Learning-aided Unary Error Correction Codes for Non-Stationary and Unknown Sources

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

***Abstract*—Unary Error Correction (UEC) codes have recently been proposed for the Joint Source and Channel Coding (JSCC) of symbol values that are selected from a set having an infinite cardinality. However, the original UEC scheme requires the knowledge of the source probability distribution, in order to achieve near-capacity operation. This limits the applicability of the UEC scheme, since the source probability distribution is typically non-stationary and is unknown in practice. In this paper, we propose a dynamic version of the UEC scheme, which can learn the unknown source statistics and gradually improve its decoding performance during a transient phase, then dynamically adapt to the non-stationary statistics and maintain reliable near-capacity operation during a steady-state phase, at the cost of only a moderate memory requirement at the decoder. Based on the same learning technique, we also propose two Separate Source and Channel Coding (SSCC) benchmarkers, namely a learning-aided Elias Gamma (EG)-Convolutional Code (CC) scheme and a learning-aided Arithmetic-CC scheme. The simulation results reveal that our proposed learning-aided UEC scheme outperforms the benchmarkers by up to 0.85 dB, without requiring any additional decoding complexity or any additional transmission-energy, -bandwidth or -duration.**

***Index Terms*—Source coding, Video coding, Channel coding, Channel capacity, Iterative decoding, Non-stationary source distribution, Unary Error Correction Codes**

NOMENCLATURE

3D	Three-Dimensional
ACS	Add, Compare and Select
APP	*A Posteriori* Probability
AWGN	Additive White Gaussian Noise
BCJR	Bahl-Cocke-Jelinek-Raviv
BICM	Bit-Interleaved Coded Modulation
CC	Convolutional Code
CRC	Cyclic Redundancy Check
DCMC	Discrete-input Continuous-output Memoryless Channel
EG	Elias Gamma
EGEC	Elias Gamma Error Correction
EWVLC	Even Weight Variable Length Code
EXIT	EXtrinsic Information Transfer
ExpG	Exponential Golomb
ExpGEC	Exponential Golomb Error Correction
FLC	Fixed length Code
HEVC	High Efficiency Video Coding
IID	Independent and Identically Distributed
IrCC	Irregular Convolutional Code
IrURC	Irregular Unity Rate Code
IrUEC	Irregular Unary Error Correction
IrVLC	Irregular Variable Length Code
LDPC	Low Density Parity Check
LLR	Logarithmic Likelihood Ratio
Log-MAP	Logarithmic Maximum *A Posteriori*
LUT	Look-Up-Table
LZ	Lempel-Ziv
JSCC	Joint Source and Channel Coding
MAP	Maximum A-Posteriori
MI	Mutual Information
ML	Maximum Likelihood
QPSK	Quaternary Phase Shift Keying
REG	Reordered Elias Gamma
REGEC	Reordered Elias Gamma Error Correction
RV	Random Variable
RVLC	Reversible Variable Length Code
SISO	Soft-Input Soft-Output
SER	Symbol Error Ratio
SNR	Signal to Noise Ratio
SSCC	Separate Source and Channel Coding
SSVLC	Self-Synchronizing Variable Length Code
SOVA	Soft-Output Viterbi Algorithm
TCM	Trellis-Coded Modulation
UEC	Unary Error Correction
UEP	Unequal Error Protection
URC	Unity Rate Code
VLEC	Variable Length Error Correction

LIST OF SYMBOLS

$\mathbf{x}$	The source symbol vector.
$\hat{\mathbf{x}}$	The reconstructed source symbol vector.
a	The length of source symbol vector.
$\Pr(\cdot)$	The probability of an arbitrary event occurring.
$\Pr(\cdot\|\cdot)$	The conditional probability.
$\Pr(\cdot,\cdot)$	The joint probability.
H_X	The symbol entropy.
H_Z	The bit entropy.
$\mathbb{N}_1$	The infinite-cardinality set comprising all positive integers.
$\zeta(s)$	The Riemann zeta function.
p_1	The probability of symbol value equals 1.
$\bar{p_1}$	The mean value of p_1.

The authors are with Electronics and Computer Science, University of Southampton, SO17 1BJ, United Kingdom, e-mail: {wz4g11,mfb2g09,tw08r,rm,lh}@ecs.soton.ac.uk

The financial support of the EPSRC, Swindon UK under the grants EP/J015520/1 and EP/L010550/1, that of the TSB Swindon UK under the grant TS/L009390/1, that of the RCUK under the India-UK Advanced Technology Centre (IU-ARC) as well as of the EU under the CONCERTO project and that of the European Research Council's Advanced Fellow grant is gratefully acknowledged. The research data for this paper is available at http://dx.doi.org/10.5258/SOTON/386973.

$\mathbf{p}$	The probabilities of the first $r/2-1$ symbol values.
$\hat{\mathbf{p}}$	The estimated value of $\mathbf{p}$.
l	The average codeword length for each symbol.
$\hat{l}$	The estimated value of l.
M	The storage memory employed in the learning-aided scheme.
m	The size of the storage memory M.
T	The number of successive symbol vectors produced per cycle.
σ	The standard deviation of the filtered Gaussian-distributed values.
π	The interleaving.
π^{-1}	The de-interleaving.
$E\{\cdot\}$	The expectation operation.
$\sum$	The summation of all elements.
$\forall$	For all elements within a certain range.
$\lim$	The limitation operation.
$\lceil\cdot\rceil$	Rounding a numerical value to its nearest higher integer.
$\lfloor\cdot\rfloor$	Rounding a numerical value to its nearest lower integer.
$\max(\cdot)$	The maximum value of a vector/matrix.
$\min(\cdot)$	The minimum value of a vector/matrix.
$mod(x,y)$	The remainder after the division of x by y.
$\overline{(\cdot)}$	The complementary operation.
$odd(\cdot)$	The function yields 1 if the operand is odd or 0 if it is even.

I. Introduction

In mobile wireless scenarios, multimedia transmission is required to be bandwidth efficient and resilient to transmission errors, motivating source coding for compression and channel coding for error correction [1]–[3]. According to Shannon's source-channel separation theorem [4], classic Separate Source and Channel Coding (SSCC) is able to achieve near-capacity communication by combining a near-entropy source code, such as an arithmetic code [5], with a near-capacity channel code, such as serially concatenated Convolutional Codes (CCs) [6]. In this way, it is theoretically possible to reconstruct the multimedia source information with an infinitesimally low probability of error, whenever the transmission throughput does not exceed the channel's capacity [4]. However, SSCC relies upon a number of assumptions that may not be valid in practice [7]. For example, near-entropy arithmetic coding requires both the transmitter and receiver to have accurate knowledge of the source probability distribution, which quantifies the occurrence probability of every value that is adopted by the source symbols. Indeed, in many practical applications, the source distribution is unknown at both the transmitter and the receiver, particularly if it is non-stationary, causing the symbol value probabilities to vary with time.

This motivates the design of universal codes, such as the Elias Gamma (EG) code [8] and its generalization, the Exponential Golomb (ExpG) code [9]. These codes facilitate the binary encoding of symbol values selected from infinite sets, without requiring any knowledge of the corresponding occurrence probabilities at either the transmitter or receiver. For example, the H.264 [10] and H.265 [11] video codecs employ the EG and ExpG codes, which may be serially concatenated with CCs to provide a separate error correction capability. Nevertheless, this SSCC approach typically suffers from capacity loss, owing to the residual redundancy that is typically retained during EG and ExpG encoding. This limits the level of compression that can be achieved, resulting in an average number of EG- or ExpG-encoded bits per symbol that exceeds the entropy of the symbols.

In order to exploit the residual redundancy that is retained in the classic SSCC approach and hence to achieve near-capacity operation, Joint Source and Channel Coding (JSCC) arrangements [12] have been proposed, such as Reversible Variable Length Codes (RVLCs) [13], Variable Length Error Correction (VLEC) codes [14] and Irregular Variable Length Codes (IrVLCs) [15]. However until recently, all previous JSCCs suffered from an excessive decoding complexity, when the cardinality of the symbol value set is large. In particular, the decoding complexity became infinite if the cardinality is infinite. As we have previously demonstrated in [16, Fig. 1], the symbols that are EG encoded in H.264 are approximately zeta distributed [17], resulting in some rare symbol values of around 1000, which are too high to be practically considered by previous JSCCs. This is also true of H.265 distributed symbols, as we shall show in this paper.

Against this background, we proposed a novel JSCC scheme referred to as the Unary Error Correction (UEC) code [16]. This was the first JSCC that mitigates the above-described capacity loss and incurs only a moderate decoding complexity, even when the cardinality of the symbol set is infinite. However, in our previous work, the UEC code was only able to achieve the near-capacity operation when the source distribution was known at the receiver. Hence, the applicability of the UEC code has been limited to some particular scenarios and has been prevented in the generalised case of unknown and non-stationary probability distributions. As depicted in Figure 1, there are a range of different aspects that have to be considered, when designing the UEC codes. In each of the papers [16], [18], [19], we focused on and made contributions to one or more of the aspects seen in Figure 1.

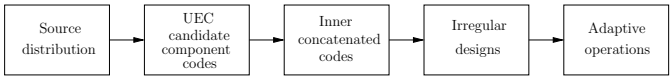


Fig. 1. The design-flow of a UEC coded scheme.

In [16], we introduced the encoding and decoding operations of the UEC code and characterized a serially concatenated scheme, namely the UEC-IrURC scheme proposed for facilitating practical near-capacity operation. The UEC encoder consists of two parts, namely the unary encoder and the trellis encoder. Owing to the synchronization between the unary codewords and the trellis transitions, the UEC decoder can invoke the BCJR algorithm to exploit all residual redundancy that remains following unary encoding, hence facilitating near-capacity operation at a moderate complexity. We also quantified the computational complexity of the UEC scheme in

order to strike a desirable trade-off between the contradictory requirements of low complexity and near-capacity operation. The EXIT chart and area properties of the UEC code were characterized and it was shown that in the case of arbitrary symbol value distributions, the capacity loss asymptotically approaches zero, as the complexity of the UEC trellis is increased. We showed that the SSCC EG-CC benchmarker suffers from capacity loss. Hence, our simulation results demonstrate that our UEC-IrURC scheme outperforms the EG-CC-IrURC benchmarker, offering a 1.3 dB gain and operating within 1.6 dB of the capacity bound.

When a UEC trellis encoder operates on the basis of an extended codebook, the corresponding trellis decoder can still employ the original codebook having a lower complexity. Based on this observation, we proposed dynamically reducing or increasing the number of states employed in the trellis decoder, in order to carefully balance the performance versus complexity trade-off. In [18], we proposed an *Adaptive* UEC-Turbo scheme, which is a three-stage concatenation applying an adaptive iterative decoding technique for expediting iterative decoding convergence. Three-Dimensional (3D) EXIT chart analysis was proposed for controlling the dynamic adaptation of the UEC trellis decoder, as well as for controlling the decoder activation order between the UEC decoder and the turbo decoder. More particularly, the 3D EXIT chart was employed for quantifying the benefit of activating each decoding component at each stage of the iterative decoding process. We showed that the UEC decoder's operation can be dynamically adjusted, and additionally its activation order controlling the iterative soft information exchange with the two turbo decoder components may also be varied. We also quantified the corresponding complexity cost in terms of the number of ACS operations performed by each decoding component. By activating the specific decoding component offering the largest benefit-to-cost ratio at each stage, we demonstrated that the convergence of the iterative decoding process may be significantly expedited, resulting in an attractive trade-off between its decoding complexity and its error correction capability.

Inspired by the irregular coding philosophy [15], in [19] we proposed an *Irregular* UEC-IrURC scheme, which facilitates nearer-capacity operation. The IrUEC scheme employs different UEC parametrizations for the encoding of different subsets of each message frame, operating on the basis of a single irregular trellis having a novel design. The irregular trellis employed by an IrUEC 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, nearer-to-capacity operation is facilitated by exploiting this fine-grained control of the IrUEC irregularity. The free-distance properties of the UEC trellis were characterised for the first time in [19], in order to conceive an attractive parametrization of the IrUEC scheme. Having characterized the free-distance of the UEC trellis using different codebooks, we carefully selected a suite of UEC codes having a wide variety of EXIT chart shapes for the component codes of our IrUEC code. A new double-sided EXIT chart matching algorithm was proposed for jointly matching the EXIT charts of the IrUEC and the IrURC codes. On the one hand, the component UEC codes having a wide variety of EXIT chart shapes provide design freedom for the IrUEC EXIT chart. On the other hand, our novel double-sided EXIT chart matching algorithm exploits this design freedom, in order to parametrize the IrUEC-IrURC scheme for creating a narrow but marginally open EXIT chart tunnel at a low $E_{\rm b}/N_0$ value that is close to the area bound and the capacity bound.

In this paper, we propose a new learning-aided UEC scheme. Like our previous UEC schemes, the proposed learning-aided scheme does not require prior knowledge of the source distribution at the transmitter. However, in contrast to our previous UEC schemes, the proposed learning-aided scheme also does not require any prior knowledge of the source distribution at the receiver. Instead, the proposed receiver can learn the source distribution based on the received symbols, enabling near-capacity operation. Starting from a position of having no prior information about the source distribution, the proposed receiver is able to recover a first frame of symbols, albeit possibly with a relatively high Symbol Error Ratio (SER) if the channel Signal to Noise Ratio (SNR) is close the capacity bound. Nonetheless, this frame of symbols can be used to make a first estimate of the source distribution, which is stored in memory. This information can then be used to aid the recovery of a second frame of symbols, with an improved SER. This allows the estimate of the source distribution that is stored in memory to be improved. In this way, the SER and the estimate of the source distribution can be gradually improved in successive frames, during a transient phase. Following this, the receiver enters a steady-state phase, during which reliable near-capacity operation can be maintained by continuing the learning process, even if the source is non-stationary. In this case, the memory is populated only with source distribution information derived from recently-received frames, replacing the out-of-date information from older frames. We will investigate the trade-off between error correction performance and the size of the memory storage and demonstrate that the amount of required memory is moderate and practical. Furthermore, we propose two learning-aided SSCC benchmarkers, namely a learning-aided EG-CC scheme and a learning-aided Arithmetic-CC scheme. The EG-CC scheme employs the proposed learning technique only in the receiver, but suffers from capacity loss. The arithmetic scheme employs the learning technique in the transmitter based on the transmitted symbols and in the receiver based on the received symbols, but suffers from desynchronization in the presence of transmission errors.

The rest of the paper is organised as shown in Figure 2. In Section II, we review the history of SSCC and JSCC schemes, as well as the motivation of the UEC scheme. In Section III, we analyse the nature of source symbols having both stationary and non-stationary probability distributions. Section IV details the transmitter and receiver of the proposed learning-aided UEC scheme, while Section V describes the learning technique advocated. Section VI details our SSCC benchmarkers and compares their SER performance to that

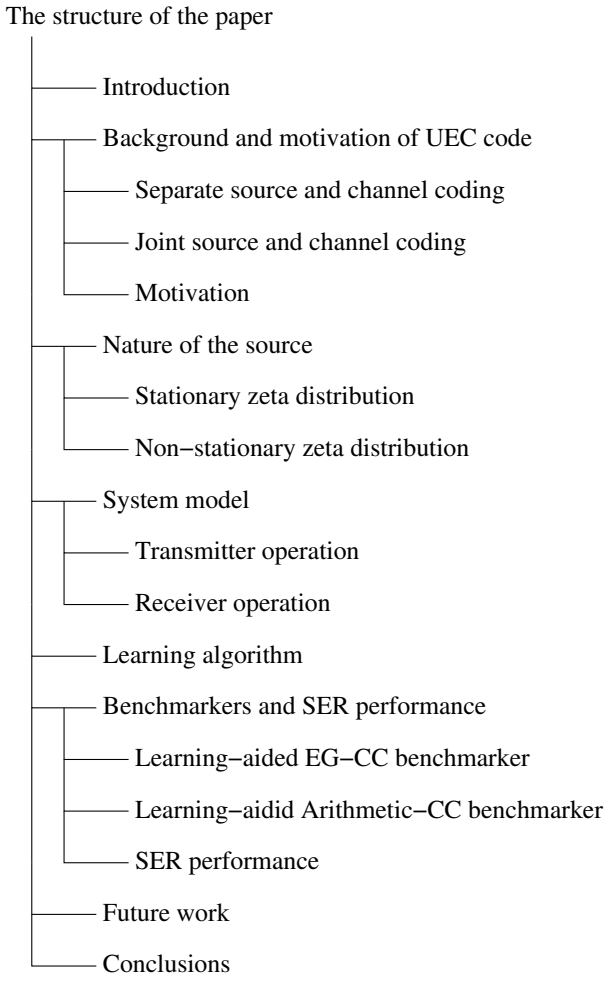


Fig. 2. The structure of the paper.

of the proposed learning-aided UEC scheme. Our simulation results demonstrate that our proposed learning-aided UEC scheme outperforms the benchmarkers by up to 0.85 dB, without requiring any additional decoding complexity or additional transmission-energy, -bandwidth or -duration. In Section VII, we provide a number of potential future research ideas based on the technique developed in this paper and our previous contributions. Finally, Section VIII concludes the paper.

II. Background and motivation of UEC code

Our UEC scheme is a JSCC [12] scheme conceived for performing both compression and error correction of multimedia information during its transmission from an encoder to a decoder. In this section, we review the background of the SSCC and JSCC schemes, as well as the motivation os the UEC code.

A. Separate Source and Channel Coding

In Shannon's seminal contribution [4], his source and channel coding separation theorem stated that source coding designed for compression and channel coding invoked for error correction can be designed entirely independently, without any loss of performance. In general, a basic SSCC scheme in wireless digital communications may be represented by Figure 3. When communicating over a perfectly noiseless wireless communications channel, Shannon [4] showed that the minimum number of bits required after compression using source coding to reliably convey perfect knowledge of the source signal's information content to the receiver is given by the source entropy, because source coding eliminates the redundancy inherent in the source information. When communicating over noisy channels, Shannon [4] showed that if a source signal's information content is conveyed at a rate (bits per second) that does not exceed the channel's capacity, then it is theoretically possible to reconstruct it with an infinitesimally low probability of error. This motivates the employment of separate channel coding, which introduces carefully controlled redundancy that can be beneficially exploited for error correction. With this SSCC guidelines, the efforts of the academic and industrial research communities expended over the past 60 years are summarized at a glance in Table I.

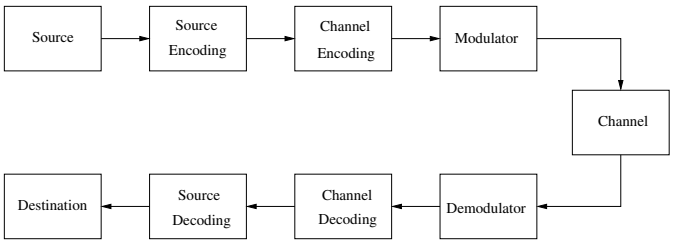


Fig. 3. A digital communications system relying on Separate Source and Channel Coding (SSCC).

However, Shannon's findings are only valid under a number of idealistic assumptions [20], namely that the information is transmitted over an uncorrelated non-dispersive narrowband Additive White Gaussian Noise (AWGN) channel, while potentially imposing an infinite decoding complexity and buffering latency. These assumptions clearly have limited validity for practical finite-delay transmissions over realistic fading wireless channels [21]. Additionally, Shannon assumed that the source is stationary and that it is losslessly encoded. These assumptions have a limited validity in the case of multimedia transmission, since video, image, audio information is typically non-stationary, having characteristics that vary in time and/or space [22], [23]. Furthermore, 'lossy' compression [24] is often readily tolerated for multimedia information, since human observers can typically tolerate moderate signal degradation in exchange for requiring a reduced bandwidth. Owing to this, some residual redundancy is typically retained during source coding, hence preventing near-capacity operation. This observation motivated the conception of Joint Source and Channel Coding (JSCC) [12], as it will be introduced in Section II-B.

B. Joint Source and Channel Coding

In order to exploit the residual redundancy and hence to achieve near-capacity operation, the classic SSCC schemes

TABLE I
MILESTONES IN SEPARATE SOURCE AND CHANNEL CODING (SSCC).

Year	Author(s)	Contribution
1948	Shannon [4]	Information theory and channel capacity
1949	Fano [25]	Discrete message transmission in noiseless systems
1950	Hamming [26]	Hamming code was proposed
1952	Huffman [27]	Huffman code was proposed
1954	Reed [28]	Reed-Muller (RM) code was introduced
1955	Elias [6]	CC was conceived
1957	Wozencraft [29]	Sequential decoding
1962	Gallager [30]	Low Density Parity Check (LDPC) code
1966	Golomb [31]	Golomb and Rice Codes
1972	Bahl *et al.* [32]	Maximum A-Posteriori (MAP) algorithm
1973	Forney [33]	The Viterbi algorithm
1974	Bahl *et al.* [34]	Symbol based MAP algorithm
1975	Elias *et al.* [8]	Elias Gamma (EG) source code
1977	Imai and Hirawaki [35]	Bandwidth-efficient MultiLevel Coding (MLC)
1978	Wolf [36] Ziv and Lempel [37]	Trellis-decoding of block codes Lempel-Ziv coding
1979	Rissanen and Landgon [38]	Describes a broad class of arithmetic codes
1982	Ungerböck [39]	Trellis-Coded Modulation (TCM)
1988	Blahut [40]	Multiple trellis-coded modulation
1989	Calderbank [41] Hagenauer *et al.* [42]	Multilevel codes and multistage decoding Soft-Output Viterbi Algorithm (SOVA)
1990	Koch and Baier [43]	Classic Log-MAP algorithm
1991	Webb *et al.* [44]	Hard-decision Star QAM/Differential Amplitude Phase Shift Keying (DAPSK)
1992	Zehavi [45]	Bit-Interleaved Coded Modulation (BICM)
1993	Berrou [46]	Turbo codes
1994	Kofman *et al.* [47] Le Goff *et al.* [48]	Performance of a multilevel coded modulation BICM-based Turbo Coded Modulation (TuCM)
1995	Robertson *et al.* [49]	Approx-Log-MAP algorithm
1997	Li and Ritcey [50]	Bit-Interleaved Coded Modulation with Iterative Decoding (BICM-ID)
1998	Robertson and Wörz [51]	Turbo trellis-coded modulation (TTCM)
2001	Richardson *et al.* [52]	Irregular LDPC code
2002	Luby [53]	Rateless Luby Transform (LT) codes
2004	Hou and Lee [54]	Multilevel LDPC codes design for semi-BICM
2006	Shokrollahi [55]	Rateless Raptor codes
2007	Yue *et al.* [56]	Finite-length LDPC code
2009	Arikan [57]	Polar codes
2012	Wang and Luo [58]	Generalized channel coding theory for random access communication
2015	Luo [59]	Generalized channel coding theory for distributed communication

may be replaced by JSCC arrangements [12] in many applications. Generally, a basic JSCC scheme routinely used in wireless digital communications system is represented by Figure 4. The history and milestones of JSCC development are listed in Table II. Diverse methods have been proposed for JSCC, which we will now briefly discuss.

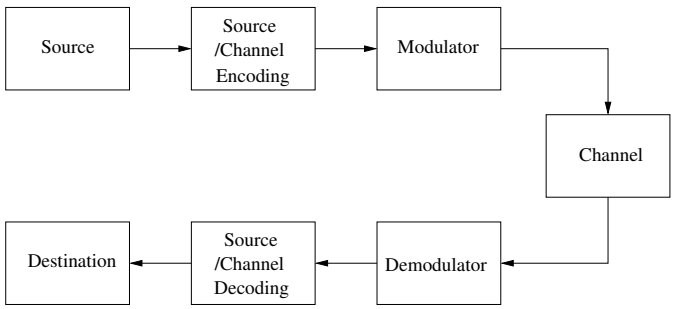


Fig. 4. A digital communications system relying on Joint Source and Channel Coding.

Channel-optimised source coding [62], [63] may be employed for reducing the reconstruction error of the source, when the channel decoder is unable to correct all transmission errors. More particularly, the source encoder is designed with special consideration of the transmission errors that are most likely to occur, namely those causing a particular binary codeword to be confused with another similar codeword. In this way, channel-optimised source encoding allocates pairs of similar codewords to represent similar reconstructed source parameter values. This principle may be applied both to scalar quantisation [87], [88] and to vector quantisation [63], [67]. Furthermore, JSCC may be beneficially employed, if some of the source correlation is not removed during source encoding [68], or some redundancy is intentionally introduced during source coding [89]. Based on this approach, the receiver is capable of exploiting the residual redundancy in order to provide an error correction capability, which may be exploited for mitigating any transmission errors that could not be eliminated during channel decoding.

JSCC has been successfully applied to the encoding of

TABLE II
MILESTONES IN JOINT SOURCE AND CHANNEL CODING (JSCC).

Year	Author(s)	Contribution
1959	Kotelnikov [60]	JSCC using Shannon-Kotelnikov mappings
1969	Kurtenbach and Wintz [61]	Quantizing for noisy channels
1978	Massey [12]	JSCC tutorial
1980	Linde *et al.* [62]	Channel-optimised indexing of vector quantisation
1984	Kumazawa *et al.* [63]	Channel-optimised vector quantisation
1986	Montgomery and Abrahams [64]	Self-Synchronizing Variable Length Codes
1987	Farvardin *et al.* [65]	Optimal quantizer design for noisy channels
1989	Wyrwas and Farrell [66]	JSCC for binary image transmission
1990	Farvardin [67]	Channel-optimised vector quantisation
1991	Sayood and Borkenhagen [68]	Use residual redundancy in JSCC design
1993	Ramchandran *et al.* [69]	Multiresolution JSCC for digital broadcast
1995	Takishima *et al.* [13]	Reversible Variable Length Codes
1998	Kozintsev and Ramchandran [70]	Multi-resolution JSCC over energy-constrained time-carrying channels
1999	Dyck and Miller [71]	JSCC in video transmission
2000	Cai and Chen [72] Buttigieg and Farrell [14]	Robust JSCC in image transmission Variable Length Error Correction codes
2001	Görtz [73] Balakirsky [74]	Iterative JSCC decoding framework JSCC using variable length codes
2002	Ramstad [75]	Analog JSCC using Shannon mapping
2003	Hagenauer and Görtz [76]	The turbo principle in JSCC
2005	Kliewer and Thobaben [77]	Iterative JSCC of variable length codes
2006	Thobaben and Kliewer [78]	Even Weight Variable Length Codes
2007	Jaspar *et al.* [79] Xu *et al.* [80]	JSCC turbo technique for discrete-sources Distributed JSCC using Raptor codes
2009	Maunder and Hanzo [81]	Irregular Variable Length Code
2011	Minero *et al.* [82]	JSCC via hybrid coding
2012	Persson *et al.* [83]	JSCC for the MIMO broadcast channel
2013	Kostina and Verdu [84]	Lossy JSCC having a finite block length
2014	Romero *et al.* [85]	Analog JSCC for wireless optical communications
2015	Tridenski *et al.* [86]	Ziv-Zakai-Rényi bound for JSCC

symbols selected from finite sets, such as the 26 letters of the English alphabet $\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \ldots, \mathrm{z}\}$. However, when the source symbol values are selected from a set having an infinite cardinality, such as the positive integers in the range of 1 to infinity $\mathbb{N}_1 = \{1, 2, 3, \ldots, \infty\}$, the existing JSCCs, such as Self-Synchronizing Variable Length Code (SSVLC) [64], RVLC [13], VLEC codes [14], Even Weight Variable Length Code (EWVLC) [78] and IrVLC [81] become unsuitable. More specifically, when the cardinality of the symbol value set is infinite, the trellis and graph structures [74], [90]–[96] employed by these codes become infinitely large, hence the corresponding decoding algorithms become infinitely complex. This motivates the work in this treatise.

C. Motivation

The motivation of our UEC code is summarized in Table III. For those source symbols that are selected from a set having finite cardinality, classic SSCC based on Huffman [4] or Shannon-Fano [25] codes may indeed be capable of reconstructing the source information with an infinitesimally low probability of error, provided that the transmission rate does not exceed the channel's capacity [4]. However, SSCC schemes require both the transmitter and receiver to accurately estimate the occurrence probability of every symbol value that the source produces. For example, in the English alphabet, the letter 'e' occurs with a much higher probability than the letter 'x'. 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 so-called universal codes, such as the EG codes [8], 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. In order to exploit the residual redundancy and hence to achieve near-capacity operation, the classic SSCC schemes may be replaced by JSCC arrangements [12], such as the VLEC code [14]. However, the decoding complexity of all previous JSCCs, such as RVLC [13] and VLEC codes [14], increases rapidly with the cardinality of the symbol set becoming excessive for the cardinality of the symbols produced by practical multimedia encoders, such as H.264 [10] and H.265 [11], and asymptotically tending to infinity, when the cardinality is infinite. Against this background, we propose a novel JSCC scheme, which is referred to as the Unary Error Correction Code (UEC). As shown in Table III, our UEC is designed to fill the gap for the combination of JSCC and infinite source symbol sets.

As it will be introduced in Section IV, our UEC encoder generates a bit sequence by concatenating unary codewords [97], 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

	Finite Symbol set e.g. $\{a, b, c, \ldots, z\}$	Infinite symbol set e.g. $\mathbb{N} = \{1, 2, 3, \ldots, \infty\}$
Separate Source and Channel Coding (SSCC)	• Shannon-Fano code [4] • Huffman code [27]	• Unary code [97] • Elias Gamma code [8]
Joint Source and Channel Coding (JSCC)	• Variable Length Error Correction (VLEC) code [14]	• **Unary Error Correction (UEC) code**

TABLE III
THE MOTIVATION FOR OUR UEC CODE.

residual redundancy using the classic Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [98]. 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.

III. NATURE OF THE SOURCE

In this section, we introduce the source distributions considered in this paper, which are inspired by those of H.265. However, the particular distribution of the source symbols produced during H.265 encoding are sensitive to the particular selection of video encoding parameters and to the particular video sequences being encoded. Owing to this, we prefer to consider model source distributions, which can be parametrized to be representative of the H.265 distribution in various applications, as well as of a wide variety of other multimedia source distributions. We begin in Section III-A by introducing a stationary zeta distribution, which is inspired by the H.265 distribution. This is extended in Section III-B, to provide the non-stationary zeta source distribution that is applied throughout the rest of this paper. This non-stationary distribution is inspired by the non-stationary nature of the H.265 source distribution.

A. Stationary zeta distribution

As shown in Figure 7, the proposed learning-aided UEC scheme is designed to convey a sequence of successive symbol vectors, where each vector $\mathbf{x} = [x_i]_{i=1}^{a}$ can be obtained as the realization of 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 the 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

$$H_X = \sum_{x \in \mathbb{N}_1} H[P(x)], \tag{1}$$

where $H[p] = p \log_2(1/p)$ [16].

Figure 5 illustrates the overall distribution of the symbol values that are entropy encoded by the H.265 video encoder, for the case of a particular video encoder parametrization and for a particular set of video sequences [99, Page 94]. The H.265 distribution of Figure 5 corresponds to a symbol entropy of $H_X = 2.348$ bits per symbol. Note that these symbol values appear to obey Zipf's law [17], since the H.265 distribution may be approximated by the zeta distribution.

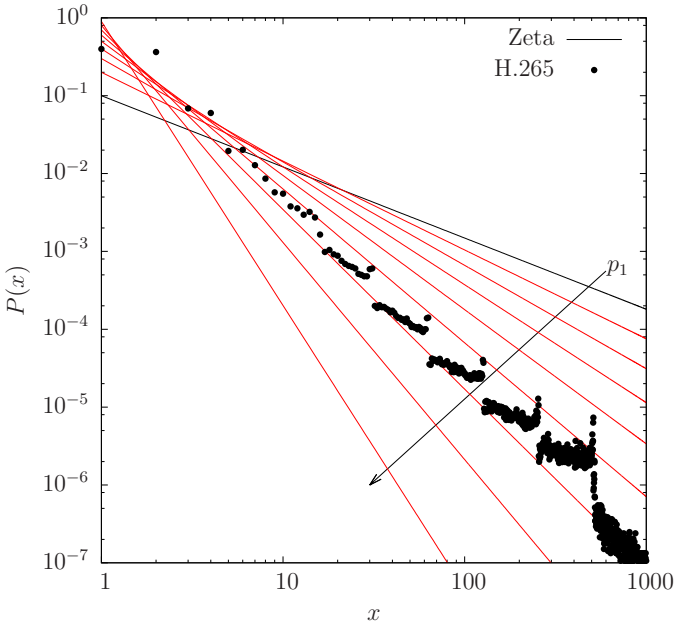


Fig. 5. The zeta probability distributions for $p_1 \in \{0.1, 0.2, 0.3, \ldots, 0.9\}$, as well as the H.265 distribution. This was obtained by recoding the values of the symbols that are EG- and UnaryMax-encoded when the HM 13.0 H.265 video encoder employs the 'encoder_randomaccess_main.cfg' and 'encoder_lowdelay_main.cfg' configurations to encode the 226 million symbols that occur during the 220 seconds of video from the 24 video sequences that are commonly used in High Efficiency Video Coding (HEVC) [99, Page 94].

More specifically, the stationary zeta probability distribution [17] is defined as

$$P(x) = \frac{x^{-s}}{\zeta(s)}, \tag{2}$$

where $\zeta(s) = \sum_{x \in \mathbb{N}_1} x^{-s}$ is the Riemann zeta function and s parametrizes the distribution. Alternatively, the zeta distribution may be parameterized by $p_1 = \Pr(X_i = 1) = 1/\zeta(s)$, which is the occurrence probability of the most frequently encountered symbol values, namely 1. Zeta distributions having the parameter $p_1 \in \{0.1, 0.2, 0.3, \ldots, 0.9\}$ and the corresponding symbol entropy $H_X \in \{17.458, 9.171, 6.104, 4.402, 3.267, 2.422, 1.740, 1.154, 0.612\}$ are depicted in Figure 5. Table IV exemplifies the source symbol probabilities $P(x)$ for the case of a stationary zeta distribution, having the parameter $p_1 = 0.797$, as was considered in our previous work [16]. This parametrization corresponds to a symbol entropy of 1.171 bits per symbol.

However, in contrast to the stationary zeta distributions that have been considered in our previous work, the H.265 source distribution is non-stationary, since it varies gradually with time, sometimes having a higher than average entropy and

sometimes having a lower than average entropy. Motivated by this, we propose a non-stationary zeta distribution model in this paper, as detailed in Section III-B.

TABLE IV
PROBABILITIES OF OCCURRENCE OF THE FIRST TEN SYMBOL VALUES PROVIDED BY A ZETA DISTRIBUTION SOURCE HAVING $p_1 = 0.797$, TOGETHER WITH THE CORRESPONDING UNARY AND EG CODEWORDS.

x_i	$P(x_i)$	$\mathbf{y}_i$	
		Unary	EG
1	0.7970	0	1
2	0.1168	10	010
3	0.0380	110	011
4	0.0171	1110	00100
5	0.0092	11110	00101
6	0.0056	111110	00110
7	0.0036	1111110	00111
8	0.0025	11111110	0001000
9	0.0018	111111110	0001001
10	0.0014	1111111110	0001010

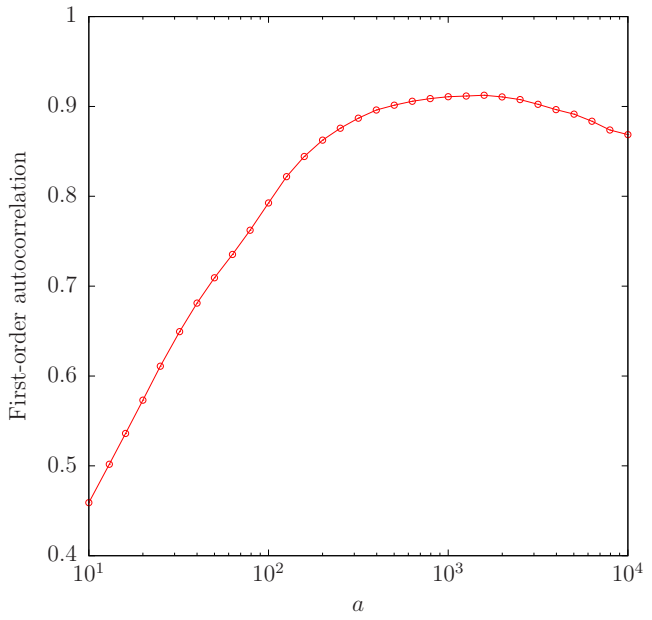


Fig. 6. The first-order autocorrelation of symbol entropy sequence, when the 226 million H.265 symbols are segmented into successive sub-vectors of length $a \in \{10, 10^{1.1}, 10^{1.2}, \ldots, 10^4\}$.

B. Non-stationary zeta distribution

In order to characterize how quickly the non-stationary source distribution varies in H.265, we segmented the sequence of 226 million symbols characterized in Figure 5 into successive vectors having a fixed length a. Following this, we measured the symbol entropy in each vector using (1), in order to obtain a corresponding sequence of symbol entropies. Finally, we measured the first-order autocorrelation of the entropy sequence, as plotted in Figure 6 as a function of the vector length a. It may be seen that the greatest first-order autocorrelation is achieved when the vector length a is around 1000. Therefore, without loss of any generality, we assume that the symbol vector $\mathbf{x}$ of Figure 7 has a length of $a = 1000$, throughout the rest of this paper.

In contrast to the stationary zeta distribution of Section III-A, the non-stationary zeta distribution produces successive symbol vectors $\mathbf{x}$ of $a = 1000$ symbols having a series of different but correlated p_1 values, as mentioned by the autocorrelation results of Figure 6. This stream of correlated p_1 values is obtained by first generating an uncorrelated stream of Gaussian distributed noise. This noise stream is then smoothed using a low-pass filter having a normalised cut-off frequency of $1/T$, where T is a parameter that specifies the number of successive symbol vectors $\mathbf{x}$ produced per cycle of variation among the correlated p_1 values. Following this, the mean and standard deviation of the filtered noise is adjusted to equal the parameter values $\bar{p_1}$ and σ, respectively. Each successive value of p_1 from this stream may then be used to parametrize the zeta distribution of (2), which is used to generate each successive vector $\mathbf{x}$ of $a = 1000$ source symbols.

As listed in Table V, the analysis of the following sections will consider seven different sets of parametrizations. In set (a) for example, the low pass filter is parametrized by $T = 40$ successive symbol vectors $\mathbf{x}$ per p_1 cycle, by a mean of $\bar{p_1} = 0.8$ and by a standard deviation of $\sigma = 1/30$, resulting in 99.7% of the non-stationary p_1 values falling in the range of $[0.7, 0.9]$. Note however that the proposed learning-aided UEC scheme may be applied to arbitrary non-stationary source distributions, not just those adhering to the model described in this section.

IV. SYSTEM MODEL

The proposed learning-aided UEC scheme of Figure 5 performs the JSCC encoding and decoding of successive symbol vectors $\mathbf{x}$, in which the symbol values are selected from a set having an infinite cardinality, as described in Section III. Like conventional UEC coding, the proposed learning-aided UEC scheme does not require any knowledge of the symbol occurrence probabilities at the transmitter. However, in contrast to conventional UEC coding, the proposed scheme does not require this knowledge at the receiver either, since it can gradually estimate the source probability distribution from the recovered symbols. In Section IV-A and IV-B, we will introduce the operations of the transmitter and the receiver, respectively. Following this, Section V will discuss the operation of the proposed learning mechanism.

A. Transmitter operation

As shown in Figure 7, the learning-aided UEC scheme encodes the source vector $\mathbf{x}$ using a unary encoder. Each symbol x_i in the vector $\mathbf{x} = [x_i]_{i=1}^{a}$ is represented by a corresponding codeword $\mathbf{y}_i$ that comprises x_i bits, namely $(x_i - 1)$ one-valued bits followed by a single zero-valued bit, as exemplified in Table IV. Note that the average codeword length l of $\mathbf{y}_i$ is given by

$$l = \sum P(x)x. \tag{3}$$

The output of the unary encoder is generated by concatenating the selected unary 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 $\mathbf{x} = [1, 4, 2, 1, 1, 3, 1, 2]$ of $a = 8$ symbols yields the $b = 15$-bit

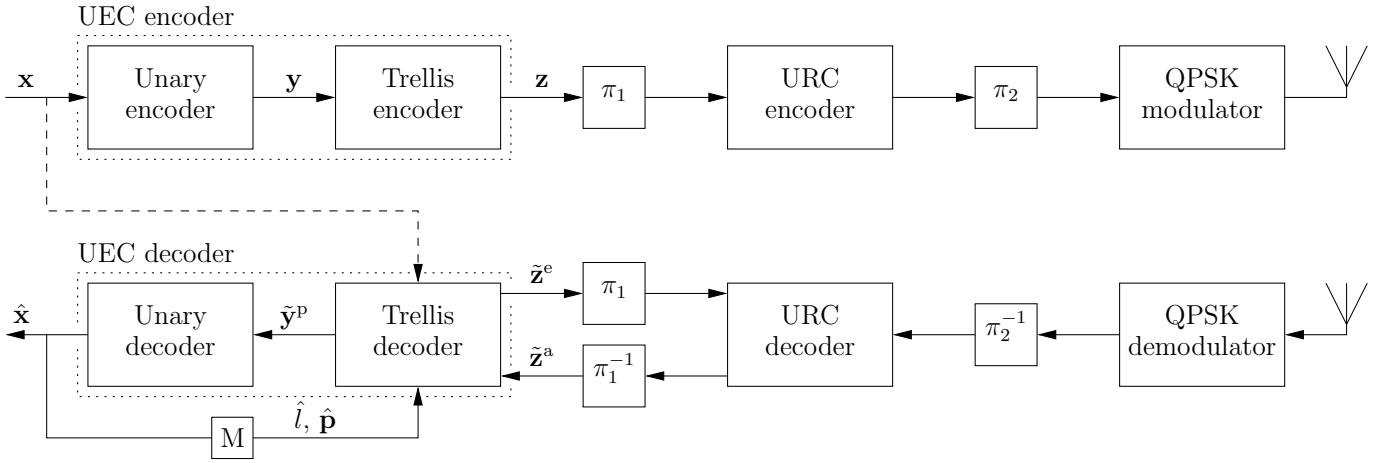


Fig. 7. Schematic of the proposed learning-aided UEC scheme, in which a UEC code is serially concatenated with a URC code and Gray-mapped QPSK modulation. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers. Block M represents the memory storage that is used to store the statistics observed from successive recovered symbol vectors $\hat{\mathbf{x}}$. Bold notation without a diacritic is used to denote a symbol or bit vector. A diacritical tilde represents an LLR vector pertaining to the bit vector with the corresponding notation. The superscripts 'a', 'e' and 'p' denote *a priori*, extrinsic and *a posteriori* LLRs, respectively.

vector $\mathbf{y} = [011101000110010]$. Note that the average length of the bit vector $\mathbf{y}$ is given by $a \cdot l$.

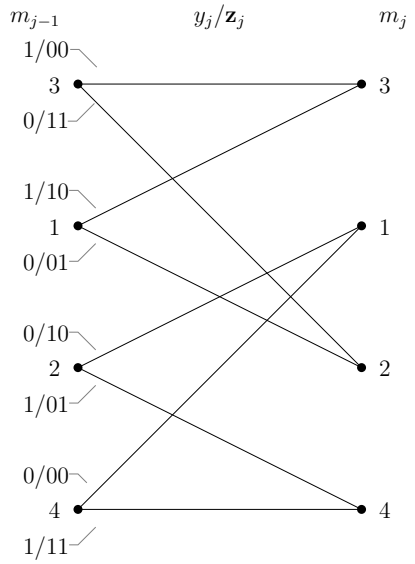


Fig. 8. An $r = 4$-state $n = 2$-bit UEC trellis, having codeword $\mathbb{C} = \{01, 11\}$.

Following unary encoding, the trellis encoder of Figure 7 is employed to encode the bit vector $\mathbf{y}$. Figure 3(a) in [16] illustrates the generalized r-state UEC trellis, while Figure 8 of this paper exemplifies an $r = 4$-state UEC trellis. Each bit y_j of the input bit sequence $\mathbf{y} = [y_j]_{j=1}^{b}$ forces the trellis encoder to traverse from its previous state $m_{j-1} \in \{1, 2, \ldots, r\}$ to its next state $m_j \in \{1, 2, \ldots, r\}$, in order of increasing bit-index j. Each next state m_j is selected from two legitimate alternatives, depending on the bit value y_j, according to

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

where the number of possible states r is required to be even and the encoding process always begins from the state $m_0 = 1$. Here, the function $\mathrm{odd}(\cdot)$ yields 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 $\mathbf{m} = [m]_{j=0}^{b}$ comprising $(b + 1)$ state values. For example, the bit vector $\mathbf{y} = [011101000110010]$ yields the path $\mathbf{m} = [1, 2, 4, 4, 4, 1, 3, 2, 1, 2, 4, 4, 1, 2, 4, 1]$ through the $r = 4$-state trellis of Figure 8. Note that the path $\mathbf{m}$ through the trellis of Figure 8 remains synchronised with the unary codewords, since the zero-valued bit at the end of each codeword $\mathbf{y}_i$ returns the path $\mathbf{m}$ to either state 1 or state 2, depending on whether the codeword $\mathbf{y}_i$ represents a symbol x_i having an odd or even index i. 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 [16, (8)].

The trellis encoder represents each bit y_j in the vector $\mathbf{y}$ by an n-bit codeword $\mathbf{z_j}$. This 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 $\overline{\mathbb{C}} = \{\overline{\mathbf{c}_1}, \overline{\mathbf{c}_2}, \ldots, \overline{\mathbf{c}_{r/2-1}}, \overline{\mathbf{c}_{r/2}}\}$, which is achieved according to

$$\mathbf{z}_j = \begin{cases} \overline{\mathbf{c}_{\lceil m_{j-1}/2 \rceil}} & \text{if } y_j = \mathrm{odd}(m_{j-1}) \\ \mathbf{c}_{\lceil m_{j-1}/2 \rceil} & \text{if } y_j \neq \mathrm{odd}(m_{j-1}) \end{cases}. \quad (5)$$

For example, the $n = 2$-bit codewords $\mathbb{C} = \{01, 11\}$ are employed in the $r = 4$-state UEC trellis of Figure 8. Finally, the selected codewords are concatenated to obtain the $(b \cdot n)$-bit vector $\mathbf{z} = [z_k]_{k=1}^{bn}$ of Figure 7. For example, the path $\mathbf{m} = [1, 2, 4, 4, 4, 1, 3, 2, 1, 2, 4, 4, 1, 2, 4, 1]$ through the $r = 4$-state $n = 2$-bit trellis of Figure 8 corresponds to the encoded bit vector $\mathbf{z} = [0101111100101110010111100010100]$. Note that the UEC encoder does not require any knowledge of the source distribution, since the output bit vector $\mathbf{z}$ of the UEC encoder only depends on the symbol vector $\mathbf{x}$ and the parametrization of the UEC trellis. This is true for both the conventional UEC scheme and our proposed learning-aided

UEC scheme.

When the source distribution is stationary, the overall average coding rate $R_{\rm o}$ of the UEC encoder is given by $R_{\rm o} = \frac{H_X}{nl}$. However, if the source distribution is non-stationary, then the symbol entropy H_X and the average unary codeword length l will vary from frame to frame. In this case, the average UEC coding rate $R_{\rm o}$ is given by the expectation

$$R_{\rm o} = E\left\{\frac{H_X}{nl}\right\}, \tag{6}$$

which may be estimated experimentally.

As shown in Figure 7, the UEC-encoded bit vector $\mathbf{z}$ is interleaved in the block π_1, encoded by the URC encoder and then interleaved again by the block π_2. Here, we recommend a 2-state URC having the generator polynomial $[1, 0]$ and the feedback polynomial $[1, 1]$, as characterized in Figure 9.6 of [7]. Following this, Gray-mapped Quaternary Phase Shift Keying (QPSK) modulation may be employed for transmission, as shown in Figure 7. Consequently, the effective throughput is given by $\eta = R_{\rm o} \cdot R_{\rm i} \cdot \log_2(M)$ bits per symbol, where we have $R_{\rm i} = 1$ for the URC coding rate and $M = 4$ for the QPSK modulation order.

B. Receiver operation

In the receiver of Figure 7, Gray-mapped QPSK demodulation and deinterleaving in the block π_2^{-1} are performed, before commencing iterative decoding between the URC and UEC decoders. Here, the two decoders exchange vectors of LLRs, which are interleaved and deinterleaved in the blocks π_1 and π_1^{-1}, respectively. Both of these decoders apply the BCJR algorithm [98] to their respective trellises, where the UEC trellis decoder may employ the trellis of Figure 8, which has only a modest complexity.

As shown in Figure 7, the UEC trellis decoder is provided with a vector of *a priori* Logarithmic Likelihood Ratios (LLRs) $\tilde{\mathbf{z}}^{\rm a} = [\tilde{z}_k^{\rm a}]_{k=1}^{bn}$ that pertain to the corresponding bits in the vector $\mathbf{z}$. These *a priori* LLRs are used to generate the vector of extrinsic LLRs $\tilde{\mathbf{z}}^{\rm e} = [\tilde{z}_k^{\rm e}]_{k=1}^{bn}$, which also pertain to the corresponding bits in the vector $\mathbf{z}$. 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 BCJR algorithm can exploit the synchronization between the UEC trellis and the unary codewords, in order to improve the receiver's error correction capability and to facilitate near-capacity operation. This is achieved by including the conditional transition probability $\Pr(M_j = m|M_{j-1} = m') = P(m|m')$ as an additional term during the BCJR algorithm's γ_t calculation of [98, Equation (9)], where we have

$$P(m|m') = \frac{P(m, m')}{\sum_{\check{m}=1}^{r} P(\check{m}, m')}, \tag{7}$$

and $P(m, m')$ is given in [16, (8)], which depends on the symbol probability distribution $P(x)$ as described in Section IV-A.

Note that knowledge of the entire symbol probability distribution $P(x)$ is not required in order to exploit (7). 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 $\mathbf{p} = [P(x)]_{x=1}^{r/2-1}$ [16]. In the conventional UEC scheme of [16], the average length l and the probability vector $\mathbf{p}$ are assumed to be stationary and known at the receiver, as represented by the dashed line in Figure 7. However, since $P(x)$ is non-stationary and unknown at the receiver of the proposed learning-aided UEC scheme, it must estimate l and $\mathbf{p}$ heuristically and iteratively, before this information can be exploited by (7). The mechanism proposed for this learning process will be discussed in Section V. In the absence of this information, the $\Pr(M_j = m|M_{j-1} = m')$ term can be simply omitted from the γ_t calculation of [98, Equation (9)], at the cost of degrading the receiver's error correction capability.

In each decoding iteration, the UEC trellis decoder may also invoke the BCJR algorithm for generating the vector of *a posteriori* LLRs $\tilde{\mathbf{y}}^{\rm p} = [\tilde{y}_j^{\rm p}]_{j=1}^{b}$ that pertain to the corresponding bits in the vector $\mathbf{y}$. The unary decoder of Figure 7 exploits the observation that each of the a unary codewords in the vector $\mathbf{y}$ contains only a single zero-valued bit. This is achieved by sorting the *a posteriori* LLRs in the vector $\tilde{\mathbf{y}}^{\rm p}$ 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 Figure 7, which is guaranteed to comprise a number of symbols owing to the above-described technique. Note that the value of a is assumed to be known to the receiver. In practice, this may be achieved by either using a constant value for a that is hard-coded into the receiver or by reliably conveying the value of a from the transmitter to the receiver using a small amount of side information. The iterative exchange of LLRs between the UEC and URC decoders of Figure 7 continues until a particular number of iterations have been completed or until the correctly decoded symbol vector $\hat{\mathbf{x}}$ has been obtained, which may be detected using a Cyclic Redundancy Check (CRC) code in practice, for example.

V. Learning Algorithm

As discussed in Section IV-B, the only knowledge that the UEC trellis decoder requires in order to facilitate near-capacity operation is the average unary codeword length l and the probabilities of the first $r/2 - 1$ symbol values $\mathbf{p} = [P(x)]_{x=1}^{r/2-1}$. When the probability distribution $P(x)$ of source symbols $\mathbf{x}$ is non-stationary and unknown, our learning-aided UEC scheme is able to heuristically and iteratively estimate l and $\mathbf{p}$ from the recovered symbol vectors $\hat{\mathbf{x}}$, and feed them back to the trellis decoder as *a priori* information, in order to improve the receiver's error correction capability.

The estimation is implemented using the memory storage block labeled M in Figure 7. This memory storage is used to store source distribution statistics that have been observed from successive recovered symbol vectors $\hat{\mathbf{x}}$. We quantify the size of this memory storage in terms of the number M of the most-recently recovered symbol vectors $\hat{\mathbf{x}}$ from which the statistics are derived. As described in Section IV-B, the number a of symbols in each vector $\mathbf{x}$ and the number b of bits in each

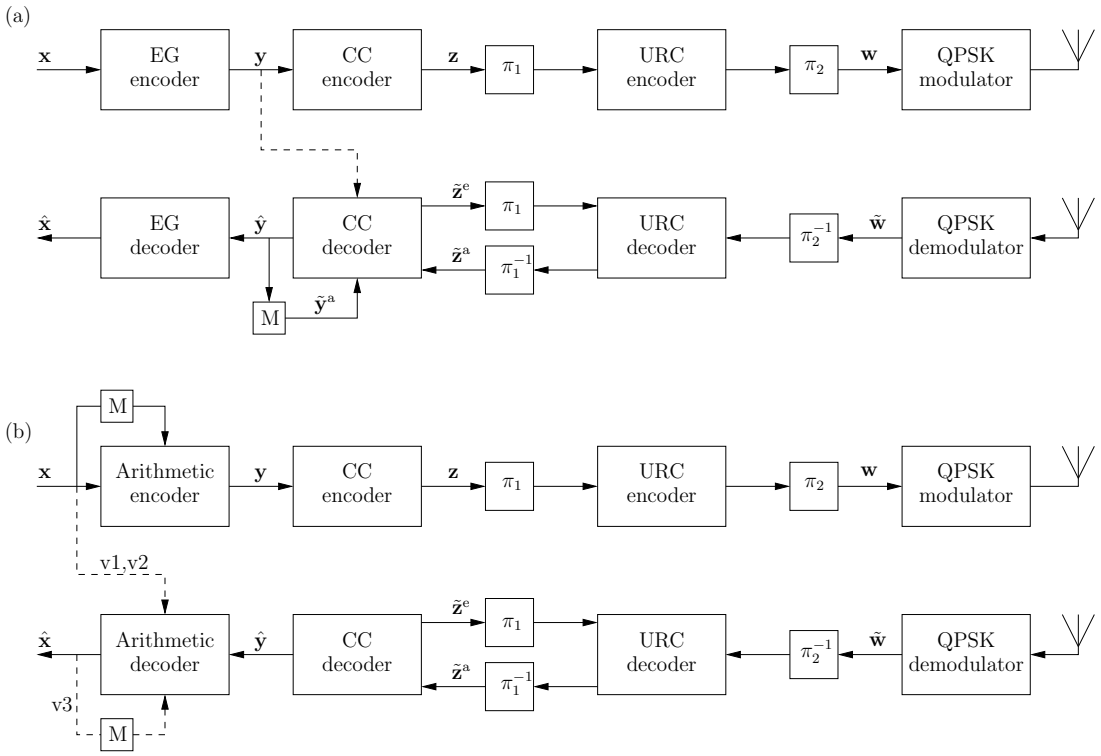


Fig. 9. Schematics of (a) the learning-aided EG-CC and (b) the learning-aided Arithmetic-CC benchmarkers, which employ serial concatenation with a URC code and a Gray-mapped QPSK modulation scheme. Here, π_1 and π_2 represent interleavers, while π_1^{-1} and π_2^{-1} represent the corresponding deinterleavers. For the sake of controlling the effective throughput η, doping or puncturing may be performed by π_2. Block M represents the memory that is used to store the statistics observed from successive recovered bit vectors $\hat{\mathbf{y}}$ or symbol vectors $\hat{\mathbf{x}}$.

vector $\mathbf{y}$ are known to the decoder. Owing to this, following the decoding of each frame, the average unary codeword length l can be estimated by

$$\hat{l} = \frac{\sum_{m=1}^{\mathrm{M}} b_m}{\sum_{m=1}^{\mathrm{M}} a_m}, \tag{8}$$

which can then be used to aid the decoding of the next frame. Here, a_m and b_m are the number of symbols and number of bits in the m-th most-recently recovered frame, respectively. In this way, we can also count the occurrences of the first $r/2-1$ symbol values in each recovered symbol vector $\hat{\mathbf{x}}$. Therefore, following the decoding of each frame, the probability vector $\mathbf{p}$ can be estimated by

$$\hat{\mathbf{p}} = \left[\frac{\sum_{m=1}^{\mathrm{M}} N(\hat{\mathbf{x}}_m = x)}{\sum_{m=1}^{\mathrm{M}} a_m} \right]_{x=1}^{r/2-1}, \tag{9}$$

which can then also be used to aid the decoding of the next frame. Here, $N(\hat{\mathbf{x}}_m = x)$ denotes the number of recovered symbols that has a value of x in the m-th most-recently recovered symbol vector $\hat{\mathbf{x}}_m$. Note that the memory will be empty during the decoding of the first frame. In this case, $\hat{l}$ and $\hat{\mathbf{p}}$ are not exploited during UEC decoding, by omitting the $\Pr(M_j = m | M_{j-1} = m')$ term from the γ_t calculation of [98, Equation (9)], as described in Section IV-B. Following this, while M frames have not been received yet, the summations in (8) and (9) are adjusted accordingly.

Note that, during this initial transient phase, the estimated $\hat{l}$ and $\hat{\mathbf{p}}$ may not have accurate values, particularly if transmission errors occur owing to low SNR. Nevertheless, this imperfect *a priori* information can still contribute to improve the error correction capability of the trellis decoder, resulting in fewer errors in $\hat{\mathbf{x}}$ and more reliable feedback of $\hat{l}$ and $\hat{\mathbf{p}}$ for the next frame. In this way, our learning-aided UEC scheme is able to gradually learn the statistics of the source distribution and iteratively update the estimated $\hat{l}$ and $\hat{\mathbf{p}}$ on a frame-by-frame basis. As a result, the performance of the decoder can be continually improved until a steady-state phase is reached, where upon near-capacity operation is obtained.

For the stationary probability distribution, it is clear to see that the more memory storage is applied, the more accurate source distribution statistics can be obtained. In this case, the difference between the estimated values of $\hat{l}$ and $\hat{\mathbf{p}}$ and the real values of l and $\mathbf{p}$ will become infinitesimal, when infinite memory is applied, giving

$$\lim_{\mathrm{M}\to\infty} \hat{l} = l \quad \text{and} \quad \lim_{\mathrm{M}\to\infty} \hat{\mathbf{p}} = \mathbf{p}. \tag{10}$$

However, in the case of non-stationary source distributions, it is not desirable to apply infinite memory storage. This is not only because of the impracticality of infinite memory, but also because the source distribution of the current frame is only correlated to those of recent frames, when the source is non-stationary. If the size of the memory storage is too large, then the estimates $\hat{l}$ and $\hat{\mathbf{p}}$ will be contaminated by out-of-date source distribution statistics, damaging the error correction capability of the scheme. Therefore, there is a trade-off between collecting enough statistics to make a good

estimation and collecting too many out-of-date statistics. In the simulations of Section VI, we will investigate this trade-off, in order to optimize the overall performance of our proposed learning-aided scheme. We will also consider an idealized but impractical version of the proposed learning-aided UEC scheme, in which a genie provides the receiver with perfect knowledge of l and $\mathbf{p}$ for each frame. As a result, this version provides a baseline, which characterizes the bound on the performance that the learning-aided UEC scheme can achieve.

VI. Benchmarkers and SER Performance

In this section, we compare the proposed learning-aided UEC scheme to two SSCC benchmarkers that employ a similar learning strategy, as well as to corresponding idealized but impractical versions of all schemes. In the learning-aided EG-CC benchmarker of Figure 9(a), the unary code of Figure 7 is replaced by an EG code and the UEC trellis code is replaced by a CC code. As a further step, the learning-aided Arithmetic-CC benchmarker is obtained by replacing the EG code by an arithmetic code, as shown in Figure 9(b). The design and parametrizations of the two benchmarkers are detailed in Section VI-A and Section VI-B, respectively. Section VI-C compares the SER performance of our proposed learning-aided UEC with that of the benchmarkers.

A. Learning-aided EG-CC benchmarker

In the transmitter of the learning-aided EG-CC scheme, an EG encoder is employed to convert the symbol vector $\mathbf{x}$ into the bit vector $\mathbf{y}$, which typically has non-equiprobable bit values. Here, the first ten codewords of the EG code are given in Table IV. When the source distribution is stationary, the average EG codeword length is given by $l = \sum_{x \in \mathbb{N}_1} P(x) \left(2\lfloor \log_2(x) \rfloor + 1\right)$. However, when the source distribution is non-stationary, the average EG codeword length will vary from frame to frame, as described in Section IV-A. Following EG encoding, $\mathbf{y}$ is CC encoded to obtain the bit vector $\mathbf{z}$, as shown in Figure 9(a). The CC encoder may be described by the generator and feedback polynomials provided in [16, Table II]. Here, we employ the $n = 2$-bit $r = 4$-state CC code, in order to facilitate a fair comparison with the case where trellis encoder of the learning-aided UEC scheme employs the UEC trellis of Figure 8.

In the case of a non-stationary source distribution, the EG-CC coding rate R_{o} may be quantified by the expectation of (6) and will typically differ from that of the UEC scheme. For the sake of fair comparisons, either doping or puncturing may be applied in the block π_2 of Figure 9 in order to achieve the same effective throughput η for all schemes. Doping is applied when the coding rate R_{o} of EG-CC encoding is higher than that of UEC encoding, while puncturing is applied when the coding rate of EG-CC encoding is lower. More particularly, in the doping operation of π_2, a certain number of bits from the tail of the interleaved bit vector $\mathbf{w}$ are duplicated and appended to the end of $\mathbf{w}$. In the receiver, the de-doping operation of π_2^{-1} is achieved by removing the corresponding LLRs of the vector $\tilde{\mathbf{w}}$, before adding them into the LLRs that now form the end of $\tilde{\mathbf{w}}$. By contrast, when puncturing is applied in π_2, a certain number of bits are truncated from the tail of the bit vector $\mathbf{w}$. In the corresponding de-puncturing operation of π_2^{-1}, the positions in the LLR vector $\tilde{\mathbf{w}}$ that correspond to the truncated bits are filled by zero-valued LLRs. The inner coding rate R_{i} is defined as the ratio of the number of bits entering π_2 to the number of bits emerging from π_2, where $R_{\mathrm{i}} > 1$ for puncturing as $R_{\mathrm{i}} < 1$ for doping, as listed in Table V.

During iterative decoding, the CC decoder employs the BCJR algorithm to convert the vector of *a priori* LLRs $\tilde{\mathbf{z}}^{\mathrm{a}}$ into the vector of extrinsic LLRs $\tilde{\mathbf{z}}^{\mathrm{e}}$, and it employs the Viterbi algorithm [20] to convert $\tilde{\mathbf{z}}^{\mathrm{a}}$ into the vector of recovered bits $\hat{\mathbf{y}}$, which is then EG decoded to obtain the recovered symbol vector $\hat{\mathbf{x}}$. Note that in the receiver of the learning-aided EG-CC scheme, the CC decoder is not able to exploit knowledge of the average codeword length l or the symbol probability distribution $P(x)$, like the UEC trellis decoder. However, the BCJR algorithm employed by the CC decoder is able to exploit knowledge of the probability of occurrence of the binary values in the bit vector $\mathbf{y}$ output by the EG encoder. More specifically, the CC decoder is provided with a vector $\tilde{\mathbf{y}}^{\mathrm{a}} = [y_j^{\mathrm{a}}]_{j=1}^{b}$ of *a priori* LLRs, having identical values. As shown in Figure 9(a), we consider two different versions of the learning-aided EG-CC benchmarker, depending on how the value used for all LLRs in the vector $\tilde{\mathbf{y}}^{\mathrm{a}}$ is obtained.

- 1. Idealized but impractical *a priori* information
  As indicated by the dashed line in Figure 9(a), the first version of the learning-aided EG-CC benchmarker relies on an impractical genie to provide the receiver with perfect knowledge of the bit value probabilities of each frame, which can be used to provide the idealized *a priori* LLR vector $\tilde{\mathbf{y}}^{\mathrm{a}}$. As a result, this version provides a baseline, which characterizes the bound on the performance that the learning-aided EG-CC benchmarker can achieve.
- 2. Estimated *a priori* information
  Based on the learning technique of Section V, the second version of the learning-aided EG-CC benchmarker employs memory M at the receiver to store the bit probability statistics obtained from the M most-recently recovered bit vectors $\hat{\mathbf{y}}$, as shown in Figure 9(a). In this way, the *a priori* LLRs can be estimated as

$$\tilde{\mathbf{y}}^{\mathrm{a}} = \ln \frac{\sum_{m=1}^{\mathrm{M}} w(\hat{\mathbf{y}}_m)}{\sum_{m=1}^{\mathrm{M}} \left(l(\hat{\mathbf{y}}_m) - w(\hat{\mathbf{y}}_m)\right)}, \tag{11}$$

  where $w(\hat{\mathbf{y}}_m)$ and $l(\hat{\mathbf{y}}_m)$ are the weight and length of the m-th most-recently recovered bit vector $\hat{\mathbf{y}}_m$, respectively. As in the learning UEC scheme, the number of frames M considered by the memory storage may be optimized in the case of non-stationary source probability distributions. Note that during the transient phase, the memory and the *a priori* LLR vector $\tilde{\mathbf{y}}^{\mathrm{a}}$ are operated in analogy with the technique employed by the proposed learning-aided UEC scheme during the transient phase.

B. Learning-aided Arithmetic-CC benchmarker

As shown in Figure 9(b), the learning-aided Arithmetic-CC benchmarker can be obtained by replacing the EG code of Figure 9(a) by an arithmetic code. Arithmetic encoding

TABLE V

THE PARAMETRIZATION OF THE SEVEN SETS OF NON-STATIONARY SOURCE DISTRIBUTION PARAMETERS, INCLUDING FRAMES PER CYCLE T, AS WELL AS MEAN $\bar{p_1}$ AND STANDARD DEVIATION σ OF THE ZETA DISTRIBUTION PARAMETER p_1. CHARACTERISTICS ARE PROVIDED FOR THE VARIOUS SCHEMES CONSIDERED, INCLUDING OUTER CODING RATE R_o, INNER CODING RATE R_i AND EFFECTIVE THROUGHPUT η. E_b/N_0 BOUNDS ARE GIVEN FOR THE CASE OF GRAY-MAPPED QPSK TRANSMISSION OVER AN UNCORRELATED NARROWBAND RAYLEIGH FADING CHANNEL. COMPLEXITY IS QUANTIFIED BY THE AVERAGE NUMBER OF ADD, COMPARE AND SELECT (ACS) OPERATIONS INCURRED PER DECODING ITERATION AND PER SYMBOL IN THE VECTOR $\mathbf{x}$. THE OPTIMISED SIZE M OF THE MEMORY USED FOR LEARNING IS ALSO PROVIDED.

Set	T	$\bar{p_1}$	σ	Scheme	R_o	R_i	η	E_b/N_0 [dB] capacity bound	E_b/N_0 [dB] area bound	E_b/N_0 [dB] tunnel bound	Complexity
a	40	0.8	1/30	UEC	0.3726	1	0.7452	0.767	1.26	2.5	189
				EG-CC	0.3765	0.9897			1.95	3.0	186
				Arith-CC	0.5000	0.7452			1.78	2.6	141
b	40	0.85	1/30	UEC	0.3330	1	0.6660	0.485	0.87	2.3	160
				EG-CC	0.3201	1.0401			1.63	3.3	166
				Arith-CC	0.5000	0.6660			1.60	2.4	105
c	40	0.75	1/30	UEC	0.3626	1	0.7252	0.695	1.58	2.7	245
				EG-CC	0.4198	0.8638			2.61	3.1	209
				Arith-CC	0.5000	0.7252			1.90	2.7	176
d	20	0.8	1/30	UEC	0.3727	1	0.7454	0.776	1.31	2.5	189
				EG-CC	0.3786	0.9845			1.98	3.1	186
				Arith-CC	0.5000	0.7454			1.78	2.6	141
e	160	0.8	1/30	UEC	0.3718	1	0.7436	0.757	1.27	2.5	188
				EG-CC	0.3732	0.9962			1.98	3.1	186
				Arith-CC	0.5000	0.7436			1.69	2.6	139
f	40	0.8	1/60	UEC	0.3767	1	0.7534	0.810	1.32	2.5	185
				EG-CC	0.3759	1.0020			1.95	3.1	186
				Arith-CC	0.5000	0.7534			1.73	2.5	139
g	40	0.8	1/20	UEC	0.3621	1	0.7242	0.681	1.18	2.4	195
				EG-CC	0.3762	0.9626			2.02	3.1	187
				Arith-CC	0.5000	0.7242			1.60	2.5	142

and decoding both require knowledge of the entire source probability distribution. However, this corresponds to an infinite amount of knowledge when the cardinality of the source alphabet is infinite. In order to overcome this problem, the learning-aided Arithmetic-CC scheme clips all symbol values in the vector $\mathbf{x}$ to a limit of 1000, before they are arithmetic encoded. This clipping leads to some arithmetic decoding errors, since the receiver will recover the clipped value, rather than correct value of each symbol. However, this effect is small since symbol values exceeding 1000 are rare in the source distributions considered here. In case of non-stationary source distributions, the arithmetic encoder and decoder's knowledge of the source distribution must be adaptively updated, in order to reflect the changing statistics of the source. As shown in Figure 9(b), depending on how this knowledge is updated and whether synchronization between the transmitter and the receiver is assumed, we consider three different versions of the learning-aided Arithmetic-CC scheme, as follows.

- v1. Idealized but impractical knowledge of the source distribution and idealized but impractical synchronization
  As indicated by the dashed line in Figure 9(b), the first version of the learning-aided Arithmetic-CC benchmarker relies on an impractical genie to provide perfect knowledge of the source distribution of each frame to both the transmitter and receiver, guaranteeing perfect synchronization between them. As a result, this version provides a baseline, which characterizes the bound on the performance that the learning-aided Arithmetic-CC scheme can achieve.
- v2. Estimated knowledge of the source distribution and idealized but impractical synchronization
  Based on the learning technique of Section V, the second version of the learning-aided Arithmetic-CC benchmarker employs memory $\mathbf{M}$ at the transmitter to store symbol probability statistics obtained from the M previous symbol vectors $\mathbf{x}$, as shown in Figure 9(b). Therefore, in analogy to (9), the symbol occurrence probabilities can be estimated as

$$\hat{\mathbf{p}} = \left[\frac{1 + \sum_{m=1}^{\mathrm{M}} N(\hat{\mathbf{x}}_m = x)}{1000 + \sum_{m=1}^{\mathrm{M}} a_m} \right]_{x=1}^{1000}. \tag{12}$$

  Here, the addition of 1 in the numerator of (12) ensures that no symbols are attributed an estimated probability of zero and naturally allows a uniform probability distribution to be assumed when encoding the first frame, for which the memory $\mathbf{M}$ is empty. During the subsequent transient phase, the memory is operated in analogy with the learning-aided UEC scheme during its transient phase. This version of the learning-aided Arithmetic-CC benchmarker relies on an impractical genie to share the estimated symbol probabilities $\hat{\mathbf{p}}$ from the transmitter to the receiver, guaranteeing perfect synchronization.
- v3. Estimated knowledge of the source distribution and imperfect but practical synchronization
  In this practical version of learning-aided Arithmetic-CC benchmarker, the learning process of (12) is performed independently in the transmitter and receiver, based on the symbol vector $\mathbf{x}$ and the recovered symbol vector $\hat{\mathbf{x}}$, respectively. This is achieved using independent memories $\mathbf{M}$ in the transmitter and receiver, as shown in Figure 9(b). Owing to this, synchronization between the memories is not guaranteed in the presence of transmission errors, which may cause $\hat{\mathbf{x}}$ and $\mathbf{x}$ to differ.

In the case of a non-stationary source distribution, the arithmetic coding rate R_{o} may be quantified in analogy to (6), once the steady state has been reached. In order to facilitate fair comparisons, doping or puncturing may be applied in the learning-aided Arithmetic-CC scheme, in order to achieve the same steady-state effective throughput η as the learning-aided UEC and EG-CC schemes.

C. SER performance

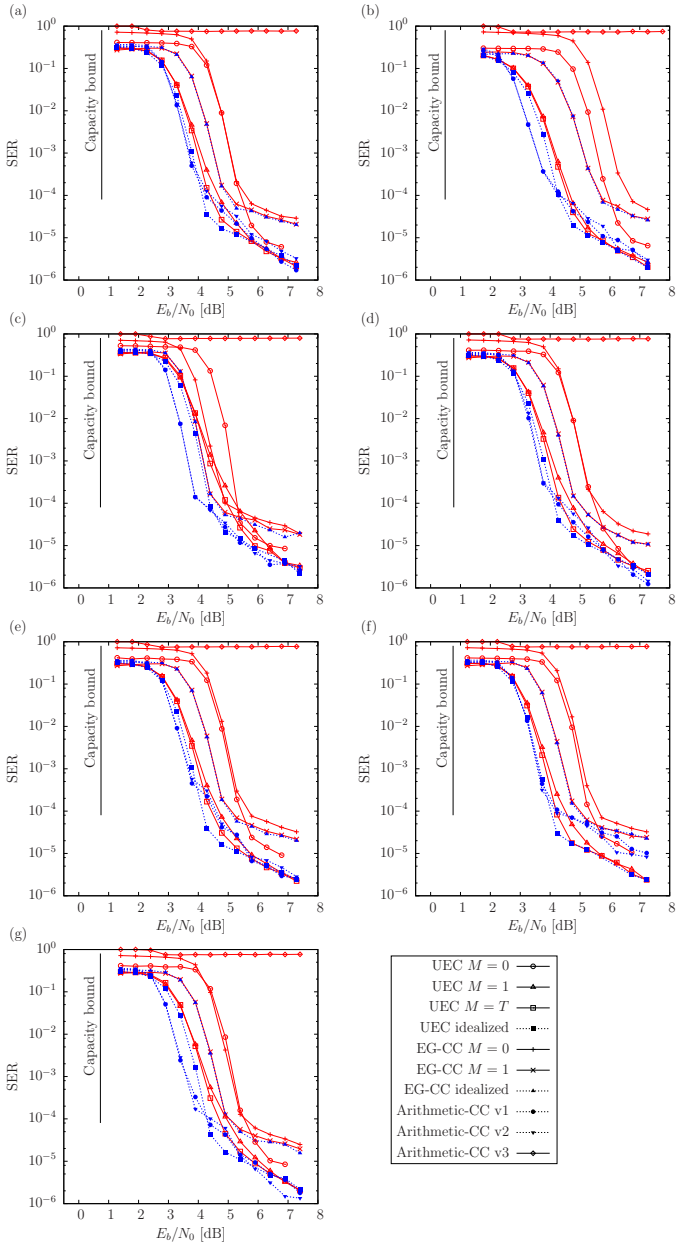


Fig. 10. SER performance for various arrangements of the proposed learning-aided UEC scheme of Figure 7, the learning-aided EG-CC benchmarker of Figure 9(a) and the learning-aided Arithmetic-CC benchmarker of Figure 9(b), when conveying symbols obeying a non-stationary zeta distribution that is generated by the corresponding parameter sets (a) to (g) of Table V, and communicating over a QPSK-modulated uncorrelated narrowband Rayleigh fading channel. A complexity limit of 7500 ACS operations per decoding iteration is imposed for decoding each of the symbols in $\mathbf{x}$.

Figure 10 compares the steady-state SER performance of the proposed learning-aided UEC scheme with those of the learning-aided EG-CC and the learning-aided Arithmetic-CC benchmarkers of Figure 9. In each of Figures 10(a) to 6(g), the source symbol vector $\mathbf{x}$ obeys the non-stationary zeta distribution of Section III-B, which is parametrized by the corresponding set shown in Table V. For example, the simulation results of Figure 10(a) corresponds to the parameter set (a), for which the parameters $T = 40$, $\bar{p_1} = 0.8$ and $\sigma = 1/30$ are employed to generate a sequence of p_1 values, as described in Section III-B. Each of those p_1 values is used to generate a symbol vector $\mathbf{x}$ comprising $a = 10^3$ symbols, which is then encoded by the various schemes considered. Transmission is performed over a Gray-mapped QPSK-modulated uncorrelated narrowband Rayleigh fading channel. Table V quantifies the number of ACS operations performed by the BCJR algorithm of each scheme per symbol of $\mathbf{x}$ and per decoding iteration [100]. In Figure 10, all SER results are obtained using a limit of 7500 ACS operations per symbol of $\mathbf{x}$, which is sufficient for achieving iterative decoding convergence in all schemes, and facilitates fair comparisons in terms of decoding complexity.

As described in Section VI-A and VI-B, we employ doping and puncturing in order to obtain the same overall effective throughput η for all schemes employing the same source distribution parameter set. Note that doping will increase the average frame length in order to give fair comparison, hence doping increases the redundancy in the transmitted bit vector and enhances the error correction capability. By contrast, puncturing will reduce the average frame length in order to give fair comparison, hence puncturing reduces the redundancy in the transmitted bit vector and degrades the error correction capability. The ninth column of Table V provides the specific E_{b}/N_0 values, where the Discrete-input Continuous-output Memoryless Channel (DCMC) capacity becomes equal to the effective 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. For example, the parameter set (a) of Table V, results in an effective throughput of $\eta = 0.7452$ bit/s/Hz for all considered schemes, yielding a corresponding DCMC capacity bound of 0.767 dB. This bound is represented by a vertical line in Figure 10(a). However, in order to facilitate the creation of an open EXtrinsic Information Transfer (EXIT) chart tunnel, it is necessary, but not sufficient, for the area A_{o} beneath the inverted outer EXIT function to exceed the area A_{i} beneath the inner EXIT function [101]. Therefore, the *area bound* provides the E_{b}/N_0 value where we have $A_{\mathrm{o}} = A_{\mathrm{i}}$, which would theoretically allow the creation of an open EXIT chart tunnel [102]. 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_{b}/N_0 value, which we refer to as the *tunnel bound*. Based on these observations, the E_{b}/N_0 bounds may be used to characterize the iterative decoding performance of our proposed scheme and the benchmarkers.

Furthermore, for the practical versions of our schemes that employ the memories shown in Figure 7 and 9, we consider different sizes M for the memories used for each parameter set. More specifically, Figure 10 considers the case of $M = 0$,

where no memory is used at the receiver and so the decoders never have any knowledge of the source distribution. When $M = 1$, only statistics derived from the previous recovered symbol vector $\hat{\mathbf{x}}$ are stored in the memory, providing the decoder with only limited knowledge about the source distribution. Despite this, the learning-aided UEC scheme benefits from an SER performance gain of up to 1 dB, when employing $M = 1$ instead of $M = 0$. As described in Section V, the size M of the memory can be optimized to suit the nature of the non-stationary source. Without loss of generality and for the sake of simplicity, Figure 10 provides SER results for the proposed learning-aided UEC scheme where the size of the memory M is set equal to the number of frames per cycle T, which parametrizes the non-stationary source distribution. Our experiments reveal that the resultant SER performance is very close to that offered by optimizing M. However, the value of optimized M may be adjusted accordingly in practice. As shown in Figure 10, the SER performance of the practical learning-aided UEC scheme having the optimized memory size M offers a performance that is as little as 0.1 dB away from the idealized but impractical UEC scheme, when the SER is 10^{-3}. By contrast, the practical learning-aided EG-CC scheme only requires $M = 1$ in order to achieve the same performance as its idealized but impractical baseline. This is because the learning-aided EG-CC scheme only needs to estimate a single source distribution statistic, namely the value of the LLR that is used for all elements of the vector $\tilde{\mathbf{y}}^{\mathrm{a}}$ provided to the CC decoder. Nevertheless, our proposed $M = 1$ learning-aided UEC scheme offers up to 0.8 dB gain compared to the $M = 1$ learning-aided EG-CC scheme at the SER of 10^{-3}, and gains of up to 0.85 dB when employing the value of $M = T$. This may explained by the capacity loss that is incurred by the SSCC used in the learning-aided EG-CC scheme, which is characterized by the large discrepancies between the E_{b}/N_0 tunnel and capacity bounds shown in Table V. Note that the gains offered by the proposed learning-aided UEC scheme are achieved for free, with no increase to decoding complexity or to transmission-energy, -bandwidth or -duration. Note that the impractical versions of the Arithmetic-CC benchmarker offer similar SER performance to our optimized practical learning-aided UEC scheme. This may be explained by the high efficiency of the arithmetic code, which results in a smaller capacity loss, compared to the EG-CC benchmarker. Note that the results of Figure 10 are provided for the case where the Arithmetic-CC scheme employs a memory size M equal to T, which was found to offer performance that closely matches that of the optimized M. However, the practical version of the learning-aided Arithmetic-CC benchmarker performances very badly, since the memories in its transmitter and receiver become easily desynchronized in the presence of transmission errors.

VII. Future Work

The research illustrated in [18], [19] and this paper can be extended in several directions. In this section, we highlight a number of potential future research ideas.

A. Adaptive/Irregular/Learning-aided EGEC, RiceEG and ExpGEC schemes

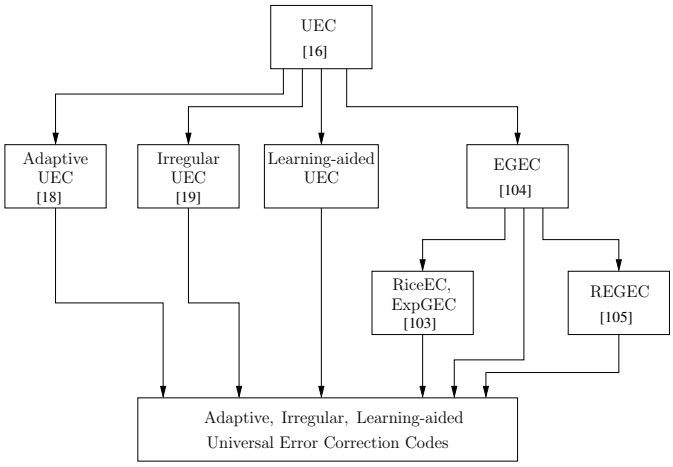


Fig. 11. The enhancement of universal error correction codes by using the techniques developed in this treatise.

The proposed UEC code is based on the unary code, which is a special case of the Rice code [31]. More specifically, the Rice code is parametrized by M, where $M = 1$ yields the unary code, as shown in Table VI. By exploiting the similarities between the unary and Rice codes, our UEC code was extended to conceive a Rice Error Correction (RiceEC) code in [103][1]. Although RiceEC codes have shorter average codeword lengths than UEC codes for some source symbol distributions, neither of them may be deemed to constitute universal codes having finite average codeword lengths for all monotonic source symbol distributions. Owing to this, both the UEC and RiceEC codes may suffer from having an average codeword length that tends to infinity for some source symbol distributions, including Zeta distributions having $p_1 \leq 0.608$ in the case of the UEC code, as described in Section IV-A.

By contrast, the Elias Gamma (EG) code [6] and the Exponential Golomb (ExpG) code [31] are universal codes, providing a finite average codeword length for any monotonic source symbol distribution. Motivated by this, universal JSCCs were obtained by extending the UEC code for conceiving the Elias Gamma Error Correction (EGEC) code of [104][2] and the Exponential Golomb Error Correction (ExpGEC) code of [103]. Both the codes are capable of facilitating the near-capacity transmission of infinite-cardinality symbol alphabets having any arbitrary monotonic probability distribution, hence having a much wider applicability. Note that the ExpG code is parametrized by k and the EG code is obtained in the special case of $k = 0$, as shown in Table VI. More specifically, Table VI characterizes the Unary, Rice, EG and ExpG codes, which map each symbol d_i in the vector $\mathbf{d} = [d_i]_{i=1}^{a}$ to the respective codeword Unary(d_i), Rice(d_i), EG(d_i) or ExpG(d_i),

[1]In which the author of this paper was co-authored and contributed to obtain the source distribution statistics.

[2]In which the author of this paper was co-authored and contributed to the simulations.

	Rice(d_i) M=1, Unary(d_i)			Rice(d_i) M=2				Rice(d_i) M=4				Rice(d_i) M=8				ExpG(d_i) k=0, EG(d_i)				ExpG(d_i) k=1				ExpG(d_i) k=2				REGEC
d_i	x_i	$\mathbf{y}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	x_i	$\mathbf{y}_i$	$\mathbf{u}_i$	t_i	
1	1	1	0	1	1	0	0	1	1	00	0	1	1	000	0	1	1		0	1	1	0	0	1	1	00	0	1
2	2	01	0	1	1	1	1	1	1	01	1	1	1	001	1	2	01	0	0	1	1	1	1	1	1	01	1	001
3	3	001	0	2	01	0	0	1	1	10	2	1	1	010	2	2	01	1	1	2	01	00	0	1	1	10	2	011
4	4	0001	0	2	01	1	1	1	1	11	3	1	1	011	3	3	001	00	0	2	01	01	1	1	1	11	3	00001
5	5	00001	0	3	001	0	0	2	01	00	0	1	1	100	4	3	001	01	1	2	01	10	2	2	01	000	0	00011
6	6	000001	0	3	001	1	1	2	01	01	1	1	1	101	5	3	001	10	2	2	01	11	3	2	01	001	1	01001
7	⋮	⋮	⋮	4	0001	0	0	2	01	10	2	1	1	110	6	3	001	11	3	3	001	000	0	2	01	010	2	01011
8	⋮	⋮	⋮	4	0001	1	1	2	01	11	3	1	1	111	7	4	0001	000	0	3	001	001	1	2	01	011	3	0000001
9				5	00001	0	0	3	001	00	0	2	01	000	0	4	0001	001	1	3	001	010	2	2	01	100	4	0000011
10				5	00001	1	1	3	001	01	1	2	01	001	1	4	0001	010	2	3	001	011	3	2	01	101	5	0001001
11				6	000001	0	0	3	001	10	2	2	01	010	2	4	0001	011	3	3	001	100	4	2	01	110	6	0001011
12				6	000001	1	1	3	001	11	3	2	01	011	3	4	0001	100	4	3	001	101	5	2	01	111	7	0100001
13								4	0001	00	0	2	01	100	4	4	0001	101	5	3	001	110	6	3	001	0000	0	
14				⋮	⋮	⋮	⋮	4	0001	01	1	2	01	101	5	4	0001	110	6	3	001	111	7	3	001	0001	1	⋮
15								4	0001	10	2	2	01	110	6	4	0001	111	7	4	0001	0000	0	3	001	0000	2	

TABLE VI

THE DECOMPOSITION OF SYMBOLS d_i INTO SUB-SYMBOLS x_i AND t_i FOR THE EGEC, RICEEC AND EXPGEC CODES [103], [104] OF FIGURE 12. EACH CODEWORD CAN BE CONSIDERED TO BE A CONCATENATION OF $\mathbf{y}_i$ AND $\mathbf{u}_i$, WHERE THE BIT-VECTOR IS DECOMPOSED INTO THE SUB-BIT-VECTORS. THE BIT-VECTORS $\mathbf{y}_i$ AND $\mathbf{u}_i$ RELATE TO THE SUB-SYMBOLS x_i AND t_i ACCORDING TO $\mathbf{y}_i$ = UNARY(x_i) AND $\mathbf{u}_i$ = FLC(t_i), RESPECTIVELY. THE FIRST 12 CODEWORDS ARE ALSO LISTED FOR THE REGEC CODE [105].

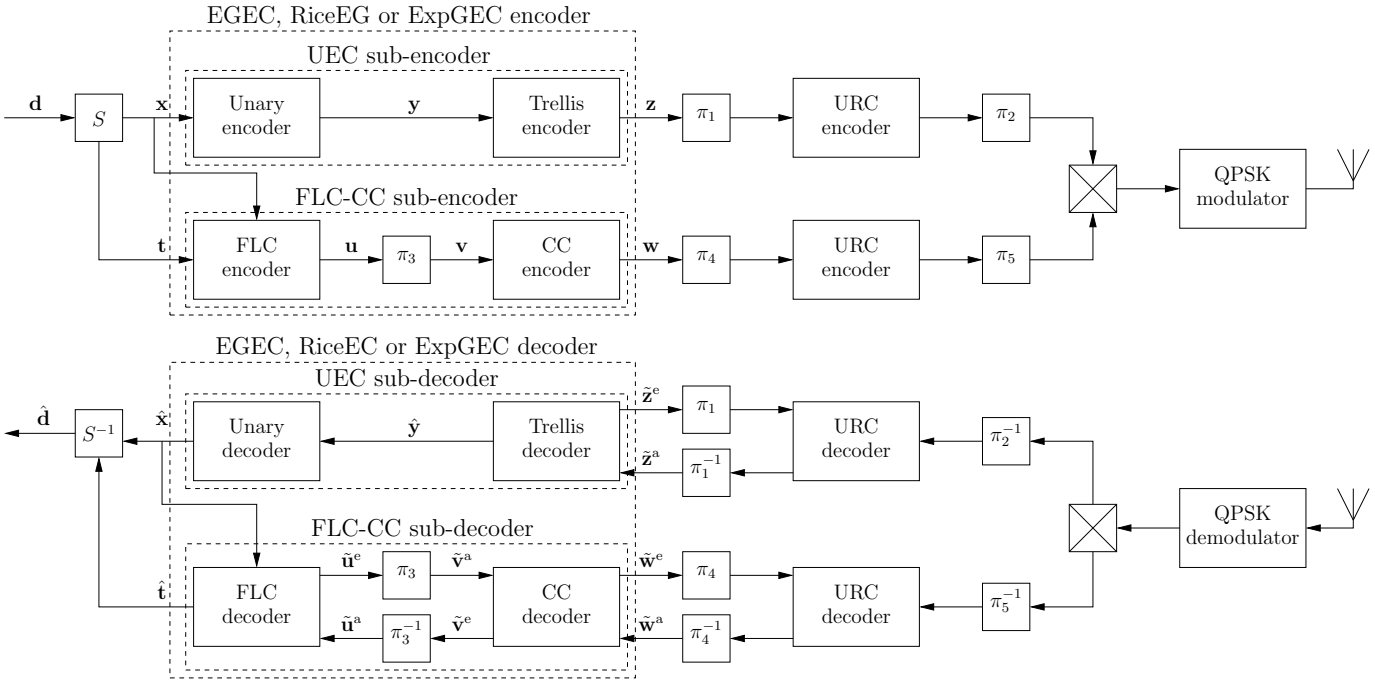


Fig. 12. The schematic of the EGEC, RiceEG and ExpGEC codes [103], [104].

where the Rice and ExpG encoders are parametrized by $M \in \{1, 2, 4, 8\}$ and $k \in \{0, 1, 2\}$, respectively.

Note that all codewords in Table VI can be considered to be a concatenation of a unary codeword and a suffix codeword. All codewords having the same unary prefix start with a suffix having the same fixed length. Motivated by this, the EGEC, RiceEG and ExpGEC encoders decompose each input symbol into two sub-symbols, which are encoded separately by two distinct sub-encoders, as shown in the schematic of Figure 12. The first sub-encoder is referred to as the UEC sub-encoder, which operates in the same manner as the UEC encoder of this treatise. The second sub-encoder employs a serial concatenation of a Fixed length Code (FLC) and of a Convolutional Code (CC), which is referred to as the FLC-CC sub-encoder. More particularly, the splitter S of Figure 12 decomposes each symbol d_i in the vector $\mathbf{d} = [d_i]_{i=1}^{a}$ into the sub-symbols x_i in the vector $\mathbf{x} = [x_i]_{i=1}^{a}$ and the sub-symbol t_i in the vector $\mathbf{t} = [t_i]_{i=1}^{a}$, according to the arrangement shown in Table VI for the various codes considered. Each sub-symbol x_i and each sub-symbol t_i is encoded by the UEC sub-encoder and the FLC-CC sub-encoder, respectively. An Unequal Error Protection (UEP) scheme was also proposed for optimizing the relative contribution of the two sub-codes to the encoding process, facilitating near-capacity operation at

a low decoder complexity.

Owing to the similarity between the UEC scheme and the UEC sub-part of the EGEC, RiceEG and ExpGEC schemes, they may be readily enhanced by extending our adaptive, irregular and learning-aided techniques of papers [16], [18], [19].

- For the adaptive design of [18], the upper URC code of Figure 12 may be replaced by a turbo code, providing a three-stage concatenation in the UEC sub-part. In this way, the EGEC, RiceEG or ExpGEC decoder can dynamically adjust the activation order of the three decoders in the UEC sub-part, as well as the number of states in the UEC trellis decoder in order to strike an attractive trade-off between the decoding complexity and the error correction capability. Likewise, the FLC-CC part of Figure 12 may be concatenated with a turbo code and the corresponding decoder activation order can be dynamically adapted.
- For the irregular design of [19], the UEC sub-part can provide a fine-grained bit-by-bit basis of the irregularity, rather than on a symbol-by-symbol basis. This allows an IrUEC sub-part to be designed in order to match the concatenated URC code. In the FLC-CC part, a conventional Irregular Convolutional Code (IrCC) may be used to obtain a similar benefit.
- For the learning-aided design of this paper, a memory storage may be employed in the scheme of Figure 12, in analogy to that of Figure 7. Therefore, the occurrence probabilities of the source symbols of the vector $\mathbf{x}$ can be estimated and hence can be fed back to the UEC trellis decoder as *a priori* information, facilitating reliable communication for unknown and non-stationary source probability distributions. Likewise, a similar mechanism may be employed in the FLC-CC part, in analogy to the learning-aided CC code used in the benchmarker of Section VI.

However, the EGEC, RiceEG and ExpGEC schemes have a complicated structure that comprises two parts, as shown in Figure 12. Owing to this, the FLC-CC sub-encoder cannot be operated until the operation of the UEC sub-encoder is completed, since the FLC-CC sub-decoder relies on the side information provided by the UEC sub-decoder. Therefore, the EGEC, RiceEG and ExpGEC schemes may suffer from the delay and loss of synchronization that are associated with the two parts. Furthermore, the UEP of the two parts must be tailored to the specific parametrization of the source distribution, leading to a degraded performance, when the source distribution is unknown or non-stationary. Additionally, the puncturing used in this UEP scheme increases the complexity and may lead to a capacity loss. Motivated by this, the so-called Reordered Elias Gamma Error Correction (REGEC) code was proposed in [105], as a universal JSCC having a simpler structure, as discussed in the following subsection.

B. Adaptive/Irregular/Learning-aided REGEC schemes

As shown in Figure 13, the REGEC code [105] combines a so-called Reordered Elias Gamma (REG) source code with a novel trellis-based channel code. As shown in Table VI, the REG codewords comprise a reordering of the bits in the EG codewords, allowing the REGEC trellis to be designed so that the transitions between its states are synchronous with the transitions between the consecutive codewords in the REG encoded bit sequence. This allows the residual redundancy in the REG encoded-bit sequence to be exploited for error correction by the REGEC trellis decoder, facilitating near-capacity operation. In this way, the REGEC scheme avoids the complicated two-part structure of the EGEC, RiceEG and ExpGEC schemes, as well as dispensing with the associated drawbacks. In particular, the REGEC code also has a simple structure comprising only a single part, which does not suffer from the delay and loss of synchronization that are associated with the two parts of the EGEC, RiceEG and ExpGEC codes. Furthermore, since the REGEC code does not need UEP, its parametrization does not have to be tailored to the particular source distribution, giving a "one size fits all" scheme.

Owing to the similarity between the UEC scheme and the REGEC scheme, our adaptive, irregular and learning techniques may be readily extended to design the corresponding Adaptive-, Irregular- and Learning-aided REGEC schemes.

- In case of the adaptive design of [18], the codebook extension technique is also applicable to the REGEC code, as discussed in [105]. This allows the number of states employed in the REGEC trellis decoder to be dynamically selected, in order to strike an attractive trade-off between the decoding complexity and the error correction capability. Furthermore, the URC code of Figure 13 may be replaced by a turbo code, providing a three-stage concatenation. Therefore, 3D EXIT chart analysis can be employed for quantifying the potential benefits associated with activating each decoder, allowing hence the activation order of the three decoders to be dynamically adjusted.
- In case of the irregular design of [19], the REGEC trellis also exhibits common features, regardless of its parametrization, in the same way as the UEC trellis. This may be exploited to create an irregular REGEC trellis having an irregularity that can be controlled on a fine-grained bit-by-bit basis, rather than on a symbol-by-symbol basis. Moreover, the URC code of Figure 13 may be replaced by an IrURC code, in order to create a narrow, but marginally open EXIT chart tunnel, hence facilitating 'nearer-to-capacity' operation.
- In case of the learning-aided design of this paper, the memory storage that is employed in the scheme of Figure 7 may also be introduced into the REGEC scheme of Figure 13. In this way, the occurrence probabilities of the source symbols can be estimated and then fed back to the trellis decoder as *a priori* information. In this way, the REGEC code would become capable of reliable near-capacity communication for unknown and non-stationary source probability distributions.

VIII. Conclusions

In this paper, we have proposed a novel learning-aided UEC scheme, which is designed for transmitting symbol values

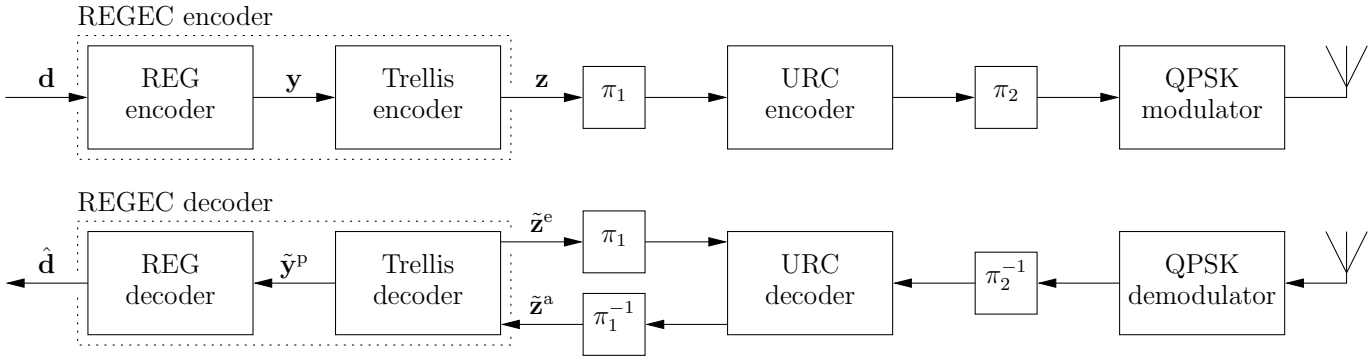


Fig. 13. The schematic of the Reordered Elias Gamma Error Correction (REGEC) code [105].

selected from unknown and non-stationary probability distributions. More specifically, the learning-aided UEC scheme is able to heuristically learn the source symbol distribution, based on the statistics obtained from the recovered symbols that are stored in memory at the receiver. By iteratively feeding these statistics back to the UEC decoder, it can dynamically adjust and maintain near-capacity operation, at the cost of only a slight memory requirement at the receiver. Our simulation results show that our practical learning-aided UEC scheme performs only 0.1 dB away from an idealized but impractical version of the learning-aided UEC scheme, where perfect knowledge of the source distribution available at the receiver. Based on the same learning technique, we also proposed two SSCC benchmarkers, namely a learning-aided EG-CC scheme and a learning-aided Arithmetic-CC scheme. The simulation results show that our learning-aided UEC scheme outperforms the benchmarkers by up to 0.85 dB in a variety of scenarios, without requiring any additional decoding complexity or any additional transmission-energy, -bandwidth, or -duration.

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 Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 1182–1194, March 2011.

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

[3] N. 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 Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 484–490, January 2009.

[4] C. E. Shannon, "The mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[5] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing (3rd Ed)*. New York: Cambridge University Press, 2007.

[6] P. Elias, "Coding for two noisy channels," *IRE Convention Record, pt.4*, pp. 37–47, 1955.

[7] 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/

[8] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 194–203, March 1975.

[9] J. Teuhola, "A compression method for clustered bit-vectors," *Information Processing Letters*, vol. 7, pp. 308–311, October 1978.

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

[11] *High efficiency video coding*. ITU-T Rec. H.265, June 2013.

[12] J. L. Massey, "Joint source and channel coding," in *Proceedings of Communication Systems and Random Process Theory*, The Netherlands: sijthofoo and Noordhoff, December 1978, pp. 279–293.

[13] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 158–162, February 1995.

[14] V. Buttigieg and P. G. Farrell, "Variable-length error-correcting codes," *IEE Proceedings on Communications*, vol. 147, no. 4, pp. 211–215, August 2000.

[15] R. G. Maunder and L. Hanzo, "Near-capacity irregular variable length coding and irregular unity rate coding," *IEEE Transactions on Wireless Communications*, vol. 8, no. 11, pp. 5500–5507, November 2009.

[16] 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 Transactions on Communications*, vol. 6, no. 5, pp. 1977–1987, May 2013.

[17] N. L. Johnson, A. W. Kemp, and S. Kotz, *Univariate Discrete Distributions*. John Wiley and Sons, Inc., Hoboken, NJ, USA, 2005.

[18] W. Zhang, Y. Jia, X. Meng, M. F. Brejza, R. G. Maunder, and L. Hanzo, "Adaptive iterative decoding for expediting the convergence of unary error correction codes," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 621–635, February 2015.

[19] W. Zhang, M. F. Brejza, T. Wang, R. G. Maunder, and L. Hanzo, "An irregular trellis for the near-capacity unary error correction coding of symbol values from an infinite set," *IEEE Transactions on Communications*, vol. 63, no. 12, pp. 5073–5088, December 2015.

[20] A. J. Viterbi and J. Omura, *Principles of Digital Communication and Coding*. New-York: McGraw-Hill, 1979.

[21] L. Hanzo, P. J. Cherriman, and J. Streit, *Video Compression and Communications: H.261, H.263, H.264, MPEG4 and Proprietary Codecs for HSDPA-Style Adaptive Turbo-Transceivers*. John Wiley and IEEE Press, September 2007.

[22] ——, *Video Compression and Communications: H.261, H.263, H.264, MPEG4 and Proprietary Codecs for HSDPA-Style Adaptive Turbo-Transceivers*. John Wiley and IEEE Press, September 2007.

[23] L. Hanzo, F. C. A. Somerville, and J. P. Woodard, *Voice and Audio Compression for Wireless Communications*. John Wiley and IEEE Press, 2007.

[24] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, January 1976.

[25] R. M. Fano, "The transmission of information," in *Research Laboratory of Electronics*, March 1949, pp. Mass. Inst. of Techn. (MIT), Technical Report No. 65.

[26] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, pp. 147–160, April 1950.

[27] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, September 1952.

[28] I. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Professional Group on Information Theory*, vol. 4, no. 4, pp. 38–49, September 1954.

[29] J. Wozencraft, "Sequential decoding for reliable communication," *IRE National Converntion Record*, vol. 5, no. 2, pp. 11–25, August 1957.

[30] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[31] S. W. Golomb, "Run-length encodings," *IEEE Transactions on Information Theory*, vol. 12, pp. 399–401, September 1966.

[32] L. Bahl, C. Cullum, W. Frazer, and F. Jelinek, "An efficient algorithm for computing free distance (corresp.)," *IEEE Transactions on Information Theory*, vol. 18, no. 3, pp. 437–439, May 1972.

[33] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.

[34] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[35] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

[36] J. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76–80, January 1978.

[37] J. Ziv and A. Lempel, "Compression of individual sequences via variable rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, September 1978.

[38] J. Rissanen and G. G. Langdon, "Arithmetic coding," *IBM Journal of Research and Development*, vol. 23, no. 2, pp. 149–162, March 1979.

[39] G. Ungerböck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, January 1982.

[40] D. Divsalar and M. K. Simon, "Multiple trellis coded modulation (MTCM)," *IEEE Transactions on Communications*, vol. 36, no. 4, pp. 410–419, April 1988.

[41] A. R. Calderbank, "Multilevel codes and multistage decoding," *IEEE Transactions on Communications*, vol. 37, no. 3, pp. 222–229, March 1989.

[42] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proceedings of IEEE Global Telecommunications Conference*, vol. 3, November 1989, pp. 1680–1686.

[43] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference [applicable to digital mobile radio receivers]," in *Proceedings of IEEE Global Telecommunications Conference*, vol. 3, December 1990, pp. 1679–1684.

[44] W. T. Webb, L. Hanzo, and R. Steele, "Bandwidth efficient QAM schemes for Rayleigh fading channels," *IEE Proceedings I, Communications, Speech and Vision*, vol. 138, no. 3, pp. 169–175, June 1991.

[45] E. Zehavi, "8-PSK trellis codes for a rayleigh channel," *IEEE Transactions on Communications*, vol. 40, no. 5, pp. 873–884, May 1992.

[46] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proceedings of the International Conference on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[47] Y. Kofman, E. Zehavi, and S. Shamai, "Performance analysis of a multilevel coded modulation system," *IEEE Transactions on Communications*, vol. 42, no. 234, pp. 299–312, February 1994.

[48] S. L. Goff, A. Glavieux, and C. Berrou, "Turbo-codes and high spectral efficiency modulation," in *IEEE International Conference on Communications*, vol. 2, May 1994, pp. 645–649.

[49] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal map decoding algorithms operating in the log domain," in *IEEE International Conference on Communications*, vol. 2, June 1995, pp. 1009–1013.

[50] X. D. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Communications Letters*, vol. 1, no. 6, pp. 169–171, November 1997.

[51] P. Robertson and T. Wörz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 2, pp. 206–218, February 1998.

[52] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[53] M. Luby, "LT codes," in *Proceedings of 43rd Annunal IEEE Symposium Foundations of Computer Science*, Vancouver, BC, Canada, November 2002, pp. 271–280.

[54] J. Hou and M. H. Lee, "Multilevel LDPC codes design for semi-BICM," *IEEE Communications Letters*, vol. 8, no. 11, pp. 674–676, November 2004.

[55] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, pp. 2551–2567, June 2006.

[56] G. Yue, B. Lu, and X. Wang, "Design of rate-compatible irregular repeat accumulate codes," *IEEE Transactions on Communications*, vol. 55, no. 6, pp. 1153–1163, June 2007.

[57] E. Arikan, "Channel polarization: A method for constructing capacityachieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, July 2009.

[58] Z. Wang and J. Luo, "Error performance of channel coding in random-access communication," *IEEE Transactions on Information Theory*, vol. 58, no. 6, pp. 3961–3974, June 2012.

[59] J. Luo, "A generalized channel coding theory for distributed communication," *IEEE Transactions on Communications*, vol. 63, no. 4, pp. 1043–1056, April 2015.

[60] V. A. Kotelnikov, *The Theory of Optimum Noise Immunity*. New York, NY, USA: McGraw-Hill, 1959.

[61] A. Kurtenbach and P. Wintz, "Quantizing for noisy channels," *IEEE Transactions on Communications*, vol. 17, no. 2, pp. 291–302, April 1969.

[62] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, January 1980.

[63] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electronics and Engineering in Japan*, vol. 67-B, no. 4, pp. 39–47, January 1984.

[64] B. L. Montgomery and J. Abrahams, "Synchronization of binary source codes," *IEEE Transactions on Information Theory*, vol. 32, no. 6, pp. 849–854, November 1986.

[65] N. Farvardin and V. Vaishampayan, "Optimal quantizer design for noisy channels: An approach to combined source-channel coding," *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 827–838, November 1987.

[66] R. R. Wyrwas and P. G. Farrell, "Joint source-channel coding for raster document transmission over mobile radio," *IEE Proceedings I on Communications, Speech and Vision*, vol. 136, no. 6, pp. 375–380, December 1989.

[67] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Transactions on Information Theory*, vol. 36, no. 4, pp. 799–809, July 1990.

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

[69] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution broadcast for digital HDTV using joint source/channel coding," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 1, pp. 6–23, January 1993.

[70] I. Kozintsev and K. Ramchandran, "Robust image transmission over energy-constrained time-varying channels using multiresolution joint source-channel coding," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 1012–1026, April 1998.

[71] R. E. V. Dyck and D. J. Miller, "Transport of wireless video using separate, concatenated, and joint source-channel coding," *in Proceedings of the IEEE*, vol. 87, no. 10, pp. 1734–1750, October 1999.

[72] J. f. Cai and C.-W. Chen, "Robust joint source-channel coding for image transmission over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 962–966, September 2000.

[73] N. Görtz, "A generalized framework for iterative source-channel decoding," *Annals of Telecommunications, Special issue on "Turbo Codes: a wide-spreading technique"*, pp. 435–446, July 2001.

[74] V. B. Balakirsky, "Joint source-channel coding using variable-length codes," *Proceedings of IEEE International Symposium on Information Theory (ISIT'97)*, p. 419, January 1997.

[75] T. A. Ramstad, "Shannon mappings for robust communication," *Telektronikk*, vol. 98, no. 1, pp. 114–128, 2002.

[76] J. Hagenauer and N. Görtz, "The turbo principle in joint source-channel coding," in *Proceedings of IEEE Information Theory Workshop*, Paris, France, March 2003, pp. 275–278.

[77] J. Kliewer and R. Thobaben, "Iterative joint source-channel decoding of variable-length codes using residual source redundancy," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 919–929, May 2005.

[78] R. Thobaben and J. Kliewer, "Design considerations for iteratively-decoded source-channel coding schemes," in *Proceedings of Allerton Conference on Communications, Control, and Computing*, Monticello, IL, USA, September 2006.

[79] X. Jaspar, C. Guillemot, and L. Vandendorpe, "Joint source-channel turbo techniques for discrete-valued sources: From theory to practice," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1345–1361, June 2007.

[80] Q. Xu, V. Stankovic, and Z.-X. Xiong, "Distributed joint source-channel coding of video using raptor codes," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 851–861, May 2007.

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

[82] P. Minero, S. Lim, and Y.-H. Kim, "Joint source-channel coding via hybrid coding," *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pp. 781–785, July 2011.

[83] D. Persson, J. Kron, M. Skoglund, and E. G. Larsson, "Joint source-channel coding for the mimo broadcast channel," *IEEE Transactions on Signal Processing*, vol. 60, no. 4, pp. 2085–2090, April 2012.

[84] V. Kostina and S. Verdu, "Lossy joint source-channel coding in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2545–2575, May 2013.

[85] S. M. Romero, M. Hassanin, J. Garcia-Frias, and G. R. Arce, "Analog joint source channel coding for wireless optical communications and image transmission," *Journal of Lightwave Technology*, vol. 32, no. 9, pp. 1654–1662, May 2014.

[86] S. Tridenski, R. Zamir, and A. Ingber, "The Ziv-Zakai-Rényi bound for joint source-channel coding," *IEEE Transactions on Information Theory*, vol. 61, no. 8, pp. 4293–4315, August 2015.

[87] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, March 1960.

[88] J. Max, "Quantizing for minimum distortion," *IRE Transactions on Information Theory*, vol. 6, no. 1, pp. 7–12, March 1960.

[89] D. J. Miller and M. Park, "A sequence-based approximate mmse decoder for source coding over noisy channels using discrete hidden markov models," *IEEE Transactions on Communications*, vol. 46, no. 2, pp. 222–231, February 1998.

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

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

[92] R. Bauer and J. Hagenauer, "Symbol by symbol MAP decoding of variable length codes," in *Proceedings of ITG Conference on Source and Channel Coding*, Munich, Germany, January 2000, pp. 111–116.

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

[94] C. Weidmann, "Reduced-complexity soft-in-soft-out decoding of variable-length codes," in *Proceedings of IEEE International Symposium on Information Theory*, Yokohama, Japan, June 2003, p. 201.

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

[96] 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 Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1327–1336, April 2007.

[97] R. G. Gallager and D. C. V. Voorhis, "Optimal source codes for geometrically distributed integer alphabets," *IEEE Transactions on Information Theory*, vol. 21, no. 2, pp. 228–230, March 1975.

[98] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate(corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[99] M. Wien, *High Efficiency Video Coding - Coding Tools and Specification*. Berlin, Heidelberg: Springer, 2014.

[100] 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," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2795–2800, April 2013.

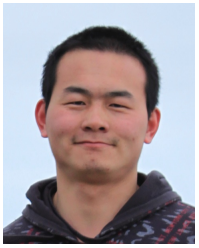
[101] A. Ashikhmin, G. Kramer, and S. ten Brink, "Code rate and the area under extrinsic information transfer curves," in *Proceedings of IEEE International Symposium on Information Theory*, Lausanne, Switzerland, June 2002, p. 115.

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

[103] M. F. Brejza, T. Wang, W. Zhang, D. A. Khalili, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, "Exponential golomb and rice error correction codes for near-capacity joint source and channel coding," *to be submitted*, December 2015.

[104] 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 Transactions on Communications*, vol. 62, no. 1, pp. 280–292, January 2014.

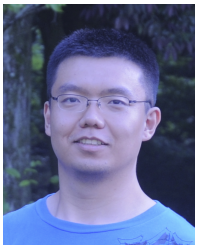
[105] T. Wang, W. Zhang, M. F. Brejza, R. G. Maunder, and L. Hanzo, "Reordered elias gamma error correction codes for the near-capacity joint source and channel coding of multimedia information," *IEEE Transactions on Vehicular Technology, submitted*, February 2016.



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.

Zeyu Song received a first class honours BEng in Electronic Engineering from the University of Southampton, UK, in July 2014, and the M.Sc. in Communications and Signal Processing from Imperial College London, UK, in November 2015. He is currently working for China Telecommunications Corporation Beijing Branch in Network Operations and Maintenance group.

Matthew F. Brejza (http://users.ecs.soton.ac.uk/mfb2g09) received a first class honors BEng in Electronic Engineering from the University of Southampton, UK, in July 2012, where he is currently working toward the Ph.D. degree with the Communications Research Group, School of Electronics and Computer Science. 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, 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.

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.

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 an honorary doctorate by the Technical University of Budapest, while in 2015 by the University of Edinburgh. During his 38-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 about 100 PhD students, co-authored 20 John Wiley/IEEE Press books on mobile radio communications totalling in excess of 10 000 pages, published 1500+ 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 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) UK, the European Research Council's Advanced Fellow Grant and the Royal Society's Wolfson Research Merit Award. 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. Lajos has 22 000+ citations. For further information on research in progress and associated publications please refer to http://www-mobile.ecs.soton.ac.uk