A Unary Error Correction Code for the Near-Capacity Joint Source and Channel Coding of Symbol Values from an Infinite Set

Robert G. Maunder, Wenbo Zhang, Tao Wang and Lajos Hanzo

***Abstract*—A novel Joint Source and Channel Code (JSCC) is proposed, which we refer to as the *Unary Error Correction (UEC) code*. Unlike existing JSCCs, our UEC facilitates the practical encoding of symbol values that are selected from a set having an infinite cardinality. Conventionally, these symbols are conveyed using Separate Source and Channel Codes (SSCCs), but we demonstrate that the residual redundancy that is retained following source coding results in capacity loss. This loss is found to have a value of 1.11 dB in a particular practical scenario, where Quaternary Phase Shift Keying (QPSK) modulation is employed for transmission over an uncorrelated narrowband Rayleigh fading channel. By contrast, the proposed UEC code can eliminate this capacity loss, or reduce it to an infinitesimally small value. Furthermore, the UEC code has only a moderate complexity, facilitating its employment in practical low-complexity applications.**

***Index Terms*—Source coding, video coding, channel coding, channel capacity, iterative decoding.**

I. Introduction

MULTIMEDIA codecs such as the H.264 video codec [1] often employ universal source codes such as the Elias Gamma (EG) code[1] [2]. These universal codes are designed for representing symbol values that are selected using a particular probability distribution from a set having an infinite cardinality, such as the set of all positive integers, as discussed in Section II. However, following the application of these source codes, typically some residual redundancy [3] is retained, which imposes capacity loss and therefore prevents near-capacity operation. As a potential remedy, Joint Source and Channel Codes (JSCCs) have been proposed for exploiting residual redundancy and avoiding capacity loss [4]. However, infinite-cardinality symbol value sets are impractical for existing JSCCs, such as Self-Synchronizing Variable Length Codes (SSVLCs) [5], Reversible Variable Length Codes (RVLCs) [6], Variable Length Error Correction (VLEC) codes [7], Even Weight Variable Length Codes (EWVLCs) [8] and for Irregular Variable Length Codes (IrVLCs) [9]. More specifically, when the cardinality of the symbol value set is infinite, the trellis and graph structures [10]–[17] employed by these codes become infinitely large, causing the corresponding decoding algorithms to become infinitely complex.

Against this background, this paper proposes a novel JSCC, which we refer to as the *Unary Error Correction (UEC) code*. As described in Section III, our UEC encoder generates a bit sequence by concatenating unary codewords [18], while the decoder employs a trellis that has only a modest complexity, even when the cardinality of the symbol value set is infinite. This trellis is designed so that the transitions between its states are synchronous with the transitions between the consecutive unary codewords in the concatenated bit sequence. This allows the UEC decoder to exploit the residual redundancy using the classic Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [19], as described in Section IV. In Section V, we demonstrate that an iteratively-decoded serial concatenation [20] of the proposed UEC code and an Irregular Unity Rate Code (IrURC) [21], [22] is capable of near-capacity operation. More specifically, we formally prove that when the symbol values obey a geometric probability distribution, our approach eliminates all capacity loss, even when the UEC trellis has a very low complexity. Furthermore, we prove that in the case of arbitrary symbol value distributions, the capacity loss asymptotically approaches zero, as the complexity of the UEC trellis is increased. In fact, we show that the capacity loss closely approaches zero, even if only a modest trellis complexity is employed. In Section VI, we compare the proposed JSCC UEC code with a benchmarker, which employs an EG source code and a separate Convolutional Code (CC). Unlike the proposed JSCC UEC code, this Separate Source and Channel Code (SSCC) benchmarker is found to suffer from 1.11 dB of capacity loss in a particular practical scenario, where Quaternary Phase Shift Keying (QPSK) modulation is employed for transmission over an uncorrelated narrowband Rayleigh fading channel. Finally, we conclude that the proposed UEC code has wide applicability and offers an attractive JSCC scheme for practical low-complexity applications.

II. Symbol Value Sets Having an Infinite Cardinality

The schemes considered in this paper are designed to convey a vector $\mathbf{x} = [x_i]_{i=1}^{a}$ comprising a number of symbols. This symbol vector is obtained as the realization of

The associate editor coordinating the review of this paper and approving it for publication was NAME. Manuscript received DATE; revised DATE.

The authors are with Electronics and Computer Science, University of Southampton, Hampshire, SO17 1BJ, UK (e-mail: {rm, wz4g11, tw08r, lh}@ecs.soton.ac.uk).

The financial support of the EPSRC under the grant EP/J015520/1, that of the RCUK under the India-UK Advanced Technology Centre (IU-ARC), as well as that of the EU under the CONCERTO project is gratefully acknowledged.

Digital Object Identifier 10.1109/TCOMM.2013.09.120460

[1]Note that the EG code can be considered to be a special case of the Exponential-Golomb code family, since it employs the same codewords as the $k = 0$ Exponential-Golomb code.

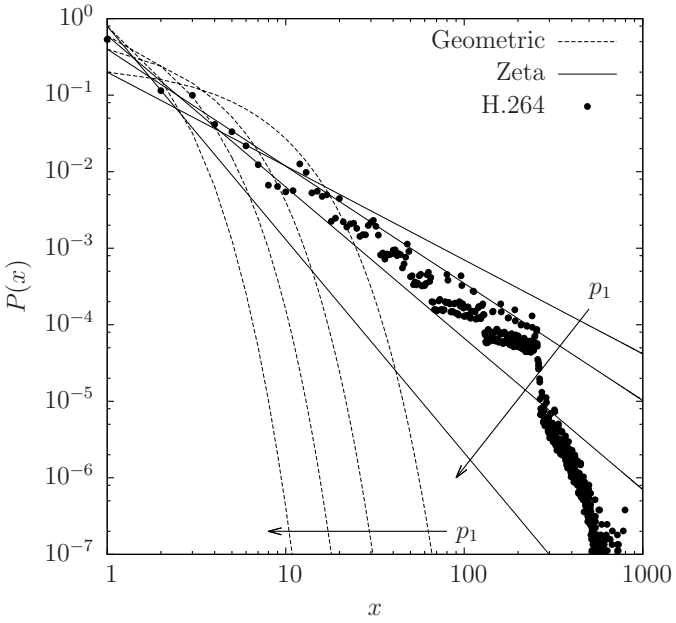


Fig. 1. The geometric and zeta probability distributions for $p_1 \in \{0.2, 0.4, 0.6, 0.8\}$, as well as the H.264 distribution, where $p_1 = 0.536$. This was obtained by recording the values of the 44.6 million symbols that are EG encoded when the JM 18.2 H.264 video encoder employs the 'encoder_baseline.cfg' configuration to encode the 175 s of video that are comprised by 4:2:0 versions of the Video Quality Expert Group (VQEG) test sequences.

a corresponding vector $\mathbf{X} = [X_i]_{i=1}^{a}$ of Independent and Identically Distributed (IID) Random Variables (RVs). Each RV X_i adopts the symbol value $x \in \mathbb{N}_1$ with probability $\Pr(X_i = x) = P(x)$, where $\mathbb{N}_1 = \{1, 2, 3, \ldots\}$ is the infinite-cardinality set comprising all positive integers. Here, the symbol entropy is given by $H_X = \sum_{x \in \mathbb{N}_1} H[P(x)]$, where $H[p] = p \log_2(1/p)$.

For example, Figure 1 depicts the geometric distribution [23], which is defined as

$$P(x) = p_1(1 - p_1)^{x-1}, \tag{1}$$

where $p_1 = \Pr(X_i = 1)$ and the symbol entropy is given by

$$H_X = \frac{H[p_1] + H[1 - p_1]}{p_1}. \tag{2}$$

By contrast, the zeta distribution [23] of Figure 1 is defined as

$$P(x) = \frac{x^{-s}}{\zeta(s)}, \tag{3}$$

where $\zeta(s) = \sum_{x \in \mathbb{N}_1} x^{-s}$ is the Riemann zeta function[2] and $s > 1$. In this case, $p_1 = 1/\zeta(s)$ and the symbol entropy is given by

$$H_X = \frac{\ln(\zeta(s))}{\ln(2)} - \frac{s\zeta'(s)}{\ln(2)\zeta(s)}, \tag{4}$$

where $\zeta'(s) = -\sum_{x \in \mathbb{N}_1} \ln(x) x^{-s}$ is the derivative of the Riemann zeta function.

[2]Expressing the zeta distribution as $P(x) = x^{-s} / \sum_{x \in \mathbb{N}_1} x^{-s}$ offers an attractive parallel with the geometric distribution, which may also be expressed in the form $P(x) = s^{-x} / \sum_{x \in \mathbb{N}_1} s^{-x}$, giving $p_1 = (s-1)/s$.

TABLE I
THE FIRST TWELVE CODEWORDS OF THE UNARY AND EG CODES.

x_i	$\mathbf{y}_i$	
	Unary	EG
1	0	1
2	10	010
3	110	011
4	1110	00100
5	11110	00101
6	111110	00110
7	1111110	00111
8	11111110	0001000
9	111111110	0001001
10	1111111110	0001010
11	11111111110	0001011
12	111111111110	0001100

Finally, Figure 1 also exemplifies the distribution of the symbol values that are EG encoded by the H.264 video encoder, corresponding to a symbol entropy of $H_X = 2.980$ bits per symbol. Note that these symbol values appear to obey Zipf's law [23], since their distribution may be approximated by the zeta distribution.

III. UEC ENCODER

Figure 2(a) illustrates the proposed UEC encoder, which performs the JSCC encoding of symbol values that are selected from a set having an infinite cardinality.

A. Unary encoder

The unary encoder [18] of Figure 2(a) represents each symbol x_i in the vector $\mathbf{x}$ using the corresponding x_i-bit unary codeword $\mathbf{y}_i$ of Table I. These codewords are then concatenated to obtain the b-bit vector $\mathbf{y} = [y_j]_{j=1}^{b}$ shown in Figure 2(a). For example, the vector $\mathbf{x} = [2, 1, 4, 2, 1, 3, 1, 1]$ of $a = 8$ symbols yields the $b = 15$-bit vector $\mathbf{y} = 100111010011000$.

The bit vector length b can be modeled as the realization of a RV B. This has an average value al that is finite, provided that the unary codewords have a finite average length

$$l = \sum_{x \in \mathbb{N}_1} P(x)x. \tag{5}$$

This is guaranteed [18] when the symbol values obey the geometric probability distribution of (1), in which case we have

$$l = \frac{1}{p_1}. \tag{6}$$

By contrast, when the symbols adopt the zeta distribution of (3), the average unary codeword length is only finite for $s > 2$ and hence for $p_1 > 0.608$, in which case we have

$$l = \zeta(s-1)/\zeta(s). \tag{7}$$

Note that for the H.264 symbol value distribution of Figure 1, we obtain the finite average unary codeword length of $l = 7.191$ bits per symbol.

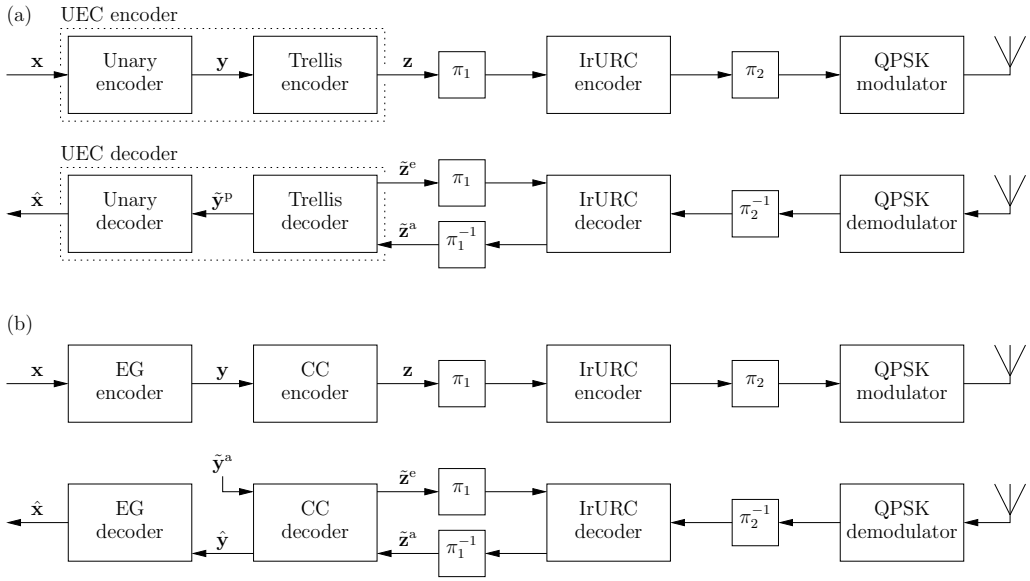


Fig. 2. Schematics of (a) the UEC JSCC scheme and (b) the EG-CC SSCC scheme, when serially concatenated with IrURC coding and Gray-coded QPSK modulation schemes. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers. Puncturing and depuncturing may also be performed by π_2 and π_2^{-1}, respectively.

B. Trellis encoder

As shown in Figure 2(a), the bit vector of concatenated unary codewords $\mathbf{y}$ is forwarded to a trellis encoder. This employs a UEC trellis of the sort depicted in Figure 3(a) to encode the value of each bit y_j in the vector $\mathbf{y}$, in order of increasing bit-index j. Each bit forces the trellis encoder to transition from its particular previous state $m_{j-1} \in \{1, 2, \ldots, r\}$ into a new state $m_j \in \{1, 2, \ldots, r\}$ that is selected from two legitimate alternatives, depending on the bit value y_j. More specifically,

$$m_j = \begin{cases} 1 + \mathrm{odd}(m_{j-1}) & \text{if } y_j = 0 \\ \min[m_{j-1} + 2, r - \mathrm{odd}(m_{j-1})] & \text{if } y_j = 1 \end{cases},$$

where the number of possible states r is required to be even and the encoding process begins from the state $m_0 = 1$, with the function $\mathrm{odd}(\cdot)$ yielding 1 if the operand is odd or 0 if it is even. In this way, the bit vector $\mathbf{y}$ identifies a path through the trellis, which may be represented by a vector $\mathbf{m} = [m_j]_{j=0}^{b}$ comprising $(b+1)$ state values. For example, the bit vector $\mathbf{y} = 100111010011000$ corresponds to the path $\mathbf{m} = [1, 3, 2, 1, 3, 3, 3, 2, 4, 1, 2, 4, 4, 1, 2, 1]$ through the $r = 4$-state trellis of Figure 3(b). Note that the UEC trellis is designed to ensure that the transitions between its states 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. A symbol having the value of $x_i < r/2$ and an odd index i results in a path section comprising the sequence of states $\mathbf{m}_i = [1, 3, 5, \ldots, 2x_i - 1, 2]$, while $\mathbf{m}_i = [2, 4, 6, \ldots, 2x_i, 1]$ is resulted when i is even. By contrast, for symbols having the value of $x_i \geq r/2$, an odd index i results in the sequence of x_i transitions $\mathbf{m}_i = [1, 3, 5, \ldots, r-1, r-1, r-1, \ldots, r-1, 2]$, while $\mathbf{m}_i = [2, 4, 6, \ldots, r, r, r, \ldots, r, 1]$ is yielded when i is even. Note that a path section $\mathbf{m}_i$ having an odd index i will always start in state 1 and end in state 2, while sections having an even index i start in state 2 and end in state 1. Owing to this, the trellis 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.

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, which are associated with the transition probabilities $\Pr(M_j = m, M_{j-1} = m') = P(m, m')$ of (8), as derived in the appendix. The conditional transition probabilities $\Pr(M_j = m | M_{j-1} = m') = P(m|m')$ are given by [24]

$$P(m|m') = \frac{P(m, m')}{\sum_{\check{m}=1}^{r} P(\check{m}, m')}, \tag{9}$$

while the entropy of the RV vector $\mathbf{M}$ is given by

$$H_{\mathbf{M}} = b \sum_{m'=1}^{r} \sum_{m=1}^{r} P(m, m') \log_2 \left(\frac{1}{P(m|m')} \right). \tag{10}$$

The trellis encoder represents each bit y_j in the vector $\mathbf{y}$ by an n-bit codeword $\mathbf{z}_j$. This codeword is selected from the set of $r/2$ codewords $\mathbb{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r/2-1}, \mathbf{c}_{r/2}\}$ or from the complementary set $\overline{\mathbb{C}} = \{\overline{\mathbf{c}_1}, \overline{\mathbf{c}_2}, \ldots, \overline{\mathbf{c}_{r/2-1}}, \overline{\mathbf{c}_{r/2}}\}$. As shown in Figure 3(a), this is achieved according to

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

Finally, the selected codewords are concatenated to obtain the bn-bit vector $\mathbf{z} = [z_k]_{k=1}^{bn}$ of Figure 2(a). For example, the bit vector $\mathbf{z} = 10111010000011010010111000110$ is obtained, when the $n = 2$-bit trellis of Figure 3(b) is employed to encode the bit vector $\mathbf{y} = 100111010011000$.

The bit vector $\mathbf{z}$ may be modeled as a specific realization of a vector $\mathbf{Z} = [Z_k]_{k=1}^{bn}$ comprising bn binary RVs. Furthermore, the UEC trellis of Figure 3(a) has been designed to obey

$$P(m,m') = \begin{cases} \frac{1}{2l}\left[1 - \sum_{x=1}^{\left\lceil \frac{m'}{2} \right\rceil} P(x)\right] & \text{if } m' \in \{1,2,3,\ldots,r-2\},\ m = m'+2 \\ \frac{1}{2l} P(x)\Big|_{x=\left\lceil \frac{m'}{2} \right\rceil} & \text{if } m' \in \{1,2,3,\ldots,r-2\},\ m = 1+\mathrm{odd}(m') \\ \frac{1}{2l}\left[1 - \sum_{x=1}^{\frac{r}{2}-1} P(x)\right] & \text{if } m' \in \{r-1,r\},\ m = 1+\mathrm{odd}(m') \\ \frac{1}{2l}\left[l - \frac{r}{2} - \sum_{x=1}^{\frac{r}{2}-1} P(x)\left(x - \frac{r}{2}\right)\right] & \text{if } m' \in \{r-1,r\},\ m = m' \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

symmetry and to rely on complementary codewords, so that it produces mostly equiprobable bits, where $\Pr(Z_k = 0) = \Pr(Z_k = 1)$ and the bit entropy is $H_Z = 1$. Note that owing to the edge effect, the binary RVs near either end of the vector $\mathbf{Z}$ do not adopt equiprobable values in general, in contrast to those in the middle of the vector. In practice however, this is only apparent for bits that are within a few positions from the ends of the vector $\mathbf{Z}$. As a result, the edge effect is negligible for practical values of the bit vector length bn and will be disregarded throughout the remainder of this paper.

The average length of the bit vector $\mathbf{z}$ is aln and the average coding rate of the UEC encoder is given by

$$R = \frac{H_X}{ln} = \frac{1}{ln} \sum_{x \in \mathbb{N}_1} H[P(x)]. \tag{11}$$

In the case where the RVs in the vector $\mathbf{X}$ obey a geometric distribution, we can substitute (1) and (6) into (11) in order to obtain [24]

$$R = \frac{H[p_1] + H[1-p_1]}{n}. \tag{12}$$

Here, the product of the UEC coding rate R and the codeword length n is related to the geometric distribution parameter p_1, as shown in Figure 4. Note that this relationship is symmetric about $p_1 = 0.5$, owing to the symmetry in (12). Figure 5 provides the corresponding plot for the case where the symbol values obey the zeta distribution of (3). Here, the coding rate R is zero for $p_1 \leq 0.608$, since the average unary codeword length l becomes infinite in this case. In the case of the H.264 symbol value distribution of Figure 1, we obtain a non-zero product having a value of $Rn = 0.414$.

C. Interleaver, IrURC encoder, puncturer and modulator

Following UEC encoding, the bit vector $\mathbf{z}$ may be interleaved by π_1 of Figure 2(a) and encoded by the IrURC [21], [22]. This encodes different portions of the interleaved bit vector using component Unity Rate Codes (URCs) having different generator polynomials. Note that the IrURC coding rate is given by the value of H_Z. If the UEC was not designed to produce equiprobable bits, then H_Z and this coding rate would be less than unity, introducing capacity loss [25]. Following IrURC encoding, puncturing π_2 is applied in order to achieve the desired overall coding rate before employing Gray-coded QPSK modulation for transmission, as shown in Figure 2(a). Alternatively, a mapping scheme other than Gray coding or a modulation scheme having a higher order can be employed, although this may require a higher complexity receiver design, as discussed in Section IV-C.

IV. UEC Decoder

In the receiver, demodulation, depuncturing π_2^{-1}, IrURC decoding and deinterleaving π_1^{-1} may be performed before invoking the proposed UEC decoder of Figure 2(a).

A. Trellis decoder

The trellis decoder of Figure 2(a) is provided with a vector of *a priori* Logarithmic Likelihood Ratios (LLRs) $\tilde{\mathbf{z}}^{\mathrm{a}} = [\tilde{z}_k^{\mathrm{a}}]_{k=1}^{bn}$ that pertain to the corresponding bits in the vector $\mathbf{z}$. The trellis decoder applies the BCJR algorithm [19] to a UEC trellis of the sort shown in Figure 3(a), in order to consider every legitimate realization of $\mathbf{Z}$ having the particular length bn. Here, the value of bn is assumed to be perfectly known to the receiver and may be reliably conveyed by the transmitter using a small amount of side information in practice. The synchronization between the UEC trellis and the unary codewords is exploited during the BCJR algorithm's γ_t calculation of [19, Equation (9)], by employing the conditional transition probabilities $P(m|m')$ of (9). Note that the UEC trellis should be terminated at $m_0 = 1$ and either at $m_b = 1$ or $m_b = 2$, depending on whether the length a of the symbol vector $\mathbf{x}$ is even or odd, respectively. As shown in Figure 2(a), the BCJR decoder generates the vector of extrinsic LLRs $\tilde{\mathbf{z}}^{\mathrm{e}} = [\tilde{z}_k^{\mathrm{e}}]_{k=1}^{bn}$. Note that this BCJR algorithm has only a modest complexity, since the trellis of Figure 3 may comprise as few as two states and because exact knowledge of the symbol probability distribution $P(x)$ is not required. Rather, the only knowledge required is that of the average unary codeword length l and the probabilities of the first $r/2-1$ symbol values, which may be accurately estimated heuristically, when $P(x)$ is unknown.

B. EXtrinsic Information Transfer (EXIT) curve

The transformation of $\tilde{\mathbf{z}}^{\mathrm{a}}$ into $\tilde{\mathbf{z}}^{\mathrm{e}}$ may be characterized by plotting the inverted UEC EXIT curve in an EXIT chart [26], as exemplified in Figure 6. Note that if codewords comprising at least $n = 2$ bits are employed, then the free distance d_{free} of the UEC code will be at least two, and its EXIT curve will reach the $(1,1)$ point in the top right corner of the EXIT chart [27].

In the case where the *a priori* LLR vector $\tilde{\mathbf{z}}^{\mathrm{a}}$ is modeled by the transmission of $\mathbf{z}$ over a BEC, the EXIT chart area A that is situated below the inverted UEC EXIT curve is given by

$$A = \frac{1}{n} \sum_{m'=1}^{r} \sum_{m=1}^{r} P(m,m') \log_2\left(\frac{1}{P(m|m')}\right). \tag{13}$$

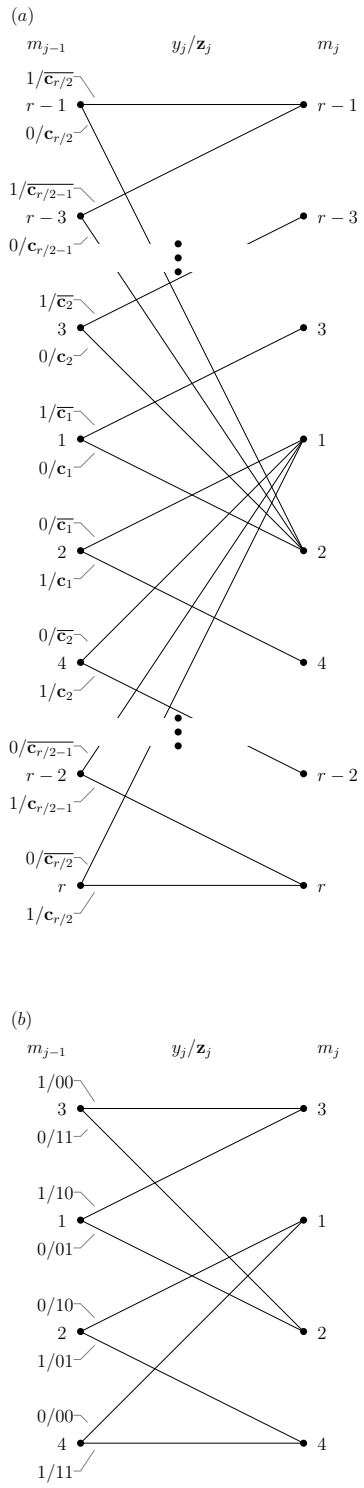


Fig. 3. (a) The generalized UEC trellis, having r states and n-bit codewords, where $\mathbb{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{r/2-1}, \mathbf{c}_{r/2}\}$. (b) An $r = 4$-state $n = 2$-bit UEC trellis, where $\mathbb{C} = \{01, 11\}$.

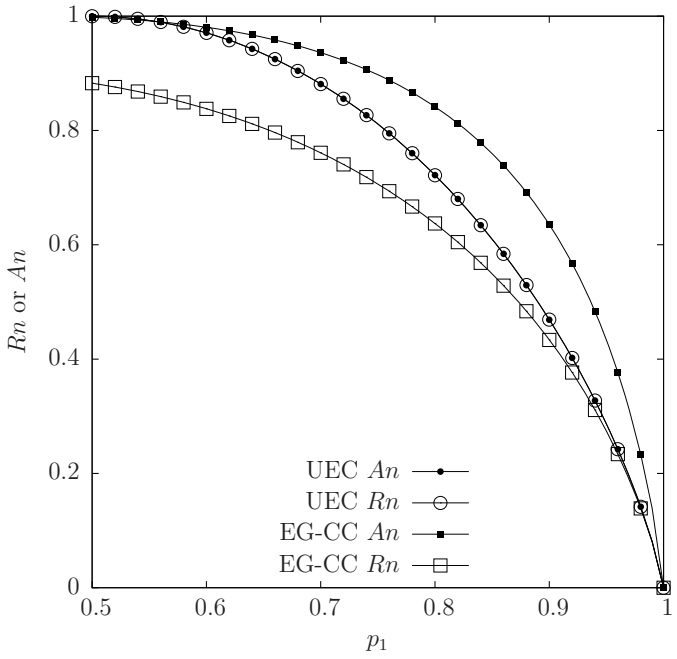


Fig. 4. Plots of Rn and An that are obtained for the UEC and EG-CC schemes, in the case where the symbol values obey a geometric distribution having the parameter p_1.

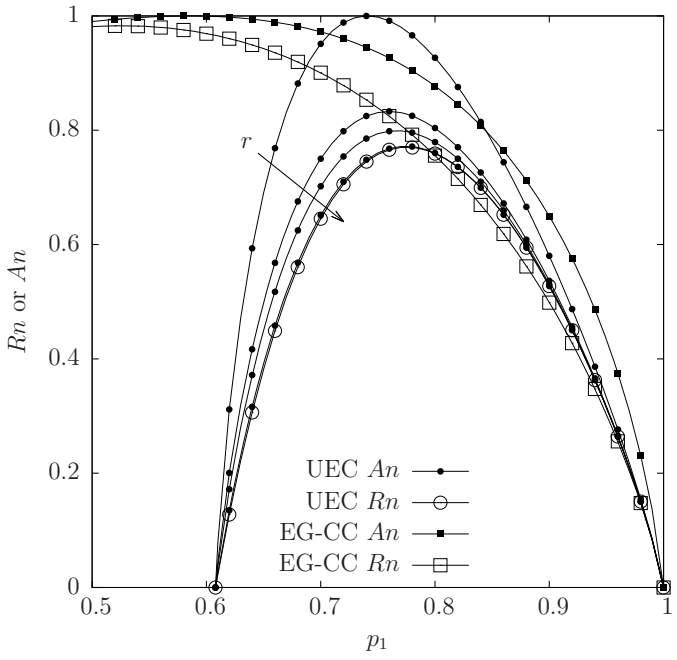


Fig. 5. Plots of Rn and An that are obtained for the UEC and EG-CC schemes, in the case where the symbol values obey a zeta distribution having the parameter p_1. The value of An is provided for UEC codes having various numbers of states $r \in \{2, 4, 6, 30\}$.

This result may be obtained from [25, Equation (23)], where the notation may be converted according to $\mathcal{A} = 1 - A$, since $\mathcal{A}$ is defined as the area *above* the inverted EXIT curve. Furthermore, we employ $I^2_{A,\max} = 1$ for the total area enclosed within the EXIT chart, while the sum of the entropies of the RVs in the vector $\mathbf{Z}$ is given by $\sum_{i=1}^{m} H(V_i) = bnH_Z$. Furthermore, we employ $H(\underline{V}|\underline{Y}) = H(\underline{V})$ as in [25, Equation (27)], since the UEC decoder is employed as an outer decoder, which has no access to channel information. Finally, we employ $H(\underline{V}) = H_{\mathbf{M}}$, which is given in (10). This is justified, since the proposed trellis decoder is an *A Posteriori* Probability (APP) decoder for a bit vector $\mathbf{z}$ that may be accurately modeled by the statistics of the trellis path $\mathbf{M}$. Note that the EXIT chart of Figure 6 and area calculation of (13) offer good approximations for cases where the LLRs

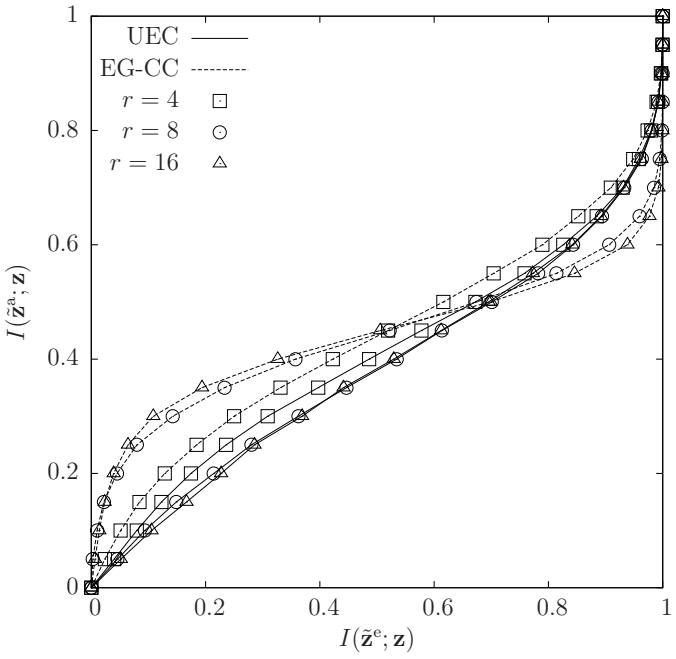


Fig. 6. Inverted EXIT curves of the r-state BCJR decoders employed in the UEC and EG-CC schemes, for the case where the *a priori* LLR vector $\tilde{\mathbf{z}}^{\mathrm{a}}$ is modeled by the transmission of $\mathbf{z}$ over a BEC and the symbol values obey a zeta probability distribution having the parameter value $p_1 = 0.797$. In the UEC scheme, the first codeword in the set $\mathbb{C}$ has the value 01 and the remaining $r/2 - 1$ codewords have the value 11, giving a free distance of $d_{\mathrm{free}} = 4 \; \forall \; r \in \{4, 6, 8, \ldots\}$. The EG-CC scheme employs the $n = 2$-bit CCs of Table II.

of $\tilde{\mathbf{z}}^{\mathrm{a}}$ obey distributions other than that modeled by a BEC [25]. Substituting (8) and (9) into (13) and then rearranging the result yields [24]

$$\begin{aligned} A &= \tfrac{1}{ln} \textstyle\sum_{x=1}^{\frac{r}{2}-1} H[P(x)] + \tfrac{2}{ln} H\left[1 - \sum_{x=1}^{\frac{r}{2}-1} P(x)\right] \\ &+ \tfrac{1}{ln} H\left[l - \tfrac{r}{2} - \textstyle\sum_{x=1}^{r/2-1} P(x)\left(x - \tfrac{r}{2}\right)\right] \\ &- \tfrac{1}{ln} H\left[1 + l - \tfrac{r}{2} - \textstyle\sum_{x=1}^{r/2-1} P(x)\left(1 + x - \tfrac{r}{2}\right)\right]. \end{aligned} \tag{14}$$

In the case where the RVs in the vector $\mathbf{X}$ obey the geometric distribution of (1), the product of the UEC EXIT chart area A and the codeword length n is related to the distribution parameter p_1, as shown in Figure 4. Note that this relationship is symmetric about $p_1 = 0.5$, where $A(p_1) = A(1 - p_1)$. Figure 5 provides the corresponding plots for the case where the symbol values obey the zeta distribution of (3) and where various numbers of states r are employed.

C. Iterative decoding

The extrinsic LLR vector $\tilde{\mathbf{z}}^{\mathrm{e}}$ of Figure 2(a) may be iteratively exchanged with the serially concatenated IrURC decoder. In-turn, the IrURC decoder may also iteratively exchange extrinsic LLRs with the demodulator [28], in order to avoid capacity loss when a mapping scheme other than Gray coding or a higher-order modulation scheme is employed. 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 [29], iterative decoder convergence towards the Maximum Likelihood (ML) performance is facilitated [30]. At this point, the trellis decoder may invoke the BCJR algorithm for generating the vector of *a posteriori* LLRs $\tilde{\mathbf{y}}^{\mathrm{p}} = [\tilde{y}_j^{\mathrm{p}}]_{j=1}^{b}$ that pertain to the corresponding bits in the vector $\mathbf{y}$. The unary decoder of Figure 2(a) sorts the values in this LLR vector 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 a bits to zero and the value of all other bits to one. Here, the value of a is assumed to be perfectly known to the receiver and may be reliably conveyed by the transmitter using a small amount of side information in practice. Finally, the bit vector $\hat{\mathbf{y}}$ can be unary decoded in order to obtain the symbol vector $\hat{\mathbf{x}}$ of Figure 2(a), which is guaranteed to comprise a number of symbols.

V. Near-capacity Performance of UEC Codes

Near-capacity operation is achieved, when reliable communication can be maintained at transmission throughputs that approach the capacity of the channel. When the UEC code of Figure 2(a) is serially concatenated with an IrURC code, near-capacity operation is facilitated, provided that the area A beneath the UEC code's inverted EXIT curve is equal to its coding rate R [25]. In this case, near-capacity operation will be achieved if the IrURC decoder's EXIT curve has a shape closely matching with that of the UEC decoder, hence creating a narrow but still open EXIT chart tunnel and facilitating iterative decoding convergence towards the ML performance.

When the RVs in the vector $\mathbf{X}$ obey the geometric distribution of (1), the UEC EXIT area A is indeed equal to the UEC coding rate R, as shown in Figure 4. In fact, this is the case, regardless of both the specific number of states r employed and of the codeword length n. This may be shown by substituting (1) and (6) into (14) and then rearranging the result, yielding [24]

$$A = \frac{H[p_1] + H[1 - p_1]}{n}, \tag{15}$$

which is identical to the expression provided for the coding rate R in (12).

In the case of the zeta distribution of (3), Figures 5 and 6 suggest that the UEC EXIT area A asymptotically approaches the UEC coding rate R as the number of states r is increased. In fact, this is the case, regardless of which particular symbol value distribution is adopted by the RVs in the vector $\mathbf{X}$. This may be proved by observing that the last three terms in (14) tend to zero as r is increased, leaving only the first term of

$$\lim_{r \to \infty} A = \frac{1}{ln} \sum_{x \in \mathbb{N}_1} H[P(x)], \tag{16}$$

which is equal to the expression provided for the coding rate R in (11). Figure 7 plots the discrepancy between Rn and An as a function of the number of UEC states r for zeta distributions employing various values for the parameter p_1. Note that in all considered cases, the discrepancy becomes less than 10^{-2}, when at least $r = 30$ states are employed. Figure 7 also provides the corresponding plot for the case where the RVs in the vector $\mathbf{X}$ obey the H.264 distribution of Figure 1. In this case $r = 14$ states are required for reducing the discrepancy below 10^{-2}.

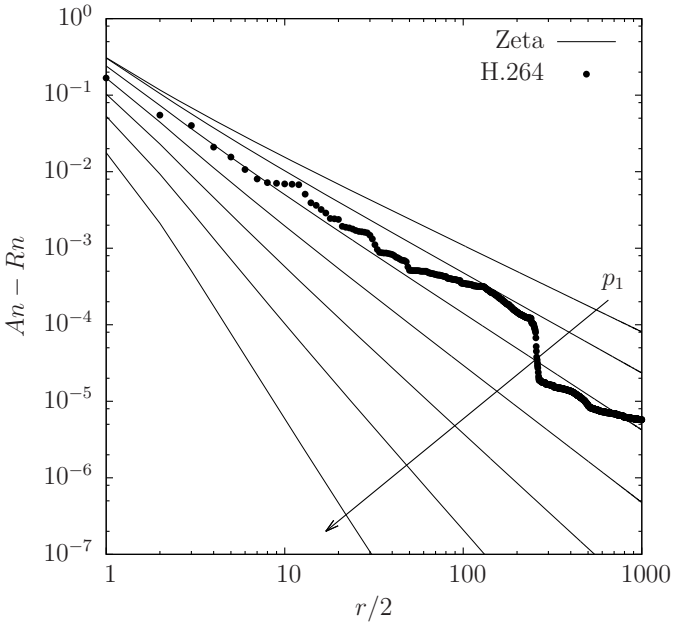


Fig. 7. The discrepancy between Rn and An that results when UEC codes having various numbers of states r are employed to encode symbol values having zeta distributions with the parameters $p_1 \in \{0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$, as well as symbol values having the H.264 distribution of Figure 1.

Note that the analysis of this section is specific to the case where the LLRs of $\tilde{\mathbf{z}}^{\mathrm{a}}$ adopt distributions that may be modeled by transmission over a BEC, since this is assumed by (14). However, the results of Section VI-C demonstrate that the proposed UEC code is also capacity-approaching in cases where the LLRs of $\tilde{\mathbf{z}}^{\mathrm{a}}$ obey other distributions, for example when communicating over an uncorrelated narrowband Rayleigh fading channel.

VI. Comparison with an SSCC Benchmarker

Figure 2(b) illustrates an EG-CC SSCC benchmarker, which can offer a fair comparison with the proposed UEC JSCC scheme. This is devised by replacing the components of the UEC code with the state-of-the-art SSCC components, on a like-for-like basis.

A. EG-CC encoder

In the transmitter, the unary encoder of the UEC scheme is replaced by an EG encoder. This employs the codewords shown in Table I, yielding the $b = 18$-bit vector $\mathbf{y} = 010100100010101111$, when the vector of $a = 8$ symbols $\mathbf{x} = [2, 1, 4, 2, 1, 3, 1, 1]$ is encoded, for example. The average EG codeword length is given by $l = \sum_{x \in \mathbb{N}_1} P(x) \left(2\lfloor \log_2(x) \rfloor + 1\right)$, which is guaranteed to be finite for any monotonic symbol value distribution having $P(x) \geq P(x+1)\ \forall\ x \in \mathbb{N}_1$, including the zeta distribution of (3) for all values of p_1. Even though the H.264 distribution of Figure 1 is not monotonic, a finite average EG codeword length of $l = 3.029$ bits per symbol is obtained. As in the proposed UEC scheme, the b-bit vector $\mathbf{y}$ may be modeled as a realization of a vector $\mathbf{Y} = [Y_j]_{j=1}^{b}$ comprising b binary RVs. However in the EG-CC scheme, these RVs do not adopt equiprobable values $\Pr(Y_j = 0) \neq \Pr(Y_j = 1)$ in the general case, giving a value of less than unity for the corresponding bit entropy $H_{Y_j} = H[\Pr(Y_j = 0)] + H[\Pr(Y_j = 1)]$.

TABLE II
THE OPTIMAL GENERATOR AND FEEDBACK POLYNOMIALS THAT SATISFY THE $H_Z = 1$ CONSTRAINT. POLYNOMIALS ARE PROVIDED IN THE FORMAT $(\mathbf{g}, f, d_{\mathrm{free}})$, WHERE $\mathbf{g}$ IS AN n-ELEMENT VECTOR OF OCTAL GENERATOR POLYNOMIALS, f IS THE OCTAL FEEDBACK POLYNOMIAL AND d_{free} IS THE DECIMAL FREE DISTANCE.

r	n		
	2	3	4
2	([2,2],3,2)	([2,2,2],3,3)	([2,2,2,2],3,4)
4	([4,7],6,4)	([4,7,7],6,6)	([4,7,7,7],6,8)
8	([15,17],16,6)	([13,15,17],16,10)	([13,15,15,17],16,13)
16	([27,31],34,7)	([25,33,37],36,12)	([25,33,35,37],32,16)

As shown in Figure 2, the trellis encoder of the UEC scheme is replaced by a $1/n$-rate r-state CC encoder. We recommend the CCs that are described by the generator and feedback polynomials provided in Table II. More specifically, we found that these non-systematic recursive CCs offer the optimal distance properties [31] subject to the constraint of producing equiprobable bits $\Pr(Z_k = 0) = \Pr(Z_k = 1)$. As described in Section III-B, this $H_Z = 1$ constraint is necessary to avoid capacity loss, when the EG-CC scheme is serially concatenated with an IrURC.

The average coding rate R of the EG-CC encoder is given by (11). When the RVs in the vector $\mathbf{X}$ obey the geometric distribution of (1), the product of the EG-CC coding rate R and the codeword length n is related to the distribution parameter p_1, as shown in Figure 4. Similarly, Figure 5 provides the corresponding plot for symbol values obeying the zeta distribution of (3). The product becomes $Rn = 0.983$ in the case of the H.264 symbol value distribution of Figure 1.

B. EG-CC decoder

In the receiver, the trellis decoder of the UEC scheme is replaced by a CC decoder, as shown in Figure 2(b). This employs the BCJR algorithm during the iterative decoding process, in order to convert the *a priori* LLR vector $\tilde{\mathbf{z}}^{\mathrm{a}}$ into the extrinsic LLR vector $\tilde{\mathbf{z}}^{\mathrm{e}}$. As shown in Figure 2(b), the BCJR algorithm can exploit some of the residual redundancy present within the bit vector $\mathbf{y}$ by employing a corresponding vector of *a priori* LLRs $\tilde{\mathbf{y}}^{\mathrm{a}} = [\tilde{y}_j^{\mathrm{a}}]_{j=1}^{b}$. Here, we have

$$\tilde{y}_j^{\mathrm{a}} = \ln\left(\frac{\Pr(Y_j = 0)}{\Pr(Y_j = 1)}\right),$$

where the bit value probabilities may be obtained heuristically.

As in the UEC scheme, the BCJR algorithm may be characterized by the corresponding inverted EG-CC EXIT curve, as exemplified in Figure 6. When the *a priori* LLR vector $\tilde{\mathbf{z}}^{\mathrm{a}}$ may be modeled by the transmission of $\mathbf{z}$ over a BEC, it can be shown that the EXIT chart area A that is situated below the inverted EG-CC EXIT curve is given by

$$A = \frac{\sum_{j=1}^{b} H_{Y_j}}{bn}.$$

Note that unlike in the UEC scheme, the EG-CC EXIT chart area A is independent of the number of states r employed in

the CC, as exemplified in Figure 6. If the RVs in the vector $\mathbf{X}$ obey the geometric distribution of (1), the product of the EG-CC EXIT chart area A and the codeword length n is related to the distribution parameter p_1, as shown in Figure 4. Similarly, Figure 5 provides the corresponding plots for symbol values obeying the zeta distribution of (3). The product is $An = 0.998$ in the case of the H.264 symbol value distribution of Figure 1.

Note that in the EG-CC scheme, the values of An and Rn are separated by significant discrepancies of up to about 0.2, preventing near-capacity operation, as discussed in Section V. This represents a significant disadvantage compared to the proposed UEC scheme, which can eliminate the discrepancy, when the symbol values obey the geometric distribution of (1). Note that if the EG code of the EG-CC scheme is replaced with a Golomb code, then the discrepancy between An and Rn is eliminated when the symbol values obey the geometric distribution of (1) [18]. However, a significant discrepancy can still be observed for other distributions. For these other symbol value distributions, the proposed UEC scheme can reduce the discrepancy to an infinitesimally small value by employing a sufficiently high number r of states, as discussed in Section V.

C. Symbol Error Ratio (SER) performance

As an example, consider the employment of the EG-CC and UEC schemes to encode symbol values that obey a zeta distribution having a parameter value of $p_1 = 0.797$, as characterized in Figures 6 and 8. This value offers a fair comparison since Figure 5 shows that $Rn = 0.762$ for both schemes, yielding the same effective throughput of $\log_2(M)R = 0.762$ bit/s/Hz, when $n = 2$-bit codewords and $M = 4$-ary QPSK are employed, for example. In an uncorrelated narrowband Rayleigh fading channel, the Discrete-input Continuous-output Memoryless Channel (DCMC) capacity exceeds this effective throughput when the E_b/N_0 value is above the 0.84 dB capacity bound. However, an open EXIT chart tunnel is prevented in the EG-CC scheme unless the DCMC capacity exceeds $\log_2(M)A = 0.882$ bit/s/Hz, which only occurs when the E_b/N_0 value is above 1.95 dB. This discrepancy indicates that the EG-CC scheme suffers from 1.11 dB of capacity loss in this case[3]. By contrast, employing as few as $r = 16$ states in the UEC scheme gives $\log_2(M)A = 0.765$ bit/s/Hz, which is exceeded by the DCMC capacity when the E_b/N_0 value is 0.87 dB. Therefore, the UEC scheme reduces the capacity loss to just 0.03 dB, and this may be further reduced to an infinitesimally small value by increasing the number of states r. This demonstrates that the proposed UEC code is capacity-approaching in cases where the LLRs of $\tilde{\mathbf{z}}^{\mathrm{a}}$ obey distributions other than that modeled by a BEC, such as when communicating over an uncorrelated narrowband Rayleigh fading channel.

Additionally, the EG-CC scheme suffers from another significant disadvantage compared to the proposed UEC scheme, once iterative decoding convergence has been achieved. In this event, the CC decoder of Figure 2(b) may generate the decoded bit vector $\hat{\mathbf{y}}$ by employing the Viterbi algorithm, which can also exploit the *a priori* LLR vector $\tilde{\mathbf{y}}^{\mathrm{a}}$. Following this, the EG decoder of Figure 2(b) extracts the symbol vector $\hat{\mathbf{x}}$ by interpreting $\hat{\mathbf{y}}$ as a concatenation of the EG codewords shown in Table I. However, if $\hat{\mathbf{y}}$ contains any bit errors, there is no guarantee that it will comprise the correct number a of codewords, or even that it will comprise an integer number of codewords. For example, the first 16 bits in the vector $\hat{\mathbf{y}} = 0101001000010101101$ correspond to a legitimate concatenation of six EG codewords, but the last two bits do not form a complete codeword. This disadvantage degrades the SER performance of the EG-CC scheme relative to that of the proposed UEC scheme, as exemplified in Figure 8 for the scenario described above, where the effective throughput is $\log_2(M)R = 0.762$ bit/s/Hz.

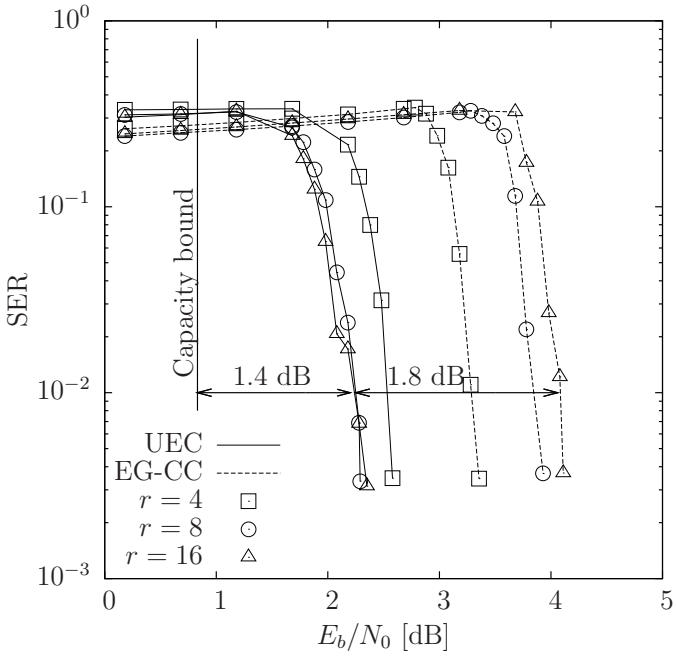


Fig. 8. The SER performance that is obtained following the achievement of iterative decoding convergence in the serially-concatenated UEC and EG-CC schemes of Figure 2, when transmitting frames comprising $a = 10^4$ symbols over an uncorrelated narrowband Rayleigh fading channel. For each scheme and for each value of r, a different IrURC parametrization was designed, in order to achieve the closest possible match with the corresponding EXIT curve of Figure 6.

As shown in Figure 8, there is a discrepancy between the 0.84 dB capacity bound and the E_b/N_0 values at which each of the schemes considered achieves an SER of 10^{-2}. In each case, this discrepancy represents the accumulation of three components, the first of which is the capacity loss that is quantified above. The second component of the discrepancy may be attributed to a constraint that was imposed upon the parametrization of the IrURC of Figure 2. More specifically, each IrURC parametrization was designed to achieve the closest possible match with the corresponding EXIT curve of Figure 6, under the constraint of only employing the ten component URCs of [21, Figure 4]. Since these component URCs employ no more than eight states, this constraint limits the IrURC complexity. However, this is achieved at the cost of increasing the E_b/N_0 threshold at which an open EXIT chart

[3]Note that an even greater capacity loss of 1.55 dB results when employing $M = 64$-ary Quadrature Amplitude Modulation (64QAM), which is a typical choice in multimedia transmission standards, such as DVB-T [32].

tunnel can be created, accounting for approximately 1 dB of the discrepancy with respect to the 0.84 dB capacity bound for the UEC scheme. The final component of the discrepancy may be attributed to the employment of a finite frame length a. Owing to this, the creation of a marginally open EXIT chart tunnel is insufficient for guaranteeing a low SER. Instead, the E_b/N_0 value must be increased, in order for the EXIT chart tunnel to become sufficiently wide for the iterative decoding trajectory to navigate through reliably [33].

As shown in Figure 8 and in contrast to the UEC scheme, the SER performance of the EG-CC scheme degrades as the number of CC states r is increased, even though this improves the CC's free distance d_{free}. As shown in Figure 6, these improved free distances are achieved at the cost of having EXIT curves with a more pronounced 'S'-shape, which cannot be closely matched by the IrURC EXIT curve, owing to the above-described constraint. As a result, higher and higher E_b/N_0 values are required for creating an open EXIT chart tunnel as r is increased, hence degrading the SER performance in the manner described. Owing to this, the proposed UEC scheme achieves an SER of 10^{-2} at an E_b/N_0 value that is 1.8 dB lower than that required by the EG-CC scheme, when $r = 16$ states are employed. This gain may be considered to be significant, since it exceeds the corresponding 1.4 dB discrepancy from the capacity bound, as shown in Figure 8.

VII. Conclusions

Multimedia codecs such as the H.264 video codec often employ source codes like the EG code, which are designed for encoding symbol values that are selected from a set having an infinite cardinality, such as the set of all positive integers. However, following the application of these source codes, some residual redundancy is typically retained, which imposes capacity loss and prevents near-capacity operation, as discussed in Section VI. This capacity loss was found to have a value of 1.11 dB in a particular practical scenario, where QPSK modulation is employed for transmission over an uncorrelated narrowband Rayleigh fading channel. Previously, JSCCs have been proposed for exploiting all the residual redundancy in order to avoid capacity loss. However, existing JSCCs are impractical for symbol values that are selected from a set having an infinite cardinality, since a distinct codeword is required for every legitimate symbol value. This motivates the novel UEC code proposed in this paper.

As discussed in Section III, the proposed UEC encoder employs the set of unary codewords and an r-state trellis, which are synchronous with each other. Owing to this, the UEC decoder is capable of employing the BCJR algorithm to exploit all residual redundancy, as discussed in Section IV. In Section V, we proved that when the symbol values obey the geometric distribution of (1), our approach eliminates all capacity loss, regardless of how many states r are employed in the trellis. Furthermore, we proved that in the case of arbitrary symbol value distributions, the capacity loss asymptotically approaches zero as the number of states r is increased. Indeed, we showed that the capacity loss closely approaches zero, even if only a modest number of states r is employed, demonstrating that the proposed UEC code can be employed practically in moderate-complexity applications.

Our future work will endeavor to develop an *Irregular* Unary Error Correction (IrUEC) code, which is comprised of multiple component UEC codes, having different codeword sets $\mathbb{C}$ and hence different EXIT curves. By carefully selecting the number of symbols in the sequence $\mathbf{x}$ that are encoded by each IrUEC component code, we will shape the irregular UEC EXIT curve so that it closely matches with that of a serially concatenated IrURC code. In this way, we will reduce the 1.4 dB discrepancy from the E_b/N_0 capacity bound of Figure 8 to a small fraction of a decibel, as in [21].

We will also aim to develop a *universal* error correction code, which operates upon the same principle as the proposed UEC code, but can be applied for all monotonic symbol value distributions. Although we expect that this could only be achieved at the cost of a greater complexity, it would overcome the limitation of the proposed UEC code, which cannot be applied when the average unary codeword length of (5) becomes infinite. This problem occurs when the symbol values obey the zeta distribution of (3) with a parameter value of $p_1 \leq 0.608$, for example. Despite this limitation, the presently proposed UEC offers an attractive solution for a wide variety of symbol value distributions, including the H.264 distribution of Figure 1.

Appendix
Derivation of (8)

The probability $P(m, m')$ of a transition from a state $m' \in \{1, 2, 3, \ldots, r-2\}$ to a state $m = m'+2$ can be derived by observing that a transition of this type will occur for each symbol in the vector $\mathbf{x}$ satisfying $x_i > \lceil m'/2 \rceil$ and simultaneously having an index that satisfies $\text{odd}(i) = \text{odd}(m')$, as shown in the path sections specified by $\mathbf{m}_i$ in Section III-B. The number of symbols in the vector $\mathbf{x}$ that satisfy these conditions has an expected value of

$$\begin{aligned} & \frac{a}{2} \sum_{x=\lceil \frac{m'}{2} \rceil + 1}^{\infty} P(x) \\ = \; & \frac{a}{2} \left[\sum_{x=1}^{\infty} P(x) - \sum_{x=1}^{\lceil \frac{m'}{2} \rceil} P(x) \right] \\ = \; & \frac{a}{2} \left[1 - \sum_{x=1}^{\lceil \frac{m'}{2} \rceil} P(x) \right]. \end{aligned}$$

Therefore we expect this number of the transitions in the path $\mathbf{m}$ to be of the above-mentioned type on average. Dividing this result by the expected number of transitions in the path $\mathbf{m}$, namely al, yields the transition probability given in (8) for the case where we have $m' \in \{1, 2, 3, \ldots, r-2\}$ and $m = m' + 2$.

Similarly, a transition from a state $m' \in \{1, 2, 3, \ldots, r-2\}$ to a state $m = 1 + \text{odd}(m')$ will occur for each symbol in the vector $\mathbf{x}$ satisfying $x_i = \lceil m'/2 \rceil$ and simultaneously having an index that satisfies $\text{odd}(i) = \text{odd}(m')$. We can expect $\frac{a}{2} P(x) \big|_{x=\lceil m'/2 \rceil}$ of the symbols in the vector $\mathbf{x}$ to satisfy these conditions and therefore we can expect this many of the transitions in the path $\mathbf{m}$ to be of the above-mentioned

type. Dividing this result by al yields the transition probability given in (8) for the case where we have $m' \in \{1, 2, 3, \ldots, r-2\}$ and $m = 1 + \mathrm{odd}(m')$.

Furthermore, a transition from a state $m' \in \{r-1, r\}$ to a state $m = 1 + \mathrm{odd}(m')$ will occur for each symbol in the vector $\mathbf{x}$ satisfying $x_i > r/2 - 1$ and simultaneously having an index that satisfies $\mathrm{odd}(i) = \mathrm{odd}(m')$. The number of transitions in the path $\mathbf{m}$ having the above-mentioned type has an expected value of $\frac{a}{2}\left[1 - \sum_{x=1}^{r/2-1} P(x)\right]$. Dividing this result by al yields the transition probability given in (8) for the case where we have $m' \in \{r-1, r\}$ and $m = 1 + \mathrm{odd}(m')$.

Finally, each symbol in the vector $\mathbf{x}$ satisfying $x_i > r/2 - 1$ will yield $(x_i - r/2)$ transitions from a state $m' \in \{r-1, r\}$ to a state $m = m'$, where $\mathrm{odd}(i) = \mathrm{odd}(m')$. Therefore, the number of transitions in the path $\mathbf{m}$ that can be expected to be of the above-mentioned type is given by

$$\begin{aligned} & \frac{a}{2} \sum_{x=\frac{r}{2}}^{\infty} P(x) \left(x - \frac{r}{2}\right) \\ = \; & \frac{a}{2} \left[\sum_{x=1}^{\infty} P(x) \left(x - \frac{r}{2}\right) - \sum_{x=1}^{\frac{r}{2}-1} P(x) \left(x - \frac{r}{2}\right) \right] \\ = \; & \frac{a}{2} \left[l - \frac{r}{2} - \sum_{x=1}^{\frac{r}{2}-1} P(x) \left(x - \frac{r}{2}\right) \right]. \end{aligned}$$

Dividing this result by al yields the transition probability given in (8) for the case where $m' \in \{r-1, r\}$ and $m = m'$.

ACKNOWLEDGMENT

L. Hanzo would also like to acknowledge the European Research Council's fiscal support under its Advanced Fellow Grant.

REFERENCES

[1] *Advanced video coding for generic audiovisual services*, ITU-T Std. H.264, Mar. 2005.

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

[3] K. Sayood and J. C. Borkenhagen, "Use of residual redundancy in the design of joint source/channel coders," *IEEE Trans. Commun.*, vol. 39, no. 6, pp. 838–846, June 1991.

[4] J. L. Massey, "Joint source and channel coding," in *Communication Systems and Random Process Theory*, J. K. Skwirzynski, Ed. Amsterdam, The Netherlands: Sijthoff and Noordhoff, Dec. 1978, pp. 279–293.

[5] B. L. Montgomery and J. Abrahams, "Synchronization of binary source codes," *IEEE Trans. Inf. Theory*, vol. 32, no. 6, pp. 849–854, Nov. 1986.

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

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

[8] R. Thobaben and J. Kliewer, "Design considerations for iteratively-decoded source-channel coding schemes," in *Proc. Allerton Conf. Commun., Control, Comput.*, Monticello, IL, USA, Sept. 2006.

[9] 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. [Online]. Available: http://eprints.ecs.soton.ac.uk/14470/

[10] V. B. Balakirsky, "Joint source-channel coding using variable-length codes," *Problems Inf. Transmission*, vol. 37, no. 1, pp. 10–23, Jan. 2001.

[11] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[12] K. Sayood, H. H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.

[13] R. Bauer and J. Hagenauer, "Symbol by symbol MAP decoding of variable length codes," in *Proc. ITG Conf. Source Channel Coding*, Munich, Germany, Jan. 2000, pp. 111–116.

[14] R. Thobaben and J. Kliewer, "Robust decoding of variable-length encoded Markov sources using a three-dimensional trellis," *IEEE Commun. Lett.*, vol. 7, no. 7, pp. 320–322, July 2003.

[15] C. Weidmann, "Reduced-complexity soft-in-soft-out decoding of variable-length codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Yokohama, Japan, June 2003, p. 201.

[16] S. Malinowski, H. Jegou, and C. Guillemot, "Synchronization recovery and state model reduction for soft decoding of variable length codes," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 368–377, Jan. 2007.

[17] R. G. Maunder, J. Kliewer, S. X. Ng, J. Wang, L.-L. Yang, and L. Hanzo, "Joint iterative decoding of trellis-based VQ and TCM," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1327–1336, Apr. 2007. [Online]. Available: http://eprints.ecs.soton.ac.uk/14077/

[18] R. G. Gallager and D. C. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Trans. Inf. Theory*, vol. 21, no. 2, pp. 228–230, Mar. 1975.

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

[20] J. Hagenauer and N. Görtz, "The turbo principle in joint source-channel coding," in *Proc. IEEE Inf. Theory Workshop*, Paris, France, Mar. 2003, pp. 275–278.

[21] 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. [Online]. Available: http://eprints.ecs.soton.ac.uk/14471/

[22] L. Hanzo, R. G. Maunder, J. Wang, and L.-L. Yang, *Near-Capacity Variable Length Coding*. Chichester, UK: Wiley, 2010. [Online]. Available: http://eprints.ecs.soton.ac.uk/20911/

[23] N. L. Johnson, A. W. Kemp, and S. Kotz, *Univariate Discrete Distributions*. New York, NY, USA: John Wiley & Sons, 2005.

[24] R. G. Maunder, W. Zhang, T. Wang, and L. Hanzo, "Derivations for 'A unary error correction code for the near-capacity joint source and channel coding of symbol values from an infinite set'." [Online]. Available: http://eprints.soton.ac.uk/341736/

[25] A. Ashikhmin, G. Kramer, and S. ten Brink, "Extrinsic information transfer functions: Model and erasure channel properties," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2657–2673, Nov. 2004.

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

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

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

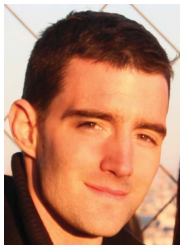
[29] R. G. Maunder and L. Hanzo, "Iterative decoding convergence and termination of serially concatenated codes," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 216–224, Jan. 2010. [Online]. Available: http://eprints.ecs.soton.ac.uk/16133/

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

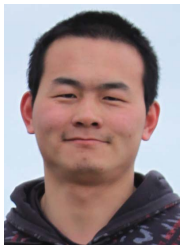
[31] P. Frenger, P. Orten, and T. Ottosson, "Convolutional codes with optimum distance spectrum," *IEEE Commun. Lett.*, vol. 3, no. 11, pp. 317–319, Nov. 1999.

[32] *Digital Video Broadcasting (DVB): Framing structure, channel coding and modulation for digital terrestrial television*, ETSI Std. EN 300 744.

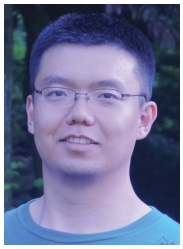
[33] J. W. Lee and R. E. Blahut, "Generalized EXIT chart and BER analysis of finite-length turbo codes," in *Proc. IEEE Global Telecommun. Conf.*, vol. 4, San Fransisco, CA, USA, Dec. 2003, pp. 2067–2072.



Robert G. Maunder (http://users.ecs.soton.ac.uk/rm) has studied with Electronics and Computer Science, University of Southampton, UK, since October 2000. He was awarded a first class honors BEng in Electronic Engineering in July 2003, as well as a PhD in Wireless Communications and a lectureship in December 2007. Rob's research interests include joint source/channel coding, iterative decoding, irregular coding and modulation techniques. He has published a number of IEEE papers in these areas.



Wenbo Zhang 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. He is currently working toward the Ph.D. degree with the Communications Research Group, Electronics and Computer Science, University of Southampton, Southampton, UK. His current research interests include joint source/channel coding and variable length coding.



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

Lajos Hanzo (http://www-mobile.ecs.soton.ac.uk) FREng, FIEEE, FIET, Fellow of EURASIP, DSc received his degree in electronics in 1976 and his doctorate in 1983. In 2009 he was awarded the honorary doctorate "Doctor Honoris Causa" by the Technical University of Budapest. During his 35-year career in telecommunications he has held various research and academic posts in Hungary, Germany and the UK. Since 1986 he has been with the School of Electronics and Computer Science, University of Southampton, UK, where he holds the chair in telecommunications. He has successfully supervised 80 PhD students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, coauthored 1300+ research entries at IEEE Xplore, acted both as TPC and General Chair of IEEE conferences, presented keynote lectures and has been awarded a number of distinctions. Currently he is directing a 100-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) UK, the European IST Programme and the Mobile Virtual Centre of Excellence (VCE), UK. He is an enthusiastic supporter of industrial and academic liaison and he offers a range of industrial courses. He is also a Governor of the IEEE VTS. During 2008 - 2012 he was the Editor-in-Chief of the IEEE Press and a Chaired Professor also at Tsinghua University, Beijing. His research is funded by the European Research Council's Senior Research Fellow Grant. For further information on research in progress and associated publications please refer to http://www-mobile.ecs.soton.ac.uk