

Joint Iterative-Detection of Reversible Variable-Length Coded Constant Bit Rate Vector-Quantized Video and Coded Modulation

S. X. Ng, R. G. Maunder, J. Wang, L-L. Yang and L. Hanzo

School of ECS, University of Southampton, SO17 1BJ, UK.

Email: {sxn,rm02r,jw02r,lly,lh}@ecs.soton.ac.uk, <http://www-mobile.ecs.soton.ac.uk>

Abstract – Joint video-coding, channel-coding and modulation schemes based on a Constant Bit Rate (CBR) video codec, Variable Length Codes (VLCs) as well as Trellis Coded Modulation (TCM) and Turbo TCM (TTCM) schemes are proposed. These arrangements have a latency of a single video frame duration. A significant coding gain is achieved without bandwidth expansion with the advent of iterative decoding exchanging extrinsic information between the VLC and the TCM or TTCM decoders. The performance of the proposed schemes was evaluated for transmission over uncorrelated Rayleigh fading channels and the best scheme was found to be about 3 dB from the Rayleigh channel’s capacity limit.

1. INTRODUCTION

Shannon has shown, under a number of idealistic assumptions [1], that lossless source and channel coding are best carried out in isolation, when communicating over Gaussian channels inflicting uncorrelated, random errors. However, in practice many of Shannon’s assumptions have to be discarded when considering interactive video telephony in the context of hostile wireless channels, as it was argued in [2, Section 1.14]. For example, the source-coding delay as well as the channel- and turbo-interleaving latency has to be low for the sake of ‘lip-synchronisation’. Secondly, turbo codecs and Low Density Parity Check (LDPC) codecs are capable of operating near the Shannon-limit over Gaussian channels, but the cut-off rate of dispersive fading channels is unknown and certainly hard to approach. Thirdly, practical video codecs are ‘lossy’, exploiting the psycho-visual imperfections of the human eye, and, in contrast to ideal lossless codecs, their bits exhibit different sensitivities to transmission errors. All these aspects are discussed in more detail in [2]. Therefore, a combined strategy, where the source and channel codecs are combined, may yield a better performance than Shannon’s separate source and channel coding approach designed for perfectly lossless codecs operating over Gaussian channels.

Lossless Variable Length Codes (VLCs) [2] constitute a family of low-complexity source compression schemes. In order to exploit the residual redundancy of VLCs, numerous trellis-based VLC decoding techniques have been proposed, such as the joint source/channel coding scheme of [3], where the VLC decoder uses the bit-based rather than VLC-symbol based trellis structure of [4]. In order to improve the bandwidth and power efficiency of the joint source/channel coding scheme contrived in [3], an amalgamated source-coding, channel-coding and modulation scheme was proposed in [5], where bandwidth efficient Coded Modulation (CM) schemes were employed as the inner constituent code.

Trellis Coded Modulation (TCM) [6, 7] and Turbo TCM [7, 8] constitute bandwidth-efficient joint channel coding and modulation schemes, which were originally designed for transmission over Additive White Gaussian Noise (AWGN) channels. In an effort to

improve the performance of TCM when communicating over Rayleigh fading channels, In-phase (I) and Quadrature-phase (Q) interleaved TCM (IQ-TCM) schemes were proposed in [9, 10]. Specifically, IQ-TCM benefits from additional signal-space diversity or IQ-diversity, owing to the independent fading of the I and Q components, when communicating over Rayleigh fading channels. Furthermore, IQ-interleaved TTCM [10] also benefits from the IQ-diversity.

In this contribution, a CBR video codec [2] is integrated with the IQ-TCM-VLC and IQ-TTCM-VLC schemes of [5], upon invoking the IQ-TCM and IQ-TTCM arrangements of [10] as the inner constituent code.

2. SYSTEM OVERVIEW

The block diagram of the proposed scheme is shown in Figure 1, which incorporates the CBR video encoder of Figure 2 supplying the encoded video frames \mathbf{u} . This video codec includes a ‘frame size analyzer’ for ensuring that each transmission frame represents exactly one video frame, as described later in Section 2.1. The block diagram of the iterative CM-VLC decoder was given in [5, Figure 1]. The output of the iterative CM-VLC decoder, which represents estimates of the video symbols $\hat{\mathbf{v}}$, is passed to the video decoder, which performs the inverse functions of the video encoder seen in Figure 2.

In the proposed scheme, we employ $M = 16$ -level Quadrature Amplitude Modulation (16QAM) for conveying $\log_2(M) = 4$ bits per transmitted 16QAM symbol. The 16QAM-based CM schemes employed both IQ-TCM [10] having a code memory of $L = 6$ as well as an iterative IQ-TTCM [10] arrangement having $L = 3$ using four decoding iterations. These CM parameters were chosen for the sake of maintaining a similar decoding complexity, since the total number of trellis-stages was $2^6 = 64$ and $2^3 \cdot 2 \cdot 4 = 64$, respectively [10]. The code generators for IQ-TCM and IQ-TTCM are [101 16 64 0] and [11 2 4 10] in octal format, respectively. The coding rate of the CM schemes is given by $R_{cm} = \frac{N_s - \hat{L}}{N_s} \cdot \frac{\log_2(M) - 1}{\log_2(M)}$,

where we have $\hat{L} = L$ for the TCM scheme, since the decoding trellis is terminated using L symbols. By contrast, we have $\hat{L} = 2L$ for the TTCM scheme, since the decoding trellises of both of the constituent TCM codes are terminated [7]. Furthermore, the transmission frame length was set to $N_s = 508$ 16QAM symbols, which is appropriate for the greyscale representation of a single (176×144) -pixel Quarter Common Intermediate Format (QCIF) video frame. As a result, the coding rate of TCM/TTCM becomes $R_{cm} = 0.7411$. Specifically, the total number of bits input to the CM encoder is $L_{cm}^{input} = (N_s - \hat{L})(\log_2(M) - 1) = 1506$ bits. A total of $L_{cm}^{output} = N_s \log_2(M) = 2032$ bits are transmitted to represent each video frame in this scheme. At a video frame rate of 10 frames/s, the total transmission rate is 20.32 kbps, which is similar to that of the schemes described in [2, Table 13.3]. With the advent of the proposed turbo transceiver substantial E_b/N_0 gains were achieved in comparison to [2, Table 13.3].

The financial support of the EPSRC, Swindon UK is gratefully acknowledged.

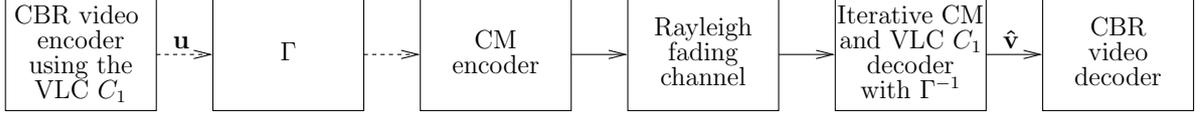


Figure 1: Block diagram of the proposed CM-VLC scheme incorporating a CBR video codec. The solid lines indicate multi-bit symbols, the dashed lines indicate bit strings, while \mathbf{u} represents the coded video frame and $\hat{\mathbf{v}}$ represents the vector of video symbol estimates. Finally, Γ represents the process of adding side information and bit-interleaving, while Γ^{-1} represents the inverse operation.

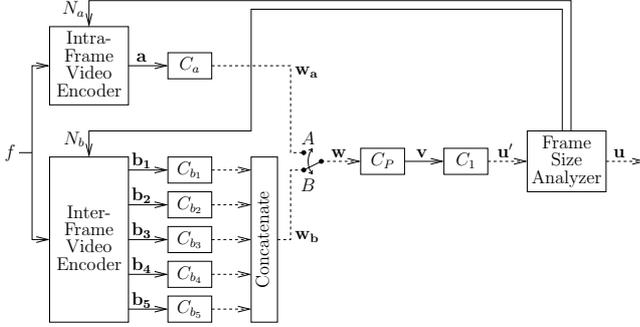


Figure 2: The CBR video encoder of Figure 1. The solid lines indicate symbols, while the dashed lines indicate bit strings.

Within the CBR video encoding scheme of Figure 2, 4-bit video symbols \mathbf{v} were created for the sake of generating a uniform transmission format, which were encoded using a VLC code C_1 to form the bit string \mathbf{u} . The VLC C_1 was designed based on the probability of occurrence of the $I = 16$ possible values of the video symbols in \mathbf{v} , as recorded during the encoding of five different training video sequences. The video symbols were observed to be fairly equiprobable, with an entropy of $\mathcal{L}_{C_1}^E = 3.9987$, which is only marginally lower than $\log_2(I) = 4$. There are numerous different VLC designs in the literature [4, 11]. Some VLC designs aim for minimising the average codeword length, whilst others opt for increasing the code's free distance. We found that symmetrical reversible VLC's [11] having a free distance of $d = 2$ strike a good compromise between the average codeword length and a high error resilience. Hence, C_1 was designed on this basis, yielding the codes $\{1001, 01110, 010010, 100001, 0110, 0111110, 10001, 1111, 00100, 100001, 110011, 011110, 11011, 101101, 001100, 0000\}$. The average code length of C_1 was $\mathcal{L}_{C_1}^A = 5.3372$ and the resultant coding rate was $R_{vlc} = \mathcal{L}_{C_1}^E / \mathcal{L}_{C_1}^A = 0.7492$.

Amongst the numerous trellis-based VLC decoding techniques found in the literature, the bit-based rather than VLC-labelled trellis structure of [4] is relatively simple and time-invariant. The only side-information required during the bit-based trellis decoding operation is the number of VLC coded bits in \mathbf{u} [3, 4], which is denoted as L_{bit} .

Although the CBR video encoder includes the 'frame size analyzer' for ensuring that \mathbf{v} contains as many video symbols as possible, in general, the VLC encoder will be unable to generate exactly the required number of bits. This is a consequence of the variable lengths of the codes in C_1 . Hence it may be necessary to concatenate some dummy bits so that the length of \mathbf{u} becomes a constant. The maximum codeword length of C_1 is $l_{max} = 7$ and hence the number of dummy bits, L_{dummy} , required is in the range of $[L_{dummy}^{min} = 0, L_{dummy}^{max} = (l_{max} - 1)]$. At the receiver, the number of VLC encoded bits in a transmission frame, L_{bit} , can be calculated with the knowledge of the value of L_{dummy} . Therefore, we only have to convey the information regarding L_{dummy} , which requires a lower number of bits compared to L_{bit} . A total of $\lceil \log_2(l_{max}) \rceil$ bits are used for representing L_{dummy} . In an effort to render our investigations as realistic as possible, the trans-

Symbol set	Possible values	Entropy \mathcal{L}^E	Average code length \mathcal{L}^A	Occurrences per frame \mathcal{N}	Information represented
\mathbf{a}	16	3.48	3.52	N_a^2	Intra-frame PFU
\mathbf{b}_1	16	3.5	3.53	30	A/P MB table
\mathbf{b}_2	16	3.61	3.63	~ 65	A/P SMB table
\mathbf{b}_3	16	3.02	3.04	60	Motion vector
\mathbf{b}_4	16	3.94	3.97	57	VQ entry
\mathbf{b}_5	256	7.01	7.04	N_b	

Table 1: Symbols generated during intra- and inter-frame encoding.

mission of the side information representing L_{dummy} is repeated three times for the sake of majority logic based detection. These bits are then further protected by the CM scheme. Hence, the total number of bits required for conveying the side information is $L_{side} = 3 \lceil \log_2(7) \rceil = 9$ bits.

The total number of bits at the input of the CM encoder amounts to $L_{cm}^{input} = L_{bit} + L_{dummy} + L_{side}$. We also know that we have $L_{cm}^{input} = 1506$ bits, therefore, the number of VLC output bits becomes $L_{bit} = 1506 - 9 - L_{dummy}$, which is in the range of $[1491, 1497]$ bits. We can represent the throughput loss imposed by the side-information upon introducing a coding rate term of $R_{side} = \frac{L_{cm}^{input} - L_{side}}{L_{cm}^{input}} = 0.9940$. Let us now describe the video system in the following section.

2.1. Video System Overview

In the proposed scheme, the CBR video encoder of [2, Chapter 13] is used, which was slightly modified by incorporating the 'frame size analyzer' for generating the encoded video frame \mathbf{u} of suitable length, as illustrated in Figure 2. Again, each transmission frame conveys exactly one video frame. This one-to-one mapping ensures that latency is limited to a single video frame duration and that video frame synchronisation may be readily reestablished in the presence of transmission errors.

The video encoder combines intra- and inter-frame encoding modes of operation. These are selected by toggling the switch A/B , as shown in Figure 2, to position A or B respectively.

Inter-frame encoding achieves high compression by encoding the Motion Compensated Error Residual (MCER) between the current and previous frames. Partial Forced Update (PFU) [2, Section 12.8] and gain-cost control [2, Section 12.6] of Motion Compensation (MC) [2, Section 12.2] as well as Vector Quantization (VQ) [2, Section 13.3] are employed. The inter-frame encoding mode operates on (8×8) -pixel blocks. Both MC and VQ are applied only to a limited number of so-called *active blocks*. A motion vector symbol is generated for each *MC-active block*, while a codebook-entry symbol is generated for each *VQ-active block*. The positions of active blocks are encoded using active/passive (A/P) tables, where blocks are grouped into the so-called Macro-Blocks (MB), which are in turn grouped into Super-Macro-Blocks (SMB) [2, Section 12.7]. The A/P tables are represented by two classes of symbols. A variable number of symbols is used for conveying the A/P status of blocks in only the active MBs, which are those that contain at least one active block. By contrast, a fixed number of symbols is used for conveying the A/P status of a MB in every SMB. The number of PFU and MC-active blocks is fixed for each video frame. By

contrast, the number VQ-active blocks employed is determined by the parameter N_b , which is controlled for the sake of adjusting the number of video symbols generated during inter-frame encoding.

However, inter-frame encoding is unable to represent the first frame of a video sequence, since in this case there is no previous frame for generating the MCER. For this reason, **intra-frame encoding** must be employed to represent the first frame of a video sequence. This is achieved by dividing it into $N_a \times N_a$ perfectly tiling blocks, where N_a is a parameter of the intra-frame encoder. Each block is crudely represented by a single symbol, encoding only its mean luminance [2, Section 12.5].

As seen in Figure 2, intra-frame encoding generates a single set of symbols \mathbf{a} , while inter-frame encoding generates five different sets of symbols $[\mathbf{b}_1 \dots \mathbf{b}_5]$. These symbols are described in Table 1, including the number of possible values for each. Furthermore the number of occurrences for each of the different video coded symbol types per frame are also provided. The bit string representations of the symbols \mathbf{a} and $[\mathbf{b}_1 \dots \mathbf{b}_5]$ are generated using appropriately designed Huffman encoders C_a and $[C_{b_1} \dots C_{b_5}]$, using the symbol occurrence probabilities recorded during the encoding of the training video sequences.

During the inter-frame encoding operation, the five different bit string representations of the symbol sets $[\mathbf{b}_1 \dots \mathbf{b}_5]$ are concatenated in order to form a single bit string \mathbf{w}_b , as shown in Figure 2. Similarly, \mathbf{w}_a of Figure 2 is assembled in case of intra-frame coding. The ‘frame size analyzer’ of Figure 2 and an iterative bit-packing procedure is employed for generating each transmission frame. In each iteration, the bit string selected by the switch A/B , \mathbf{w} , is packetized using a fixed-length encoder C_P , where one of $I = 16$ symbol values is selected for each 4-bit section. Zero-padding is employed, if \mathbf{w} is not exactly divisible by 4. The set of these video symbols is labelled \mathbf{v} in Figure 2. These are encoded using the VLC C_1 in order to form the coded video frame \mathbf{u}' and the ‘frame size analyzer’ is employed for adjusting the values of N_a and N_b , in order to generate the required number of bits. These N_a and N_b values result in a certain ‘buffer-fullness’, hence providing feedback for the next iteration. The employment of Huffman encoders results in \mathbf{w} of Figure 2, where \mathbf{w} has nearly-equiprobable bit values and hence accounts for the near-uniform probability distribution of C_1 , as noted above. This iterative re-encoding and bit-packing process continues until the maximal value of N_a or N_b , as appropriate, that yields a coded video frame \mathbf{u}' containing no more than $L_{cm}^{input} - L_{side} = 1497$ bits is found. The most typical final values of N_a and N_b were recorded as 17 and 52, respectively.

In the vast majority of cases, the inter-frame encoding mode of operation is used. The following calculations are made based on the stated most typical value of N_b and on the properties of the symbols $[\mathbf{b}_1, \mathbf{b}_5]$ to reflect this. The effective video-VLC encoding rate can be computed as: $R_{video} = \frac{\sum_{i=1}^5 \mathcal{L}_{C_{b_i}}^E \mathcal{N}_{b_i}}{\sum_{i=1}^5 \mathcal{L}_{b_i}^A \mathcal{N}_{b_i}} = \frac{1110}{1117} = 0.994$. The average length of \mathbf{w} is the denominator of the R_{video} calculation above, which is 1117 bits. The average length of \mathbf{u}' can be calculated as $L_{video} = \frac{1117}{\log_2(I)} \times \mathcal{L}_{C_1}^A = 1490$ bits.

If necessary, the ‘frame size analyzer’ will append the minimum required number of dummy C_1 symbols, as distinct from dummy bits, to \mathbf{u}' for increasing its length to a minimum of $L_{cm}^{input} - L_{side} - L_{dummy}^{max} = 1491$ bits. Additionally, the ‘frame size analyzer’ appends dummy bits, further increasing the length of \mathbf{u}' to $L_{cm}^{input} - L_{side} = 1497$ bits. Dummy-symbols and -bits account for an average of $L_{cm}^{input} - L_{side} - L_{video} = 7$ bits per frame during inter-frame encoding. The corresponding loss of throughput is represented by the coding rate term of $R_{dummy} = \frac{L_{video}}{L_{cm}^{input} - L_{side}} = 0.9953$. Following these operations, \mathbf{u}' is latched by the frame size analyzer as the final transmission frame \mathbf{u} .

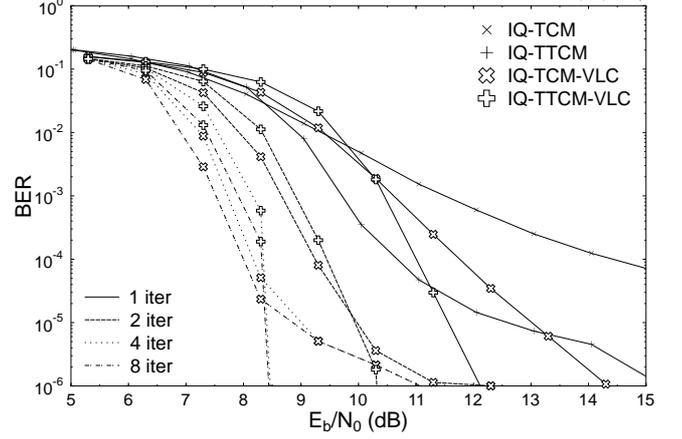


Figure 3: BER versus E_b/N_0 performance of the proposed 16QAM-based (IQ-)CM-VLC schemes, when communicating over narrowband, uncorrelated Rayleigh fading channels.

At the receiver the video symbol estimates $\hat{\mathbf{v}}$ supplied by the joint source-channel decoder are de-packetized using the inverse of C_P and intra- or inter-frame decoded, as appropriate. During intra-frame decoding, the specific value of N_a used by the encoder is inferred by counting the number of symbols in $\hat{\mathbf{v}}$. By contrast, during inter-frame decoding, the number of A/P SMB symbols in the transmission frame can be inferred from the contents of the decoded MB symbols. Likewise, the number of VQ active blocks can be inferred from the decoded A/P tables.

3. SIMULATION RESULTS

The effective coding rate of the proposed system can be computed from $R_{eff} = R_{video} R_{vlc} R_{dummy} R_{side} R_{cm} = 0.9940 \times 0.7492 \times 0.9953 \times 0.9940 \times 0.7411 = 0.5460$. The bandwidth efficiency of the 16QAM system becomes $\eta = R_{eff} \log_2(M) = 2.184$ bit/s/Hz, when ideal Nyquist filtering is assumed. We will also use the IQ-TCM and IQ-TTCM schemes as benchmarkers, which take the video symbols \mathbf{v} as the input directly from the CBR video codec of Figure 2 without invoking the VLC of C_1 . In these simplified benchmarker schemes, the effective coding rate is $R_{eff} = R_{video} R_s R_{cm}$, where $R_s = \mathcal{L}_{C_1}^E / \log_2(I) = 0.9997$. Hence we have $R_{eff} = 0.7364$ and $\eta = 2.946$ bit/s/Hz. Note that at $\eta = 2.184$ and 2.946 bit/s/Hz, the uncorrelated Rayleigh fading channel capacity limit for 16QAM assuming equiprobable occurrence for the 16QAM symbols is $E_b/N_0 = 4.47$ dB and 7.30 dB, respectively¹, where $E_b/N_0 = \text{SNR}/\eta$ is the Signal to Noise Ratio (SNR) per bit.

Figure 3 depicts the Bit Error Ratio (BER) versus E_b/N_0 performance of the proposed schemes, when communicating over narrowband, uncorrelated Rayleigh fading channels. As shown in Figure 3, at a BER of 10^{-4} the IQ-TCM-VLC scheme managed to achieve an additional 2.5 dB gain, when the number of outer iterations was increased from one to two. Explicitly, one outer iteration is constituted by one CM decoding and one VLC decoding operation. However, as the number of outer iterations is increased, the achievable incremental iteration gain reduces. At a BER of 10^{-4} the proposed IQ-TCM-VLC scheme having eight outer iterations managed to outperform the IQ-TCM benchmarker by 6.4 dB. On the other hand, IQ-TTCM-VLC exhibits a steeper BER performance curve than that of the IQ-TCM-VLC arrangement. Note that during the first outer iteration, IQ-TTCM-VLC

¹These values were computed based on [6, 12].

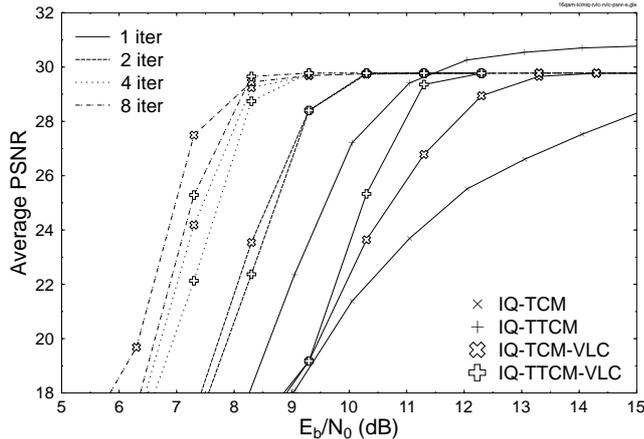


Figure 4: Average PSNR versus E_b/N_0 performance of the proposed 16QAM-based (IQ-)CM-VLC schemes, when communicating over narrowband, uncorrelated Rayleigh fading channels using the ‘Lab-sequence’ video clip [2, Fig.13.5].

outperforms IQ-TTCM-VLC at BERs below 2×10^{-3} . However, the crossover point between the BER curves of IQ-TTCM-VLC and IQ-TTCM-VLC occurs only at progressively lower BERs, as the number of outer iterations increases. Therefore, IQ-TTCM-VLC is only preferable compared to IQ-TTCM-VLC, when a very low BER is required or when the affordable number of outer iterations is limited. When comparing the performance of IQ-TTCM-VLC and IQ-TTCM, IQ-TTCM-VLC using one outer iteration fares worse than the IQ-TTCM benchmark scheme for BERs $> 4 \times 10^{-5}$ after shifting the curves according to their different effective throughput or bandwidth efficiency values on the E_b/N_0 scale. Nonetheless, after the second outer iteration, the IQ-TTCM-VLC scheme outperforms the IQ-TTCM benchmark.

Figure 4 illustrates the average Peak Signal to Noise Ratio (PSNR) [2, Equation 12.8] versus E_b/N_0 performance of the proposed schemes, when communicating over narrowband uncorrelated Rayleigh fading channels using the ‘Lab-sequence’ video clip [2, Fig.13.5]. The maximum attainable average PSNRs for IQ-CM-VLC and IQ-CM are 29.8 dB and 30.8 dB, respectively, owing to their different video source rates. At an average PSNR of 29 dB associated with near-unimpaired video quality, a total iteration gain of 4.3 dB is attained by the IQ-TTCM-VLC scheme, when the number of outer iterations was increased from one to four. Interestingly, IQ-TTCM-VLC only outperforms IQ-TTCM-VLC at the first outer iteration. The proposed IQ-TTCM-VLC scheme outperforms the IQ-TTCM benchmark for all iterations in terms of its average PSNR performance. Explicitly, after the eighth outer iteration, the IQ-TTCM-VLC scheme provides a 7 dB E_b/N_0 gain over that IQ-TTCM benchmark. Furthermore, the IQ-TTCM-VLC scheme using four outer iterations outperforms the IQ-TTCM benchmark by approximately 2.2 dB at the cost of four times higher detection complexity.

By comparing Figures 3 and 4, we observe that a BER of 10^{-3} is sufficiently low for attaining a near un-impaired video quality associated with average PSNRs higher than 28 dB. Explicitly, at a BER of 10^{-3} , IQ-TTCM-VLC using eight outer iterations is about 3.03 dB away from its channel capacity limit of 4.47 dB. By contrast, IQ-TTCM is 4.2 dB away from the corresponding channel capacity limit of 7.30 dB. Furthermore, when the number of outer iterations is higher than or equal to four, the IQ-TTCM-VLC scheme is capable of providing a BER of lower than 10^{-6} at an E_b/N_0 value around 8.5 dB, which is about 4.03 dB from its channel capacity limit of 4.47 dB. By contrast, the IQ-TTCM benchmark is 8.3 dB away from the associated channel capacity limit of 7.30 dB

at a BER of 10^{-5} . Therefore, the performance of the proposed IQ-CM-VLC scheme is closer to the relevant channel capacity limit compared to that of the IQ-CM benchmarks. Finally, note that the channel cutoff rate of 16QAM [12] at $\eta = 2.184$ is $E_b/N_0 = 8.78$ dB. Hence, the IQ-TTCM-VLC and IQ-TTCM-VLC performs better than the corresponding channel cutoff rate by 1.28 dB (at BER= 10^{-3}) and 0.28 dB (at BER= 10^{-6}), respectively.

4. CONCLUSIONS

In this contribution the concept of amalgamated source-coding, channel-coding and modulation was implemented in the context of a realistic CBR video system, where the transmission frame length was fixed to 2032 bits and hence the latency of the system was limited to 100 ms for enabling real-time interactive video communications over narrowband, uncorrelated Rayleigh channels.

With the advent of bandwidth-efficient coded modulation schemes as well as with the aid of IQ-interleaving, the performance of the proposed joint video-coding, channel-coding and modulation schemes was found to be within 3 to 5 dBs from the corresponding channel capacity limit but better than the channel’s cutoff rate. Our future research will be aiming to design schemes which exploit sources of extrinsic information within the video codec and to design similar MPEG4 and H.264 based systems.

5. REFERENCES

- [1] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell system Technical Journal*, vol. 27, pp. 379–656, July 1948.
- [2] L. Hanzo, P.J. Cherriman and J. Street, *Wireless Video Communications: Second to Third Generation Systems and Beyond*. NJ, USA : IEEE Press., 2001.
- [3] R. Bauer, J. Hagenauer, “On Variable Length Codes for Iterative Source/Channel Decoding,” in *IEEE Data Compression Conference*, (UT, USA), pp. 273–282, 27–29 March 2001.
- [4] V. B. Balakirsky, “Joint Source-Channel Coding with Variable Length Codes,” in *IEEE International Symposium on Information Theory*, (Ulm, Germany), p. 419, 29 June – 4 July 1997.
- [5] S. X. Ng, F. Guo, J. Wang, L-L. Yang and L. Hanzo, “Joint Source-coding, Channel-coding and Modulation schemes for AWGN and Rayleigh Fading Channels,” *IEE Electronics Letters*, vol. 39, pp. 1259–1261, August 2003.
- [6] G. Ungerböck, “Channel Coding with Multilevel/Phase Signals,” *IEEE Transactions on Information Theory*, vol. 28, pp. 55–67, January 1982.
- [7] L. Hanzo, T. H. Liew and B. L. Yeap, *Turbo Coding, Turbo Equalisation and Space Time Coding for Transmission over Wireless channels*. New York, USA: John Wiley IEEE Press, 2002.
- [8] P. Robertson, T. Wörz, “Bandwidth-Efficient Turbo Trellis-Coded Modulation Using Punctured Component Codes,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 206–218, February 1998.
- [9] B. D. Jelicic and S. Roy, “Design of Trellis Coded QAM for Flat Fading and AWGN Channels,” *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 192–201, February 1994.
- [10] S. X. Ng and L. Hanzo, “Space-Time IQ-interleaved TCM and TTCM for AWGN and Rayleigh Fading Channels,” *IEE Electronics Letters*, vol. 38, pp. 1553–1555, November 2002.
- [11] V. Buttigieg and P. G. Farrell, “Variable-Length Error-Correcting Codes,” *IEE Proceedings of Communications*, vol. 147, pp. 211–215, August 2000.
- [12] G. Caire, G. Taricco and E. Biglieri, “Bit-Interleaved Coded Modulation,” *IEEE Transactions on Information Theory*, vol. IT-44, pp. 927–946, May 1998.