8 is provided for an E_b/N_0 value of 1.1 dB, which is the lowest value for which an open EXIT chart tunnel is created. As shown in Table III, this E_b/N_0 tunnel bound is just 1.15 dB away from the DCMC capacity bound of -0.05 dB. This improvement relative to the EG-IrCC(sys)-IrURC scheme may be attributed to the non-systematic nature of the EG-IrCC(nonsys)-IrURC scheme, which has reduced the capacity loss to 1.07 dB, as quantified by considering the difference between the E_b/N_0 area bound of 1.02 dB and the capacity bound.

C. Parallel Component UEC Codes

In order to make a comprehensive comparison, we also consider a Parallel IrUEC-IrURC scheme. As shown in Fig. 9, this scheme employs a parallel concatenation of S number

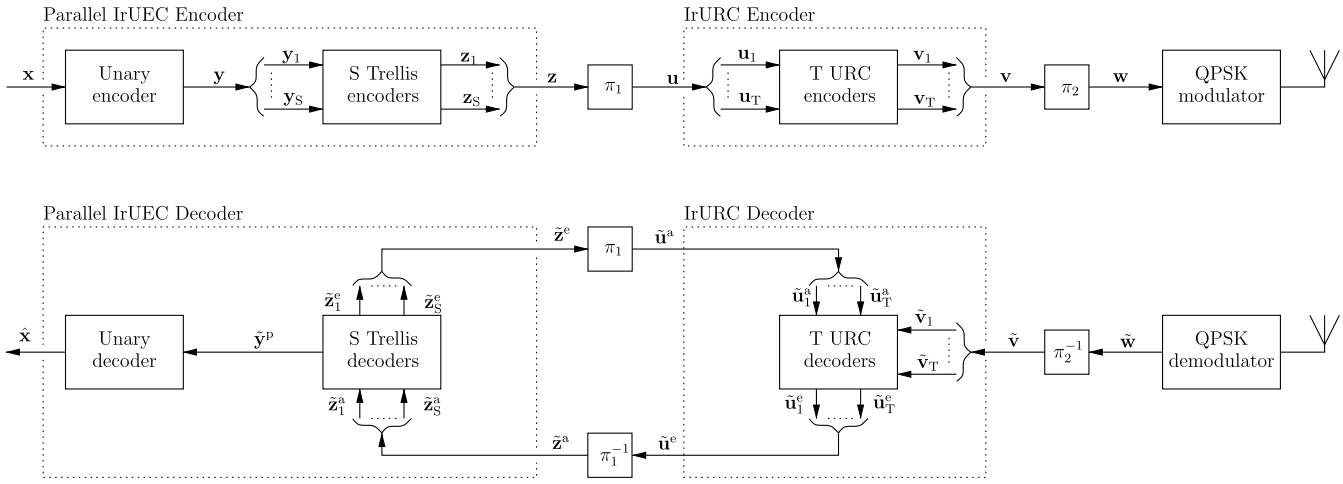


Fig. 9. Schematic of the Parallel IrUEC-IrURC benchmarker, in which a parallel IrUEC code is serially concatenated with IrURC code and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers.

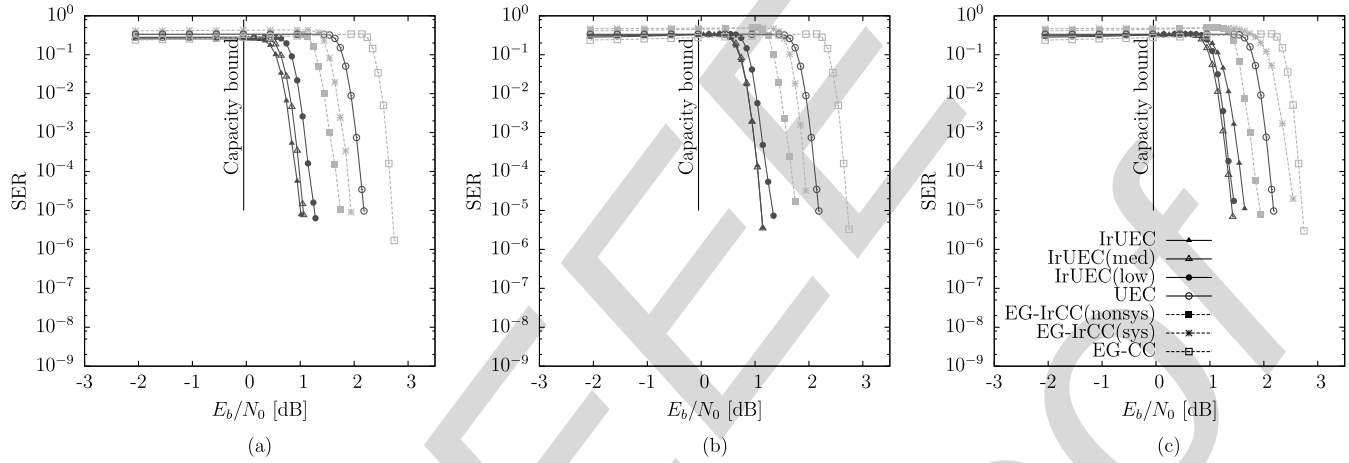


Fig. 10. SER performance for various arrangements of the proposed IrUEC-IrURC scheme of Fig. 1, the EG-IrCC-IrURC of Fig. 7, the Parallel IrUEC-IrURC scheme of Fig. 9, as well as the UEC-IrURC and the EG-IrURC schemes of [10], when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values. A complexity limit of (a) unlimited, (b) 10,000 and (c) 5,000 ACS operations per decoding iteration is imposed for decoding each of the bits in $\mathbf{z}$.

of separate UEC trellis encoders to encode the bit vector $\mathbf{y}$, in analogy with the structure of the EG-IrCC scheme. More specifically, the component UEC codes of the Parallel IrUEC encoder are selected from the five constituent codes provided in Table II, while the component UEC codes of the Parallel IrUEC decoder are extended to $r = 10$ states. The irregular fractions employed by the Parallel IrUEC scheme are the same as those used in our proposed IrUEC scheme. However, in order for each component UEC trellis encoder to remain synchronized with the unary codewords in the bit vector $\mathbf{y}$, it is necessary for each component trellis to commence its encoding action from state $m_0 = 1$ and end at state $m_b = 1$ or $m_b = 2$. Owing to this, the subvectors of $\mathbf{y}$ input to each component UEC must comprise an integer number of complete unary codewords. The irregular coding fractions can only be controlled at the symbol level in the case of the parallel IrUEC scheme, rather than at the bit level, as in the proposed IrUEC scheme. Therefore, the corresponding EXIT chart of the parallel IrUEC scheme is not guaranteed to have an open tunnel, when the E_b/N_0 value approaches the tunnel bound of Table III, hence resulting in a degraded SER performance. However, if the frame length a was orders of magnitude higher, the difference between the symbol-based and bit-based segmentations of the bit vector $\mathbf{y}$ would become insignificantly small. As a result, a similar SER performance may be expected for the parallel IrUEC scheme in this case. In the following section, we will compare the performances of the Parallel IrUEC and the proposed IrUEC schemes, using different values for the frame length a.

D. SER Results

The SER performance of the IrUEC-IrURC, the EG-IrCC(sys)-IrURC and the EG-IrCC(nonsys)-IrURC, UEC-IrURC and EG-CC-IrURC schemes is characterised in Fig. 10. In each case, the source symbol sequence $\mathbf{x}$ comprises $a = 10^4$ symbols, the values of which obey a zeta distribution having a parameter value of $p_1 = 0.797$. As shown above, the parametrizations of the irregular codes in each scheme are

designed to achieve the closest possible matching of EXIT charts, while giving the same overall effective throughput of $\eta = 0.508$ bit/s/Hz. Transmission is performed over a Gray-coded QPSK-modulated uncorrelated narrowband Rayleigh fading channel, resulting in the DCMC capacity bound of -0.05 dB. We select two parametrizations of the schemes in [10] to create two of our four benchmarkers, namely the $r = 4$-state UEC-IrURC and the $r = 4$-state EG-CC-IrURC schemes. Note that the $r = 4$-state EG-CC-IrURC scheme was found to outperform other parametrizations of the same scheme having higher number of states, owing to its superior EXIT chart matching accordingly. With the same effective throughput η, a fair comparison is provided between our proposed IrUEC-IrURC scheme and the four benchmarkers.

Note that the practical implementation of the time-variant IrTrellis used in our IrUEC-IrURC scheme follows the same principles as the parallel time-invariant trellises of the benchmarker schemes, such as the EG-IrCC-IrURC scheme and the regular UEC-IrURC scheme. Once the irregular coding fractions have been determined, the specific portions of message that should be encoded and decoded by the corresponding trellises are also determined. In both time-variant and parallel time-invariant trellises, the hardware is required to support different trellis structures, which may be implemented by appropriately changing the connections among the states of a single hardware implementation of a trellis. Although the proposed time-invariant trellis has some peculiarities at the interface between its different sections, these can also be implemented using the same hardware at either side of the interface. As an example platform for hardware implementation, the computation unit of [32] performs one ACS arithmetic operation per clock cycle, which are the fundamental operations used in BCJR decoders [18]. Therefore, the implementational complexity depends only on the computational complexity, as quantified per decoding iteration in Table III. Since a common computational complexity limit is used in our comparisons of the various schemes, they can be deemed to have the same implementational complexity. Although the routing and control of the proposed IrTrellis may be expected to be more complicated than in the parallel time-invariant trellises of the benchmarkers, it may be expected that the associated overhead is negligible compared to the overall implementational complexity.

As shown in Table III, our IrUEC-IrURC scheme imposes a complexity of 258 ACS operations per iteration per bit, when employing $r = 10$ states for each component UEC code in the IrTrellis decoder. We also consider alternative parametrizations of our IrUEC-IrURC scheme, which employ an IrTrellis having fewer states, in order to achieve lower complexities. The IrUEC(med)-IrURC scheme relies on $r = 6$ trellis states for different stages of the IrTrellis, which results in a total complexity of 192 ACS operations per iteration per bit. This matches that of the UEC-IrURC benchmarker. At the same time, the IrUEC(low)-IrURC scheme employs the minimal number of states for each stage of the IrTrellis, namely either $r = 4$ states, as listed in Table II, hence resulting in a complexity of 157 ACS operations per iteration per bit.

During the simulation of each scheme, we recorded both the SER and the complexity incurred after each decoding iteration, resulting in a 3D plot of SER versus E_b/N_0 and versus complexity. Fig. 10 presents 2D plots of SER versus E_b/N_0 relationship, which were obtained by slicing through these 3D plots at a particular complexity. More specifically, we select the complexity limits of $10,000$ and $5,000$ ACS operations per iteration per bit in Fig. 10(b) and (c), respectively. Meanwhile, Fig.10 (a) characterizes the SER performance achieved after iterative decoding convergence, regardless of the complexity.

As shown in Table III, the proposed IrUEC-IrURC scheme has an area bound of 0.21 dB, which is the E_b/N_0 value where the area A_o beneath the inverted IrUEC EXIT function equals that beneath the IrURC EXIT function. Although the UEC-IrURC benchmarker has a similar area bound of $E_b/N_0 = 0.49$ dB, it has an inferior EXIT chart matching capability owing to its employment of regular UEC constituent codes. By contrast, the employment of two irregular codes in the proposed IrUEC-IrURC scheme facilitates an open EXIT chart tunnel at an E_b/N_0 value of 0.3 dB, which is 1.4 dB lower than the open tunnel bound of the UEC-IrURC benchmarker. Note that the area and tunnel bounds are degraded in the context of the lower complexity versions of the proposed IrUEC-IrURC scheme, which have fewer states in the IrTrellis. This may be explained by the increased capacity loss encountered when the number of UEC states is reduced [10]. Note however that even with a reduced complexity, the proposed IrUEC-IrURC scheme tends to exhibit superior area and tunnel bounds, when compared to the EG-IrCC-IrURC and EG-CC-IrURC benchmarkers, as shown in Table III. This may be attributed to the large capacity loss that is associated with SSCC scheme [10].

Fig. 10 demonstrates that our proposed IrUEC-IrURC scheme has a superior SER performance compared to all other benchmarkers, regardless of which complexity limit is selected in this particular scenario. For example, as shown in Fig. 10(a), our IrUEC-IrURC scheme facilitates operation within 0.4 dB of the capacity bound, offering a 0.8 dB gain compared to the EG-IrCC(nonsys)-IrURC scheme, which is the best-performing of the SSCC benchmarkers. This is achieved without any increase in transmission energy, bandwidth, transmit duration or decoding complexity. Note that the EG-IrCC(nonsys)-IrURC benchmarker offers a 0.9 dB gain over the EG-IrCC(sys)-IrURC benchmarker, which is owing to the capacity loss that is associated with systematic IrCC component codes. As expected, the reduced complexity versions of the proposed IrUEC-IrURC scheme exhibit a degraded SER performance. However, the IrUEC(low)-IrURC scheme can be seen to offer up to 0.5 dB gain over the UEC-IrURC benchmarker, which has a close decoding complexity per bit per iteration. Since the Parallel IrUEC-IrURC scheme can only provide a symbol-level control of the irregular coding fractions, the EXIT chart tunnel is not guaranteed to be open at low E_b/N_0 values. As a result, Fig. 11 shows that the Parallel IrUEC-IrURC scheme of Fig. 9 performs relatively poorly compared to the proposed IrUEC-IrURC scheme, particularly when the frame length has values of $a = 10^2$ and $a = 10^3$ symbols. Note that this performance gain offered by the proposed scheme is obtained without imposing any additional decoding complexity and without requiring any additional transmission-energy, -bandwidth, or -duration. In analogy with Fig. 10(a), an additional set of SER results

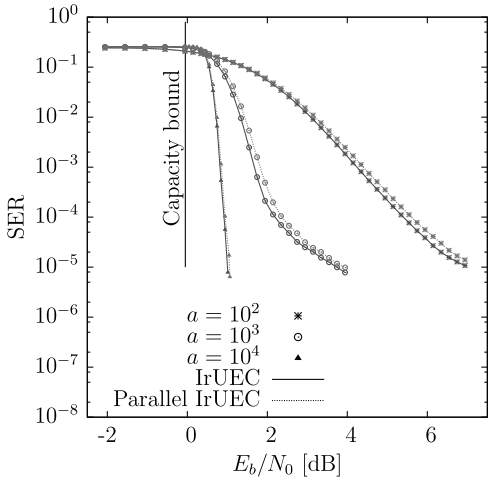


Fig. 11. SER performance for various frame lengths $a \in \{10^2, 10^3, 10^4\}$ of the proposed IrUEC-IrURC scheme of Fig. 1 and the Parallel IrUEC-IrURC scheme of Fig. 9, when conveying symbols obeying a zeta distribution having the parameter $p_1 = 0.797$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values.

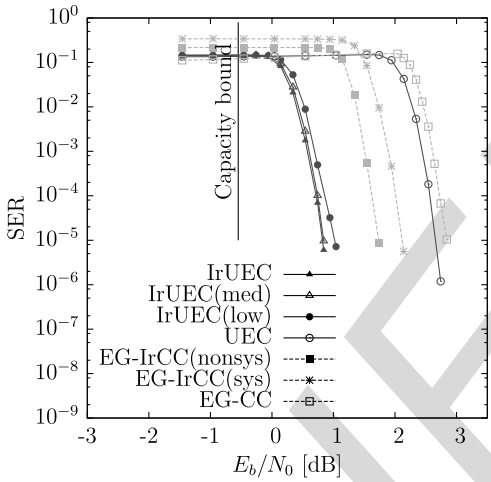


Fig. 12. SER performance for various arrangements of the proposed IrUEC-IrURC scheme of Fig. 1, the EG-IrCC-IrURC of Fig. 7, the Parallel IrUEC-IrURC scheme of Fig. 9, as well as the UEC-IrURC and the EG-IrURC schemes of [10], when conveying symbols obey a zeta distribution having the parameter $p_1 = 0.9$, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel having a range of E_b/N_0 values. The complexity is unlimited for decoding each of the bits in $\mathbf{z}$.

is provided in Fig. 12 for the various schemes considered, where the source symbols obey a zeta distribution having the parameter $p_1 = 0.9$, where the complexity is potentially unlimited. It can be seen that the proposed IrUEC-IrURC scheme also outperforms all other benchmarkers in this situation, offering a 1 dB gain compared to the EG-IrCC(nonsys)-IrURC scheme, which is the best-performing one of the set of SSCC benchmarkers.

Note that the performance gain of the proposed IrUEC-IrURC scheme is obtained by elaborately designing the IrUEC EXIT function, in order to create a narrow but marginally open EXIT chart tunnel at a low E_b/N_0 value that is close to the area bound and capacity bound, as discussed in Section IV-B. Since the benchmarker schemes suffer from capacity loss which separates their tunnel, area and capacity bounds, the performance gain of the proposed IrUEC-IrURC scheme depicted in Fig. 10 and 12 may be expected in the general case, regardless of the specific source probability distribution and the parametrization of the scheme. As an additional benefit of the proposed IrUEC-IrURC scheme, a single bit error within a particular codeword can only result in splitting it into two codewords, or into merging it with the next codeword, since every unary codeword contains only a single 0. Fortunately, the decoding of the other unary codewords will be unaffected. Owing to this, a single bit error in the IrUEC-IrURC scheme can only cause a Levenshtein distance [33] of 2, hence preventing error propagation. By contrast, in the EG-based benchmarkers, a single bit error can cause error propagation, resulting in a Levenshtein distance that is bounded only by the length of the message.

VI. Conclusions

In this paper, we have proposed a novel near-capacity JSCC scheme, which we refer to as the IrUEC code. Like the regular UEC code of [10], this employs a unary code, but replaces the UEC's trellis code with a novel IrTrellis code. Unlike a conventional irregular code, the IrTrellis code operates on the basis of a single amalgamated irregular trellis, rather than a number of separate trellises. Our results demonstrated that this single amalgamated trellis offers gains of up to 0.2 dB over the use of separate trellises, without imposing any increase in transmission energy, bandwidth, latency or decoding complexity. By characterizing the free distance property of the UEC trellis, we have selected a suite of UEC codes having a wide variety of EXIT chart shapes for the component codes of our IrUEC code. We concatenated the proposed IrUEC code with an IrURC code in Fig. 1 and introduced a new double-sided EXIT chart matching algorithm. On the one hand, the component UEC codes having a wide variety of EXIT chart shapes provide a great design freedom of the IrUEC EXIT chart. On the other hand, the novel double-sided EXIT chart matching algorithm utilize this design freedom sufficiently, in order to parametrize the IrUEC-IrURC scheme for creating a narrow but marginally open EXIT chart tunnel at a low E_b/N_0 value that is close the area bound and the capacity bound. As a result, near-capacity operation is facilitated at E_b/N_0 values that are within 0.4 dB of the DCMC capacity bound, when achieving an effective throughput of $\eta = 0.508$ bit/s/Hz and employing (QPSK) for transmission over an uncorrelated narrowband Rayleigh fading channel. This corresponds to a gain of 0.8 dB compared to the best of several SSCC benchmarkers, which is achieved without any increase in transmission energy, bandwidth, transmit duration or decoding complexity.

References

[1] J. Zou, H. Xiong, C. Li, R. Zhang, and Z. He, "Lifetime and distortion optimization with joint source/channel rate adaptation and network coding-based error control in wireless video sensor networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1182–1194, Mar. 2011.

[2] Y. Huo, C. Zhu, and L. Hanzo, "Spatio-temporal iterative source-channel decoding aided video transmission," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1597–1609, May 2013.

[3] N. S. Othman, M. El-Hajjar, O. Alamri, S. X. Ng, and L. Hanzo, "Iterative AMR-WB source and channel decoding using differential space-time spreading-assisted sphere-packing modulation," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 484–490, Jan. 2009.

[4] C. E. Shannon, "The mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, Jul. 1948.

[5] B. Ryabko and J. Rissanen, "Fast adaptive arithmetic code for large alphabet sources with asymmetrical distributions," in *Proc. IEEE Int. Symp. Inf. Theory*, 2002, p. 319.

[6] J. Ziv and A. Lempel, "Compression of individual sequences via variable rate coding," *IEEE Trans. Inf. Theory*, vol. 24, no. 5, pp. 530–536, Sep. 1978.

[7] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 194–203, Mar. 1975.

[8] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Std. H.264, Mar. 2005.

[9] J. L. Massey, "Joint source and channel coding," in *Proc. Commun. Syst. Random Process Theory*, Dec. 1978, pp. 279–293.

[10] R. G. Maunder, W. Zhang, T. Wang, and L. Hanzo, "A unary error correction code for the near-capacity joint source and channel coding of symbol values from an infinite set," *IEEE Trans. Commun.*, vol. 61, no. 5, pp. 1977–1987, May 2013.

[11] N. L. Johnson, A. W. Kemp, and S. Kotz, *Univariate Discrete Distributions*. Hoboken, NJ, USA: Wiley, 2005.

[12] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 234, pp. 158–162, Feb. 1995.

[13] V. Buttigieg and P. G. Farrell, "Variable-length error-correcting codes," *IEE Proc. Commun.*, vol. 147, no. 4, pp. 211–215, Aug. 2000.

[14] 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.

[15] M. Tüchler, "Design of serially concatenated systems depending on the block length," *IEEE Trans. Commun.*, vol. 52, no. 2, pp. 209–218, Feb. 2004.

[16] R. G. Maunder and L. Hanzo, "Genetic algorithm aided design of component codes for irregular variable length coding," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1290–1297, May 2009.

[17] 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.

[18] W. Zhang, R. G. Maunder, and L. Hanzo, "On the complexity of unary error correction codes for the near-capacity transmission of symbol values from an infinite set," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2013, pp. 2795–2800.

[19] T. Wang, W. Zhang, R. G. Maunder, and L. Hanzo, "Near-capacity joint source and channel coding of symbol values from an infinite source set using Elias gamma error correction codes," *IEEE Trans. Commun.*, vol. 62, no. 1, pp. 280–292, Jan. 2014.

[20] L. Hanzo, R. G. Maunder, J. Wang, and L.-L. Yang, *Near-Capacity Variable Length Coding*. Hoboken, NJ, USA: Wiley, 2010.

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

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

[23] 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.

[24] J. Kliewer, N. Goertz, 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.

[25] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo like codes," in *Proc. 36th Allerton Conf. Commun. Control Comput.*, Allerton House, IA, USA, Sep. 1998, pp. 201–210.

[26] A. Diallo, C. Weidmann, and M. Kieffer, "Efficient computation and optimization of the free distance of variable-length finite-state joint source-channel codes," *IEEE Trans. Commun.*, vol. 59, no. 4, pp. 1043–1052, Apr. 2011.

[27] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.

[28] M. Tüchler and J. Hagenauer, "EXIT charts of irregular codes," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, USA, Mar. 2002, pp. 748–753.

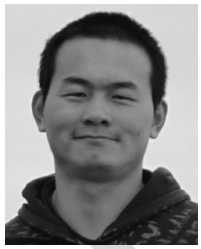
[29] L. Hanzo, T. H. Liew, B. L. Yeap, R. Y. S. Tee, and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart Aided Near-Capacity Designs for Wireless Channels*. Hoboken, NJ, USA: Wiley, 2010.

[30] 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, USA, Jul. 2006, pp. 2680–2684.

[31] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Commun. Lett.*, vol. 3, no. 11, pp. 317–319, Nov. 1999.

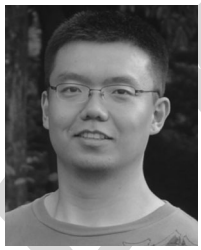
[32] L. Li, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "A low-complexity turbo decoder architecture for energy-efficient wireless sensor networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 14–22, Jan. 2013.

[33] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Reading, MA, USA: Addison-Wesley, 1983.

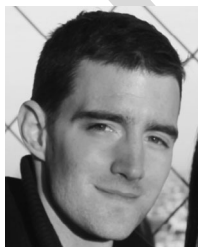


Wenbo Zhang (S'14) received the M.E. degree in information and communication engineering from the University of Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011. Currently, he is pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton, Southampton, U.K. His research interests include joint source/channel coding and variable length coding.

Matthew F. Brejza received the B.Eng. degree (first class Hons.) in electronic engineering from the University of Southampton, Southampton, U.K., in 2012, and is currently pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton. His research interests include flexible hardware implementation, channel coding and their applications in low power data communications.



Tao Wang received the B.S. degree in information engineering from the University of Science and Technology of Beijing (USTB), Beijing, China, and the M.Sc. degree in communication from University of Southampton, Southampton, U.K., in 2006 and 2008, respectively, and is currently pursuing the Ph.D. degree at the Communications Research Group, School of Electronics and Computer Science, University of Southampton. His research interests include joint source/channel coding and distributed video coding.



Robert G. Maunder (SM'12) received the B.Eng. degree (first class Hons.) in electronic engineering and the Ph.D. degree in wireless communications, in 2003 and 2007, respectively. He has authored a number of IEEE papers in his research interest areas. His research interests include joint source/channel coding, iterative decoding, irregular coding, and modulation techniques. He received a lectureship in December 2007.

Lajos Hanzo (F'04) received the degree in electronics and the doctorate degree, in 1976 and 1983, respectively. During his 38-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, Southampton, U.K., where he holds the Chair in telecommunications. He has coauthored 20 Wiley/IEEE Press books on mobile radio communications totalling in excess of 10000 pages, authored more than 1500 research entries at the IEEE Xplore, acted both as TPC and General Chair of the IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. Currently, he is directing a 60-strong 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 Research Council's Advanced Fellow Grant, and the Royal Society's Wolfson Research Merit Award. He is a Governor of the IEEE VTS. From 2008 to 2012, he was the Editor-in-Chief of the IEEE Press and also a Chaired Professor at Tsinghua University, Beijing, China. His research is funded by the European Research Council's Senior Research Fellow Grant. Lajos has more than 22 000 citations. In 2009, he was was the recipient of an Honorary Doctorate by the Technical University of Budapest, while in 2015, by the University of Edinburgh.