Distributed Source–Channel Coding Using Reduced-Complexity Syndrome-Based TTCM

Abdullah Aljohani, Zunaira Babar, Soon Xin Ng, and Lajos Hanzo

***Abstract*— In the context of distributed joint source–channel coding, we conceive reduced-complexity turbo trellis coded modulation (TTCM)-aided syndrome-based block decoding for estimating the cross-over probability p_e of the binary symmetric channel, which models the correlation between a pair of sources. Our joint decoder achieves an accurate correlation estimation for varying correlation coefficients at 3 dB lower SNR, than conventional TTCM decoder, despite its considerable complexity reduction.**

***Index Terms*— Distributed joint source-channel coding, Slepian-Wolf coding, distributed source coding, turbo trellis coded modulation, syndrome decoding.**

I. Introduction

DISTRIBUTED Source Coding (DSC) refers to the problem of compressing several physically separated, but correlated sources, which are unable to communicate with each other by exploiting that the receiver can perform joint decoding of the encoded signals [1]. The Slepian-Wolf (SW) theorem [2] has laid down the theoretical foundations of DSC through specifying the compression rate regions for noiseless channel transmission. However, Distributed Joint Source-Channel coding (DJSC) is specific to the practical case, when the correlated source signals are transmitted over noisy channels.

Since the correlation between the sources may be interpreted as the ameliorating effect of a "virtual" error channel, powerful channel codes such as Low-Density Parity-Check (LDPC) codes and turbo codes are capable of achieving a significant performance enhancement in DJSC design [1], [3]. Recently the bandwidth-efficient Turbo Trellis Coded Modulation (TTCM) was invoked in [4], where the parity bits are absorbed without any bandwidth expansion by doubling the number of constellation points. An error trellis-based Block Syndrome Decoder (BSD) (TTCM-BSD) was designed for TTCM in [5], where the state probabilities of the trellis directly depend on the channel errors, rather than on the coded sequence. Hence, for high SNRs or for highly correlated sources, the syndrome decoder would be more likely to experience a syndrome of all zeros because of the predominantly near-error-free transmissions. The BSD divides the received sequence into erroneous and error-free blocks according to their syndrome, and only the erroneous blocks are decoded by the BSD decoder.

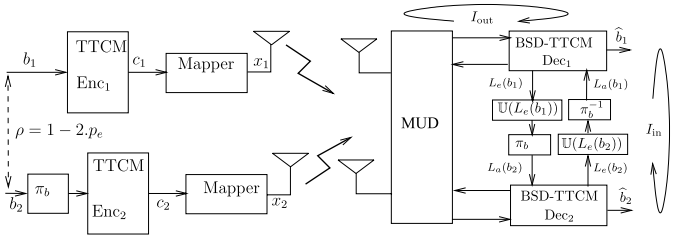

Fig. 1. Schematic diagram of the DJSTTCM scheme when communicating over uncorrelated Rayleigh fading MAC. The notation $L(\cdot)$ denotes the Log-Likelihood Ratio (LLR) while the subscripts a and e represent the *a priori* and *extrinsic* nature of the LLRs, respectively.

Blind online estimation of the cross-over probability p_e[1] at the Base Station (BS) still remains a persistent challenge in the practical design of DJSC schemes, since the correlation coefficient $\rho = 1-2p_e$ tends to vary over both space and time [6]. Additionally, at the BS the information exchanged between the decoders of each user has to be updated with an accurate p_e, since its inaccuracy would mislead the joint decoder, hence potentially inflicting catastrophic error propagation during the iterative decoding process. Several solutions have been proposed for addressing this issue [7], [9], which rely on perfect side-information DSC, while in this work we consider more practical DJSC scenario, where both user signals are transmitted over uncorrelated Rayleigh fading Multiple Access Channel (MAC).

Against this background, we propose a Distributed Joint Source and TTCM (DJSTTCM) coded scheme relying on the reduced-complexity BSD decoder of [5], for transmission over Rayleigh fading MAC. Additionally, our iterative decoder is capable of accurately estimating the time-variant cross-over probability p_e, hence enhancing the attainable Bit Error Ratio (BER) performance of the scheme.

II. Distributed Joint Source-Channel Coding Using BSD-TTCM

The schematic of our proposed DJSTTCM system is depicted in Fig. 1, where the binary sources $\{b_{1,i}\}_{i=1}^{N}$ and $\{b_{2,i}\}_{i=1}^{N}$ differ by the noise process $\{e_i\}_{i=1}^{N}$, where N is the length of each source block. The noise process is constituted by a BSC realization of a Bernoulli distributed random variable $\mathbb{E}$ associated with the parameter $p_e \in [0, 0.5]$ ($\mathbb{E} \sim \mathfrak{B}(p_e)$), s.t. $b_{2,i} = b_{1,i} \oplus e_i$ with a probability of $\Pr(b_1) \neq \Pr(b_2) = p_e$. Each user will encode its information sequence using TTCM at a coding rate of $R_{cm} = \frac{m}{m+1}$ and an $2^{m+1} = M$-level PSK/QAM modulation scheme. Both TTCM-coded sequences, namely

Manuscript received April 30, 2015; revised June 18, 2016; accepted June 21, 2016. Date of publication June 23, 2016; date of current version October 7, 2016. This work was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/Noo4558/1 and Grant EP/L018659/1 and in part by the European Research Council's Advanced Fellow Grant under the Beam-Me-Up project and of the Royal Society's Wolfson Research Merit Award is gratefully acknowledged. The associate editor coordinating the review of this letter and approving it for publication was T. Ngatched.

The authors are with the University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: ajra1c09@ecs.soton.ac.uk; zb2g10@ecs.soton.ac.uk; sxn@ecs.soton.ac.uk; lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/LCOMM.2016.2584598

[1] p_e is the cross-over probability when using the Binary Symmetric Channel (BSC) model, while the correlation coefficient ρ signifies the correlation between the sources and can be estimated as: $\rho = 1 - 2p_e$.

$\{c_{1,i}\}_{i=1}^{K}$ and $\{c_{2,i}\}_{i=1}^{K}$ associated with $K = N \cdot (1/R_{cm})$, will be mapped onto the corresponding modulated symbols $\{x_1\}$ and $\{x_2\}$, respectively, before transmission over an uncorrelated Rayleigh fading MAC.[2] Note that $\{x_1\}$ and $\{x_2\}$ are complex-valued phasors that represent the $(m+1)$-bit transmitted codewords $\{c_1\}$ and $\{c_2\}$, which can be obtained using the $\mu(\cdot)$-QAM/PSK mapping function.

Our transmission scenario might be interpreted as an Space-Division Multiple Access (SDMA) [10] system that support L users, each of whom is equipped with a single antenna, while the BS has P receive antennas. Thus, the signal $\mathbf{y}$ received at the BS is a $(P \times 1)$-element vector, which can be written as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{H}$ is an $(P \times L)$-element channel matrix, while $\mathbf{x}$ is a $(L \times 1)$-element transmitted signal vector and $\mathbf{n}$ is an $(P \times 1)$-element noise vector with a zero mean and a variance of $N_0/2$ per dimension. In order to avoid the computational complexity associated with Maximum Likelihood (ML)-based Multi-User Detector (MUD), we opted for the low-complexity MMSE-assisted Successive Interference Cancellation (SIC) MUD of [10, Sec. 8.3].

A. Joint Source-Channel Decoder

As seen in Fig. 1, our decoder employs two different iterations, namely:

- Inner Iteration (I_{in}): is the iteration between the two TTCM-BSD decoders, in which the correlation is exploited by exchanging the *extrinsic* LLRs.
- Outer Iteration (I_{out}): is the iteration between the MUD and BSD-TTCM decoders, which aims for combating the deleterious effects of channel fading by exchanging the *extrinsic* probabilities.

Similar to the TTCM decoder [11], each of the BSD-TTCM decoders of Fig. 1 consists of two parallel concatenated syndrome-based TCM MAP decoders. Fig. 2 shows the schematic of one of the two constituent decoders of the BSD-TTCM decoder. First, the k^{th} received symbol associated with our BSD-TTCM decoder's output is hard-demapped onto the nearest constellation point of the corresponding transmitted symbol x_k, yielding the hard-demapped symbol $\hat{y}_k$. Recall that in the TTCM scheme, the odd and even symbols are punctured for the upper and lower TCM encoders, respectively [11]. Accordingly, the parity bits associated with the punctured hard-demapped symbols are set to zero. Next, a pre-correction sequence $\hat{\mathbf{e}_k}$ is needed for updating any predicted errors in the hard-demapped sequence, where this sequence is set to zero during the first iteration. Then, the syndrome $\mathbf{s}$ is computed for estimating the corrected symbol stream $\mathbf{r}$ with the aid of the syndrome former matrix $\mathbb{H}^T$ as [5]:

$$\mathbf{s} = \mathbf{r}\mathbb{H}^T, \tag{2}$$

where each bit of the corrected symbol stream r_k is related to both $\hat{y}_k$ and $\hat{\mathbf{e}_k}$ as [5]:

$$r_k = \hat{y}_k \oplus \hat{\mathbf{e}}_k, \tag{3}$$

[2]Neither of the TTCM-coded sequences - namely neither $\{c_1\}$ nor $\{c_2\}$ - is punctured, since we aim for investigating the performance of our joint decoder in terms of both complexity reduction as well as cross-over probability estimation. Further details on the rate-region analysis can be found in [3].

where r_k, $\hat{y}_k$ and $\hat{\mathbf{e}}_k$ are represented by $(m+1)$ bits.

The syndrome sequence is then analysed in order to split the received sequence into error-free and erroneous sub-blocks. Hard-decisions are applied to the error-free sub-blocks without feeding the erroneous sub-blocks into the MAP decoder. The *a posteriori* LLRs corresponding to the error-free sub-blocks will be employed for estimating the cross-over probability $\hat{p}_e$, as it will be illustrated in Sec. II-C.

B. Syndrome-Based Joint MAP Decoder

The syndrome-based MAP decoder of [5] is employed in our BSD-TTCM decoder. In contrast to the conventional MAP decoder, which relies upon the codeword trellis, the syndrome-based MAP counterpart operates on the basis of the error trellis constructed using the syndrome former $\mathbb{H}^{\text{T}}$. In the conventional code-based trellis, each trellis path represents a possible legitimate codeword. By contrast, each path in the error-based trellis represents a hypothetical error sequence. The channel information probability $\Pr(y_k \mid x_k)$ of receiving y_k given x_k was transmitted gleaned from the MUD will be modified to $\Pr(y_k \mid \mathbf{e}_k)$ of receiving y_k, given that the channel error $\mathbf{e}_k$ is encountered, which can be expressed as:

$$\Pr(y_k \mid \mathbf{e}_k) = \frac{1}{\pi N_0} \exp\left(-\frac{|y_k - h_k x_k|^2}{N_0}\right). \tag{4}$$

The syndrome-based TCM MAP decoder shown in Fig. 2 calculates the *a posteriori* probabilities corresponding to the error-free sub-blocks $P_{\text{ef}}(b_k)$ using the classic forward and backward recursive coefficients namely α_k and β_k [11]. More explicitly, the channel's transition metric is formulated as:

$$\gamma_k(\grave{s}, s) = \Pr(b_k) \cdot \Pr(y_k \mid \mathbf{e}_k), \tag{5}$$

where $\Pr(b_k)$ is the *a priori* probability of the information part of $\mathbf{e}_k$, which is initialised to be equi-probable for the first iteration, while $(\grave{s}, s)$ denotes the transition emerging from state $\grave{s}$ to state s. However, the forward and backward recursions coefficients α_k and β_k have to consider the side-information provided by the other decoder which is formulated additionally as:

$$\alpha_k(s) = \sum_{\text{all } \grave{s}} \alpha_{k-1}(\grave{s}) \cdot \gamma_k(\grave{s}, s) \cdot A(b_k), \tag{6}$$

$$\beta_{k-1}(\grave{s}) = \sum_{\text{all } s} \beta_k(s) \cdot \gamma_k(\grave{s}, s) \cdot A(b_k), \tag{7}$$

where as shown in Fig. 2, $A(b_k)$ is the *a priori* probability provided by the other BSD-TTCM decoder of Fig. 1 corresponding to the LLRs $L_a(b_k)$, which is the interleaved and updated version of $L_e(b_k)$.

C. Cross-Over Probability Estimation Using BSD-TTCM

During the I_{in}, the exchanged LLRs have to be updated using an accurate estimate of the cross-over probability $\hat{p}_e$ by invoking the following update function:

$$\mathbb{U}(L_e(b_{1,2})) = \ln\frac{(1-\hat{p}_e)\exp[L_e(b_{1,2})] + \hat{p}_e}{(1-\hat{p}_e) + \hat{p}_e\exp[L_e(b_{1,2})]}, \tag{8}$$

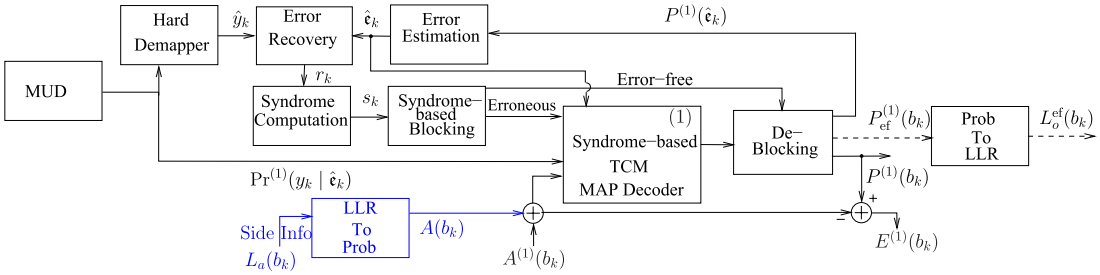

Fig. 2. Schematic of the proposed BSD-TTCM decoder. Here only one constituent decoder is shown, where $A^{(1)}$, $E^{(1)}$ and $P^{(1)}$ represent the *a priori*, *extrinsic* and *a posteriori* probabilities related to the corresponding syndrome-based TCM MAP decoder.

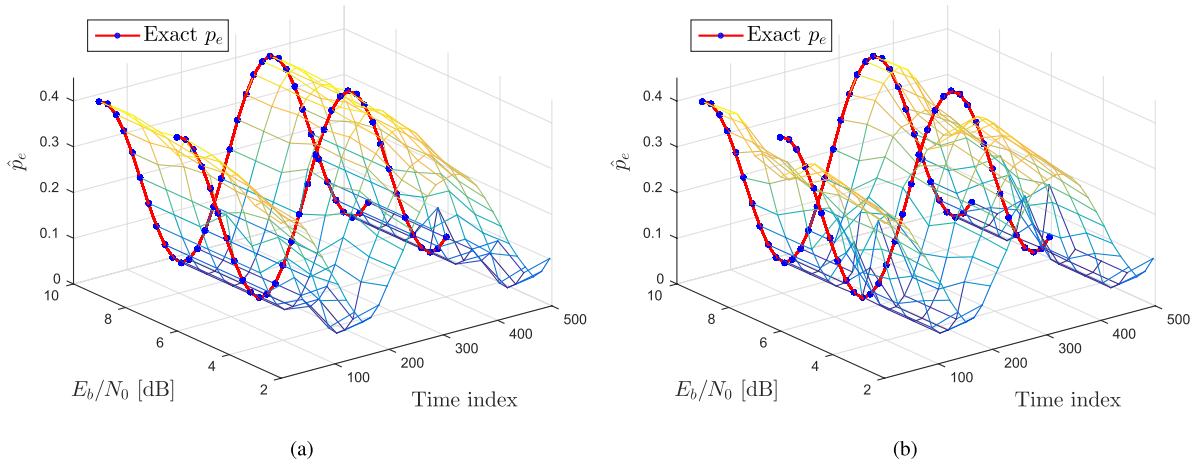

Fig. 3. Estimated $\hat{p}_e$ versus both E_b/N_0 and time comparison between **DJSTTCM-BSD** and **DJSTTCM-Conventional** for sinusoidal variation of p_e, where **MMSE-SIC** has been invoked as our MUD.

which is shown in Fig. 1. As stated previously, inaccurate BSC cross-over estimate $\hat{p}_e$ could lead to an error propagation during the joint iterative decoding. Our proposed joint decoder would provide an accurate estimate of $\hat{p}_e$ based on the reliable error-free LLRs gleaned from the two BSD-TTCM decoders' output from Fig. 2, namely, $L_o^{\rm ef}(b_{1,2})$ that are linked to the error-free sub-blocks as follows:

$$\hat{p}_e = \frac{1}{N_{\rm ef}} \sum_{i=1}^{N_{\rm ef}} \frac{\exp[L_o^{\rm ef}(b_1^i)] + \exp[L_o^{\rm ef}(b_2^i)]}{(1+\exp[L_o^{\rm ef}(b_1^i)])(1+\exp[L_o^{\rm ef}(b_2^i)])}, \tag{9}$$

where $N_{\rm ef}$ is the error-free LLR sub-block size. Similarly, the conventional codeword-trellis based TTCM benchmark employs the same formula of Eq. (9) when evaluating $\hat{p}_e$. However using the entire LLR block as the conventional TTCM decoder is unable to separate the received sequence into erroneous and error-free sub-blocks.

Fig. 3(a) demonstrates the accuracy of our probability estimator based on the DJSTTCM-BSD in comparison to the conventional DJSTTCM based method that its performance is shown in Fig. 3(b), where p_e is artificially varied sinusoidally between 0.025 and 0.42. Note that in order to make the figure legible, we portray the exact p_e based cosine curve at two values, namely at $E_b/N_0 = 6$ dB and $E_b/N_0 = 9$ dB, which are represented by filled circles. We opted for using 2/3-rate TTCM-8PSK transmission over a uncorrelated Rayleigh fading MAC using a block length of 12 000 8PSK symbols. Additionally, the decoder invokes "$I_{\rm in} = 2$" and "$I_{\rm out} = 2$", while the iteration between the two TCM components of the TTCM decoder $I = 8$ iterations were used inside the TTCM and BSD-TTCM decoders, respectively. As Fig. 3(a) demonstrates, our proposed estimator was capable of achieving an accurate $\hat{p}_e$ estimation of the sinusoidally varied p_e values. More explicitly, observes in Fig. 3(a) that our proposed estimator is capable of attaining an accurate BSC cross-over probability prediction at $E_b/N_0 = 6$ dB. Quantitatively, our BSD-based estimator exhibits a Mean Squared Error (MSE) of 3.5×10^{-5}, while its conventional counterpart imposes more than 100 times higher MSE of 5.5×10^{-3}. By contrast, the conventional DJSTTCM estimator characterised in Fig. 3(b) only achieved a similar estimation accuracy at the 3 dB higher value of $E_b/N_0 = 9$ dB. In order to ensure an accurate $\hat{p}_e$ estimation, we first have to determine the optimum size of the error-free sub-block LLR $L^o_{\rm ef}(b^{1,2})$. This is achieved by heuristically finding the minimum size of $L^o_{\rm ef}(b^{1,2})_{\rm min}$, which was found to be $L^o_{\rm ef}(b^{1,2})_{\rm min} \geq 2400$.

III. Simulation Results

The BER versus E_b/N_0 performance of our proposed scheme is shown in Fig. 4, where the two user signals are transmitted over a Rayleigh fading MAC. Again, both users employ 2/3-rate TTCM-8PSK where "$I_{\rm in} = 2$" and "$I_{\rm out} = 2$" are used by the decoder, while the correlation coefficient is $\rho = 0.4$ which corresponds to $p_e = 0.3$.

It may be readily observed from Fig. 4 that for the MMSE-SIC arrangement which is referred to as "DJSTTCM-Conventional MMSE-SIC" outperforms our proposed scheme

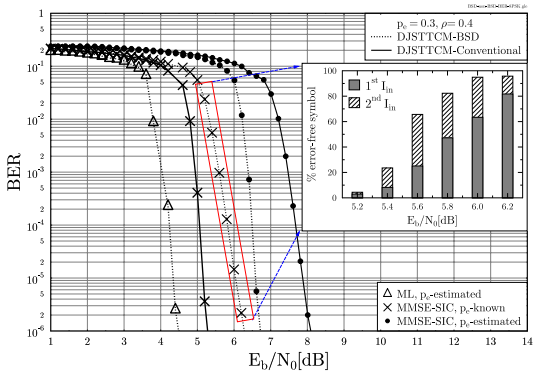

Fig. 4. BER result of both the **DJSTTCM-BSD** and **DJSTTCM-Conventional**, when considering ML and MMSE-SIC MUD, for the cases when p_e is known or estimated at the decoder.

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON FOR $I_{\text{in}} = 1$

E_b/N_0	**BSD**	**Conventional**	Complexity Reduction
5.2	3017	3048	1.0%
5.6	2797	3048	8.2%
6.0	1608	3048	47.2%
6.4	556	3048	81.75%
6.8	229	3048	92.5%
7.2	65	3048	97.8%
7.6	25	3048	99.0%

for the idealised scenario when p_e is perfectly known at the BS side by about 1 dB at a BER of 10^{-6}.

This is not unexpected, because the BSD scheme of Fig. 2 has been proposed mainly as low-complexity design [5]. However, for the realistic scenario, when the p_e is unknown at the decoder, our proposed "DJSTTCM-BSD-MMSE-SIC" scheme has a valuable 1.5 dB gain over the conventional DJSTTCM scheme as seen in Fig. 4. Moreover, as expected, when invoking the complex ML-based MUD in the DJSTTCM-BSD scheme the "DJSTTCM-BSD-ML" scheme outperforms the MMSE-SIC-based scheme "DJSTTCM-BSD-MMSE-SIC", by an E_b/N_0 gain of 2.5 dB as seen in Fig. 4. In order to investigate the achievable decoding complexity reduction, we have analysed the complexity of the "DJSTTCM-BSD-MMSE-SIC" scheme for the E_b/N_0 values spanning from 5.2 dB to 6.2 dB, as shown in Fig. 4. Here the complexity reduction is quantified by determining the number of error-free symbols that would avoid entering the error-trellis based MAP-BSD decoder of Fig. 2 at each iteration as a percentage of the total frame length. As Fig. 4 suggested, upon increasing the E_b/N_0 value the complexity was considerably reduced. It can also be readily observed that doubling the number of iterations between the BSD-TTCM decoders from "$I_{\text{in}} = 1$" to "$I_{\text{in}} = 2$" here the turbo-cliff region of the E_b/N_0 scale would lead to a significant increase in the percentage of the error-free symbols. Our computational complexity comparison is summarised in Table I.[3]

IV. CONCLUSIONS

In this letter, we have conceived a reduced-complexity DJSTTCM-aided BSD technique for practical DJSC design. The proposed iterative decoder was shown to be capable of estimating the BSC's cross-over probability with the aid of the LLR blocks that were deemed to be error-free, as identified by the syndrome sequence. Additionally, the decoding complexity is reduced further by invoking a MMSE-SIC based MUD, which has a complexity that is linearly proportional to both the number of users and to the constellation size.

REFERENCES

[1] Z. Xiong, A. D. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004.
[2] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, Jul. 1973.
[3] K. Anwar and T. Matsumoto, "Spatially concatenated codes with turbo equalization for correlated sources," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5572–5577, Oct. 2012.
[4] A. J. Aljohani, S. X. Ng, and L. Hanzo, "TTCM-aided rate-adaptive distributed source coding for Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 63, no. 3, pp. 1126–1134, Mar. 2014.
[5] Z. Babar, S. X. Ng, and L. Hanzo, "Reduced-complexity syndrome-based TTCM decoding," *IEEE Commun. Lett.*, vol. 17, no. 6, pp. 1220–1223, Jun. 2013.
[6] Y. Fang, "Joint source-channel estimation using accumulated LDPC syndrome," *IEEE Commun. Lett.*, vol. 14, no. 11, pp. 1044–1046, Nov. 2010.
[7] Y. Fang, "Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome," *IEEE Commun. Lett.*, vol. 13, no. 9, pp. 679–681, Sep. 2009.
[8] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, "Maximum likelihood BSC parameter estimation for the Slepian–Wolf problem," *IEEE Commun. Lett.*, vol. 15, no. 2, pp. 232–234, Feb. 2011.
[9] L. Cui, S. Wang, and S. Cheng, "Adaptive Slepian–Wolf decoding based on expectation propagation," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 252–255, Feb. 2012.
[10] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. New York, NY, USA: Cambridge Univ. Press, 2005.
[11] P. Robertson and T. Worz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 206–218, Feb. 1998.
[12] P. H.-Y. Wu, "On the complexity of turbo decoding algorithms," in *Proc. IEEE Veh. Technol. Conf.*, vol. 2. Rhodes, Greece, May 2001, pp. 1439–1443.

[3]The computational complexity is calculated using $C = I \cdot (4n + 18) \cdot 2^m - 3$ [12], where m is the code memory, while $n = 1/R_{cm}$. In our simulations the corresponding parameters are given by $m = 4$ and $n = 3/2$. Again, the number of iterations between the two TCM components of the TTCM decoder is $I = 8$. Explicitly, the complexity C includes the multiplications, divisions, comparisons, maximum, minimum and look-up table evaluations required by our max-log-MAP algorithm-based TTCM decoder.