EXIT Chart Based Joint Source-Channel Coding for Binary Markov Sources

Xiaobo Zhou*, Khoirul Anwar*, and Tad Matsumoto*†
*School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan.
†Centre for Wireless Communications, University of Oulu
P.O. Box 4500, 90014 University of Oulu, Finland.
{xiaobo, anwar-k, matumoto}@jaist.ac.jp

***Abstract*—In this paper, we propose a new joint source-channel decoding technique for transmitting binary Markov sources over AWGN channels. Our approach is based on serially concatenated coding and code doping for inner code. By combining the Markov source and the outer code trellis diagrams, a super trellis is constructed to exploit the time-domain correlation of the source. A modified version of BCJR algorithm is derived based on this super trellis, that can achieve considerable gain in terms of mutual information. The standard BCJR algorithm is used for decoding of inner code where code doping is adopted for better matching of extrinsic information transfer (EXIT) characteristics. EXIT chart analysis is performed to investigate convergence property of the proposed technique and to optimize the code parameters. Simulation results for bit error rate (BER) evaluation and EXIT chart analysis indicate that the proposed technique can achieve significant gains over the system in which source redundancy are not exploited, and thereby the BER performance of the proposed system is very close to the Shannon limit.**

I. Introduction

In conventional point-to-point (P2P) system, source coding and channel coding are usually designed independently. The optimal design of conventional system has largely relied on the design criterion supported by the Shannon's separation theorem. However, there are several impractical limitations with the separation theorem, which prevents communication systems from achieving desired performance in practice. As a consequence, joint source-channel (JSC) optimal design has drawn considerable attentions over the last two decades [1]–[4].

In the majority of the approaches to JSC decoding, variable-length code (VLC) is employed as source encoder and the implicit residual redundancy after source encoding is additionally used for error correction in the decoding process. A common idea is to represent the VLC as a symbol-level or bit-level trellis and decode it with maximum a posteriori probability (MAP) algorithm by modifying the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [5], [6]. The extended work of this category is presented in [7]–[9], where CC is replaced by more powerful turbo code.

Instead of utilizing the residual redundancy after source encoding, there are several literatures which focus on exploiting the memory structure of the source directly. There are two advantageous points with this class of JSC design: (1) no source encoding is needed and hence it can be employed in universal applications, while it simplifies the encoder structure and reduces power consumption at the transmitter, (2) the error correction capability of the system, as a whole, can further be improved. An approach of combining Hidden Markov Model (HMM) with the turbo code design framework is presented in [10]–[12], while combining HMM in the framework of low density parity check code (LDPC) design in [13], [14]. Besides HMM, Markov Chain (MC) can also be used to describe the source and combined with turbo code, which is presented in [12].

In this paper, we propose a JSC decoding technique for transmitting binary Markov sources over AWGN channels. The main idea is originated from [10], [12] which combine HMM source models with parallel-concatenated convolutional code (PCCC) based turbo codes. Instead, in this paper, serial-concatenated convolutional codes (SCCC) are used. By combining trellis diagrams of the source and the outer code, a super trellis can be constructed, over which a modified version of BCJR algorithm is performed. Another advantageous point of the technique proposed in this paper is code doping for the inner code, where the interleaved outer-coded bits are doped and transmitted at a certain rate. By choosing appropriate doping rate, the EXIT curves of inner and outer codes can be closely matched, which achieves near-limit performance.

The rest of the paper is organized as follow. Section II provides a brief description of the system model assumed in this paper. Section III defines the super trellis combining the source and outer code structure, and describes how the BCJR algorithm has to be modified to perform maximum a posteriori probability (MAP) decoding over the super trellis. EXIT chart analysis for the proposed SCCC based JSC is demonstrated in section IV, where encoder polynomial as well as doping rate optimization is sought for based on the EXIT analysis. The simulation results are presented in section V to confirm the superiority of the proposed technique. Section VI concludes this paper with some concluding statements.

II. System Model

A block diagram of the JSC model this paper assumes is illustrated in Fig. 1. The source we considered here is a stationary state emitting binary Markov source $\{U_t\}, 1 \leq t \leq T$,

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

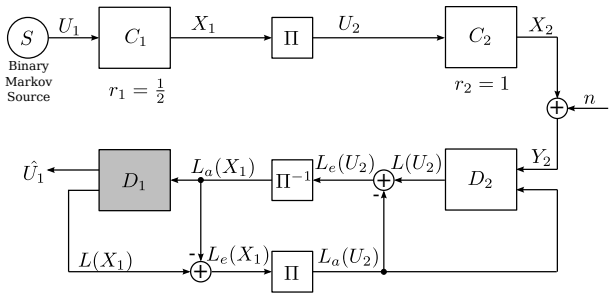


Fig. 1. Block diagram of the proposed system

with transition matrix:

$$A = [a_{i,j}] = \begin{bmatrix} a_{0,0} & a_{0,1} \\ a_{1,0} & a_{1,1} \end{bmatrix} = \begin{bmatrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{bmatrix}, \quad (1)$$

where $a_{i,j}$ is the transition probability defined by

$$a_{i,j} = \Pr\{U_t = j | U_{t-1} = i\}, \quad i, j = 0, 1. \quad (2)$$

The entropy rate of stationary Markov source [15] is given by

$$H(S) = -\sum_{i,j \in \{0,1\}} \mu_i a_{i,j} \log a_{i,j}, \quad (3)$$

where $\{\mu_i\}$ is the stationary state probability distribution.

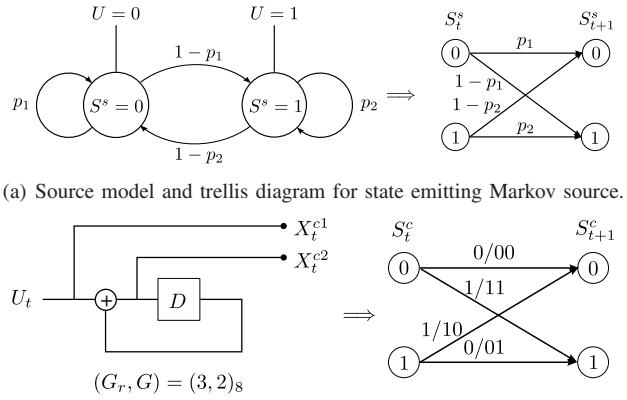
At the transmitter side, a rate $r_1 = 1/2$ outer code C_1 and a rate $r_2 = 1$ inner code C_2 are serially concatenated, yielding a overall rate r $(= r_1 r_2 = 1/2)$. The information bits from the binary Markov source are fed directly to C_1. The output of C_1, including systematic and parity bits, are then bit-interleaved by a random interleaver Π and encoded by C_2. The inner code C_2 is a rate-1 recursive convolutional code, and its output bits are replaced by its input (interleaved version of C_1 encoded bits, which are also the systematic bits for C_2 encoder). This process is referred to as code doping for inner code, which is detailed in Section IV. The C_2-encoded sequence including the doped bits are BPSK modulated and then transmitted over the channel affected by a zero mean one-dimensional Additive White Gaussian Noise (AWGN) with variance σ^2.

At the receiver side, the Markov source model and outer code trellis are integrated to form a super trellis. A modified BCJR algorithm can be derived to jointly perform decoding of the C_1 code and Markov source using the super trellis, as described in the next section. Iterative decoding is invoked between two soft input soft output (SISO) modules D_1 and D_2, according to the turbo decoding principle, where the standard BCJR and its modified version are performed for the decoding of inner code and joint outer code composed of C_1 and the Markov source, respectively.

III. Combining Outer Code and Source Model

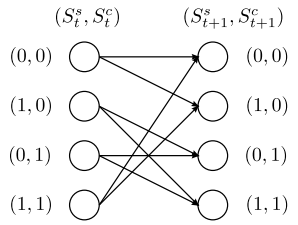
A. Representation of Super Trellis

Assume that the outer code is a convolutional code (CC), with which the memory length is v. There are 2^v states in the trellis diagram of this code, which are indexed by m, $m = 0, 1, \ldots, 2^v - 1$. The state of the outer encoder at



(a) Source model and trellis diagram for state emitting Markov source.

(b) A example for $(G_r, G) = (3, 2)_8$ RSC code and its trellis diagram.



(c) Super trellis with compound states derived from Markov source and $(G_r, G) = (3, 2)_8$ code.

Fig. 2. Trellis diagram for (a) binary Markov source, (b) RSC code with $(G_r, G) = (3, 2)_8$, (c) joint outer code.

time index t is denoted as S_t^c. Similarly, there are two states in order-1 binary Markov source, and the state at time index t is denoted as S_t^s, $S_t^s = 0, 1$. For a binary Markov model described in section II, the source model and its corresponding trellis diagram are illustrated in Fig. 2(a). The output value at a time instant t from the source is the same with the state value of S_t^s. The trellis branches represent the state transition probabilities, which is defined by (2). On the other hand, for the outer code, the branches in its trellis diagram indicate input/output characteristics.

At time instant t, the state of the source and the state of the outer code can be regarded as a new state (S_t^s, S_t^c), which leads to the super trellis diagram. A simple example of combining binary Markov source with a recursive convolutional code (RSC) with generator polynomial $(G_r, G) = (3, 2)_8$ is depicted in Fig. 2(c). At each state (S_t^s, S_t^c), the input to the outer encoder is determined, given the state of the Markov source. Actually, the new trellis branches represent both state transition probabilities of the Markov source and input/output characteristics of the outer code defined in its trellis diagram.

B. Modified BCJR Algorithm for Super Trellis

In this sub-section, we ignore momentarily the serially concatenated structure, and only focus on the decoding process performed over the super trellis diagram. The input sequence to the encoder, which is also a series of the states of Markov source $\{S_t^s\}$, is assumed to have length L. The output of the encoder is denoted as $\{X_t\} = \{X_t^{p1}, X_t^{p2}\}$. The coded binary sequence is BPSK mapped, i.e., logical "0" $\rightarrow$ +1

and logical "1" $\rightarrow$ -1, and then transmitted over AWGN channels. The received signal is a noise-corrupted version of the BPSK mapped sequence, denoted as $\{Y_t\} = \{Y_t^{p1}, Y_t^{p2}\}$. The received sequence from the time indexes t_1 to t_2 is denoted as $Y_{t_1}^{t_2} = [Y_{t_1}, Y_{t_1+1}, \ldots, Y_{t_2}]$. In the following, upper-case letters are used to represent random variables and lower-case letters their realizations.

The aim of the modified BCJR algorithm is to calculate conditional *log-likelihood ratio (LLR)* of the coded bits $\{X_t^{p1}\}$, based on the whole received sequence Y_1^L, which is defined by

$$\begin{aligned} L(X_t^{p1}) &= \log \frac{p(X_t^{p1}=1|Y_1^L=y_1^L)}{p(X_t^{p1}=0|Y_1^L=y_1^L)} \\ &= \log \frac{\sum\limits_{(i,m)\in B_t^1} p(S_t^s=i, S_t^c=m, Y_1^L=y_1^L)}{\sum\limits_{(i,m)\in B_t^0} p(S_t^s=i, S_t^c=m, Y_1^L=y_1^L)}, \end{aligned} \quad (4)$$

where B_t^k denotes the sets of states $\{(S_t^s = i, S_t^c = m)\}$ yielding the systematic output X_t^{p1} of the encoder being $k,\ k = 0, 1$.

In order to compute the last term in (4), three parameters indicating the probabilities defined as below have to be introduced:

$$\alpha_t(i,m) = p(S_t^s = i, S_t^c = m, Y_1^t = y_1^t), \quad (5)$$

$$\beta_t(i,m) = p(Y_{t+1}^L = y_{t+1}^L | S_t^s = i, S_t^c = m), \quad (6)$$

$$\begin{aligned} &\gamma_t(y_t, i', m', i, m) = \\ &\quad p(S_t^s = i, S_t^c = m, Y_t = y_t | S_{t-1}^s = i', S_{t-1}^c = m'). \end{aligned} \quad (7)$$

Now we have

$$p(S_t^s = i, S_t^c = m, Y_1^L) = \alpha_t(i,m)\beta_t(i,m). \quad (8)$$

Substituting (8) in (4), we obtain the whole set of equations for the modified BCJR algorithm. As mentioned in previous subsection, the new trellis branch contains information of input/output relationship corresponding to the state transition $S_t^c = m' \rightarrow S_t^c = m$, specified by the outer code trellis, as well as of the state transition probabilities depending on Markov source. Therefore, γ can be decomposed as

$$\begin{aligned} &\gamma_t(y_t, i', m', i, m) \\ &\quad = \begin{cases} a_{i',i}\gamma_t^*(y_t, m', m), & \text{if } (i', m') \in E_t(i,m); \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \quad (9)$$

where $a_{i',i}$ is defined in (2), and $\gamma_t^*(y_t, m', m)$ is defined as

$$\gamma_t^*(y_t, m', m) = p(Y_t = y_t, S_t^c = m | S_{t-1}^c = m'). \quad (10)$$

$E_t(i,m)$ is the set of states $\{(S_{t-1}^s, S_{t-1}^c)\}$ that have a trellis branch connected with state $(S_t^s = i, S_t^c = m)$.

Now that once γ is obtained, α and β can also be computed via the following recursive formulae

$$\alpha_t(i,m) = \sum_{i',m'} \alpha_{t-1}(i',m')\gamma_t(y_t, i', m', i, m), \quad (11)$$

$$\beta_t(i,m) = \sum_{i',m'} \gamma_t(y_t, i', m', i, m)\beta_{t+1}(i',m'), \quad (12)$$

Since the output encoder always starts from the state zero, while the probabilities for the Markov source starts from state "0" or state "1" is equal. Hence, the appropriate boundary condition for α is $\alpha_0(i,0) = 1/2,\ i = 0, 1$ and $\alpha_0(i,m) = 0,\ m \neq 0$. Similarly, the boundary conditions for β is $\beta_L(i,m) = 1/2^{v+1},\ i = 0, 1; m = 0, 1, \cdots, 2^v - 1$.

Combining all the results described above, we can obtain the conditional *LLR*s for X_t^{p1}, as

$$L(X_t^{p1}) = L_{ap}(X_t^{p1}) + L_{ch}(X_t^{p1}) + L_{ex}(X_t^{p1}), \quad (13)$$

where

$$L_{ap}(X_t^{p1}) = \log \frac{p(X_t^{p1}=1)}{p(X_t^{P1}=0)}, \quad (14)$$

$$L_{ch}(X_t^{p1}) = \log \frac{p(Y_t^{p1}=y_t^{p1}|X_t^{p1}=1)}{p(Y_t^{p1}=y_t^{p1}|X_t^{p1}=0)}, \quad (15)$$

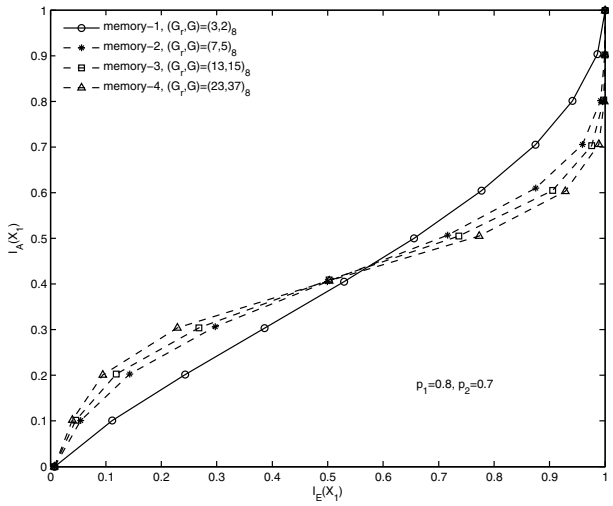
$$\begin{aligned} &L_{ex}(X_t^{p1}) = \\ &\log \frac{\sum\limits_{(i,m)\in B_t^1}\sum\limits_{i',m'} \alpha_{t-1}(i',m')\gamma_t(y_t^{p2}, i', m', i, m)\beta_t(i,m)}{\sum\limits_{(i,m)\in B_t^0}\sum\limits_{i',m'} \alpha_{t-1}(i',m')\gamma_t(y_t^{p2}, i', m', i, m)\beta_t(i,m)}, \end{aligned} \quad (16)$$

which constitutes the a priori *LLR*, the channel *LLR* and the extrinsic *LLR*, respectively, obtained as the result of the modified BCJR algorithm. The same representation should be applied to $\{X_t^{p2}\}$. Using this modified BCJR algorithm, an optimal decoder can be obtained but at twice the decoding complexity.

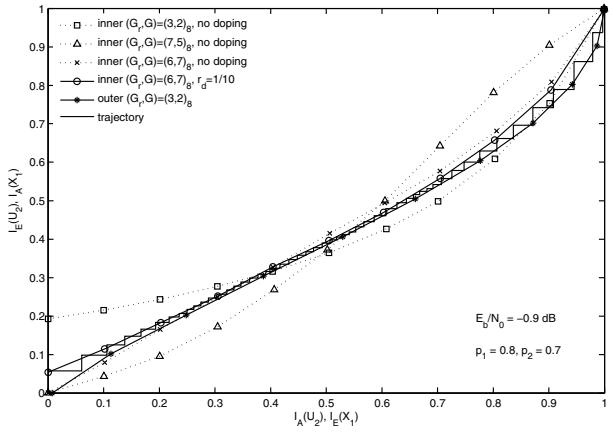
IV. Code Optimization with EXIT Chart Analysis

In this section, we present results of extrinsic information transfer (EXIT) chart analysis [16] conducted to determine the best matched pair of outer and inner code, and the doping rate for inner code. A example of the code optimization for the state-emitting Markov source with $p_1 = 0.8, p_2 = 0.7$ is demonstrated in Fig. 3. First, the outer code parameters can be determined using the EXIT chart analysis. The EXIT curves of some outer codes for Markov source with $p_1 = 0.8, p_2 = 0.7$ are shown in Fig. 3(a). It can be observed that memory-1 outer code with generator polynomials $(G_r, G) = (3, 2)_8$ exhibits an excellent extrinsic information transfer characteristic, and hence, it is the optimal outer code for Markov source with $p_1 = 0.8, p_2 = 0.7$.

With this fixed outer code, next we can determine the best inner code that roughly matches with outer code (i.e., the gap between the EXIT curves is minimized among those inner



(a) Determining optimal outer code for this Markov source.



(b) Determining optimal inner code and doping rate.

Fig. 3. Code optimization for state-emitting Markov source with $p_1 = 0.8, p_2 = 0.7$.

codes parameters), assuming no *code doping*[1] used. As shown in Fig. 3(b), at $E_b/N_0 = -0.9$ dB, a memory-2 inner code with $(G_r, G) = (6, 7)_8$ has a better matching with outer $(3, 2)_8$ code, compared with the inner $(3, 2)_8$ and $(7, 5)_8$ codes.

After the optimal inner code is obtained, which is best matched with the Markov source memory, the optimal code doping can be identified for the inner code by changing the doping rate to gradually change its EXIT curve, until there is a tunnel open between EXIT curves. Here, doping rate is the systematic-to-coded bits ratio, which can be denoted by $r_d = n_s/n_c$. In this example, with $r_d = 1/10$, there is a narrow tunnel open which the trajectory can sneak through until convergence point, as shown in Fig. 3(b). Now those three optimal code parameters can be obtained for Markov source with $p_1 = 0.8, p_2 = 0.7$. The optimization process is performed for the Markov source with different p_1 and p_2 values, and the optimal code parameters are summarized in Table I.

[1]The terminology "doping" was first proposed by ten Brink in [16] to express the notion that a small portion of the coded bits are substituted by their systematic counterpart, without changing the code rate. Here we use "code doping" rather than "doping" for the sake of clarity.

TABLE I
OPTIMIZED CODE PARAMETERS FOR MARKOV SOURCES WITH DIFFERENT p_1 AND p_2 VALUES

Source Parameters			Code Parameters		
p_1	p_2	$H(S)$	Outer	Inner	r_d
0.9	0.9	0.47	$(3,2)_8$	$(6,7)_8$	1/25
0.8	0.8	0.72	$(3,2)_8$	$(6,7)_8$	1/15
0.7	0.7	0.88	$(3,2)_8$	$(6,7)_8$	1/4
0.9	0.8	0.55	$(3,2)_8$	$(6,7)_8$	1/25
0.9	0.7	0.57	$(3,2)_8$	$(6,7)_8$	1/25
0.8	0.7	0.79	$(3,2)_8$	$(6,7)_8$	1/10

V. BER PERFORMANCE EVALUATION

In this section, we evaluate the bit error rate (BER) performance of the proposed joint source-channel decoding technique based on the results of code optimization provided in the previous sections. Both symmetric and asymmetric Markov sources are considered. The length of binary sequence generated by the Markov sources is 10000 bits, and in total 1000 different blocks were transmitted to guarantee the BER evaluation accuracy.

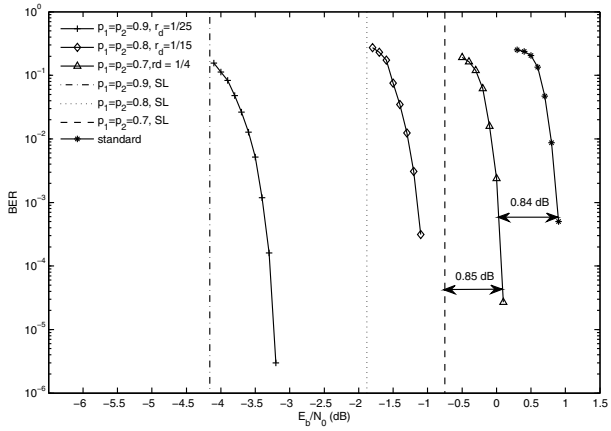
Assume that $H(S)$ denotes the source entropy and R the code rate, the threshold E_b/N_0 is determined from the condition $H(S)R \leq C$, where $C = \frac{1}{2}\log_2(1 + \frac{2E_bR}{N_0})$ is the capacity of the AWGN channel in bits per channel use. These conditions yield

$$(\frac{E_b}{N_0})_{lim} = \frac{2^{2\cdot H(S)\cdot R} - 1}{2R}. \quad (17)$$

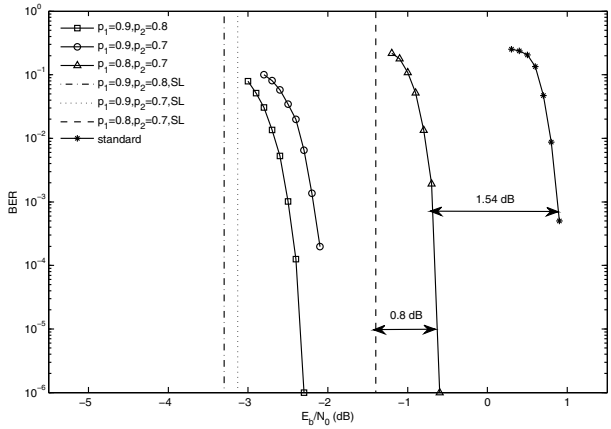
Using (3) and (17), the Shannon limit for Markov source with different p_1 and p_2 values can be obtained.

The performance of our proposed JSC decoding technique for transmitting binary Markov sources is shown in Fig. 4. Simulations are performed for different Markov sources with overall code rate $r = 1/2$. Theoretical limits for those cases are also plotted in the same figure. To demonstrate the performance gains obtained by the exploitation of the memory structure, the BER curve of serially concatenated codes with the same code parameters, decoded with the standard BCJR algorithm, is provided in the same figure, which is referred to as "standard". It can be clearly observed that at the BER level of 10^{-5}, when $p_1 = p_2 = 0.7$, our system offers a gain of 0.84 dB over the "standard" system, while achieving a gap of 0.85 dB to the Shannon limit. All the gains and gaps are summarized in Table II, together with the results of the technique proposed in [12], which is referred to as Joint Source Channel Turbo Coding (JSCTC) as a reference.

It can be found from the table that substantial gains can be achieved for different Markov sources with both our proposed SCCC based and [12]'s proposed PCCC based systems. This implies that exploiting the source redundancy in JSC technique provides us with significant advantage. Besides the gap in the case of $p_1 = p_2 = 0.7$, our system outperforms JSCTC proposed in [12] in terms of both gains over the standard BCJR



(a) Symmetric cases



(b) Asymmetric cases

Fig. 4. BER performance of proposed system for binary Markov sources, AWGN channel.

TABLE II
BER PERFORMANCE COMPARISON BETWEEN PROPOSED SYSTEM AND JSCTC

Source Parameters		JSCTC		Our system	
p_1	p_2	Gain(dB)	Gap(dB)	Gain(dB)	Gap(dB)
0.9	0.9	3.03	1.36	4.24	0.86
0.8	0.8	1.29	0.94	2.04	0.78
0.7	0.7	0.7	0.73	0.84	0.85
0.9	0.8	2.31	1.14	3.24	1.0
0.9	0.7	2.02	1.31	3.04	1.02
0.8	0.7	0.92	0.83	1.54	0.8

decoding and the gaps to the Shannon limit for binary Markov sources.

It should be emphasized that JSCTC employs two memory-4 constituent codes and extrinsic *LLR*s generated by the decoders have to be modified before being fed into another decoder; while our proposed system uses a memory-1 outer code (as described in Section III-B, the decoding complexity is equivalent to a memory-2 RSC code) and a memory-2 inner coder, and the extrinsic *LLR*s obtained by the decoders can be exchanged directly between two decoders. Hence, the decoding complexity of our proposed system is significantly smaller than that of JSCTC.

VI. Conclusions

In this paper, we have presented a joint source-channel decoding scheme for binary Markov sources using serial concatenation of CCs where code doping was adopted for inner code. To fully exploit the Markov source memory, the trellis of Markov source and the trellis of outer code are combined to construct a super trellis. A modified BCJR algorithm, suitable for this structure, has been derived based on the super trellis diagram. To further approach the Shannon limits, code doping is employed for inner code, where the standard BCJR algorithm can be used to decode the inner code. The extrinsic information generated by the inner and joint outer decoders can be exchanged via interleaver/deinterleaver in the same way as the decoding of serially concatenated codes. Simulation results show that our system can achieve equivalent or even better performance than JSCTC technique, while requiring much smaller decoding complexity.

References

[1] L. Hanzo, R. G. Maunder, J. Wang, and L.-l. Yang, *Near-Capacity Variable-Length Coding: Regular and Exit-Chart-Aided Irregular Designs*. John Wiley & Sons, Inc., Nov. 23 2010.

[2] P. Duhamel and kieffer Michel, *Joint Source-Channel Decoding: A Cross-Layer Perspective with Applications in Video Broadcasting over Mobile and Wireless Networks*. Eurasip and Academic Press Series in Signal and Image Processing, Jan. 7 2010.

[3] L. Schmalen, M. Adrat, T. Clevorn, and P. Vary, "Exit chart based system design for iterative source-channel decoding with fixed-length codes," *IEEE Trans. Commun.*, no. 99, pp. 1–8, 2011, early Access.

[4] L. Schmalen and P. Vary, "Error resilient turbo compression of source codec parameters using inner irregular codes," in *Proc. Int Source and Channel Coding (SCC) ITG Conf*, 2010, pp. 1–6.

[5] R. Thobaben and J. Kliewer, "On iterative source-channel decoding for variable-length encoded markov sources using a bit-level trellis," in *Proc. 4th IEEE Workshop Signal Processing Advances in Wireless Communications SPAWC 2003*, 2003, pp. 50–54.

[6] ——, "Low-complexity iterative joint source-channel decoding for variable-length encoded markov sources," *IEEE Trans. Commun.*, vol. 53, no. 12, pp. 2054–2064, 2005.

[7] K. Lakovic and J. Villasenor, "Combining variable length codes and turbo codes," in *Proc. IEEE 55th Vehicular Technology Conf. VTC Spring 2002*, vol. 4, 2002, pp. 1719–1723.

[8] M. Jeanne, J. C. Carlach, P. Siohan, and L. Guivarch, "Source and joint source-channel decoding of variable length codes," in *Proc. IEEE Int. Conf. Communications ICC 2002*, vol. 2, 2002, pp. 768–772.

[9] M. Jeanne, J.-C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 10–15, 2005.

[10] J. Garcia-Frias and J. D. Villasenor, "Combining hidden markov source models and parallel concatenated codes," *IEEE Commun. Lett.*, vol. 1, no. 4, pp. 111–113, 1997.

[11] ——, "Joint turbo decoding and estimation of hidden markov sources," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1671–1679, 2001.

[12] G.-C. Zhu and F. Alajaji, "Joint source-channel turbo coding for binary markov sources," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, pp. 1065–1075, 2006.

[13] L. Yin, J. Lu, and Y. Wu, "LDPC-based joint source-channel coding scheme for multimedia communications," in *Proc. 8th Int. Conf. Communication Systems ICCS 2002*, vol. 1, 2002, pp. 337–341.

[14] M. Fresia, F. Perez-Cruz, and H. V. Poor, "Optimized concatenated LDPC codes for joint source-channel coding," in *Proc. IEEE Int. Symp. Information Theory ISIT 2009*, 2009, pp. 2131–2135.

[15] T. M. Cover and J. A. Thomas, *Elements of Information theory 2nd Edition*. USA: John Wiley & Sons, Inc., 2006.

[16] S. ten Brink, "Code characteristic matching for iterative decoding of serially concatenated codes," *Annals of Telecommunications*, vol. 56, pp. 394–408, 2001.