Serially Concatenated Joint Source-Channel Coding for Binary Markov Sources

Xiaobo Zhou*, Khoirul Anwar*, Member, IEEE, and Tad Matsumoto*†, Fellow, IEEE
*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*—concatenated source channel coding for binary Markov sources over AWGN channels. To exploit the memory structure inherent within the sequence output from the source, modifications are made on the BCJR algorithm. To decode the outer code, the modified version of the BCJR algorithm is used, while the inner code by the standard version of the algorithm. Since optimal design of serially concatenated convolutional code falls into the problem of curve matching between the extrinsic information transfer (EXIT) curves of the inner and outer codes, we first evaluate the EXIT curve of the outer code decoded by the modified BCJR algorithm. It is then shown that the EXIT curve obtained by the modified BCJR algorithm is better matched with short memory inner convolutional code, which significantly reduces coding/decoding complexity. Numerical results demonstrate significant gains over the systems in which source statistics are not exploited (i.e., the standard BCJR algorithm is used for the both codes), and thereby narrowing the performance gap to the Shannon limit. We also compare in this paper the performance of the proposed design with the algorithm presented in [1], designed also for transmission of binary Markov source using parallel concatenated convolutional code (the authors of Ref. [1] refer the technique as Joint Source Channel Turbo Code (JSCTC)). It is shown that our proposed system is superior in both system complexity and BER performance to the JSCTC technique presented in [1].**

I. Introduction

The fundamental problem of future ubiquitous communication systems is how to transmit information data efficiently and reliably from source to destination over the channels suffering from various impairments such as fading, interference, and/or distortions due to practical limitations. In traditional communication systems, source and channel coding are performed separately: the source coding removes the redundancy of the data sequence for the source, while channel coding adds redundancy to correct errors cause by the noise in the communication channel. The global optimality of separately optimizing the source and channel coding scheme is guaranteed by the Shannon separation theorem, with the assumption that the source entropy is lower than the channel capacity. However, there are three drawbacks of seeking separately for the optimality based on the separation theorem: (1) if the source have memory structure, such as speech and image/video, there may be still redundancy left after source encoder; (2) it needs infinite code length, which usually incurs unacceptable large latency of communications; (3) the system tends to break down completely when the channel quality falls under a certain threshold, which means the channel code is no longer capable of correcting the errors [2].

As a remedy to those fundamental drawbacks, joint source-channel coding (JSC) has gained considerable attention over the last decade. In many approaches to JSC coding, the memory structure of the source or the implicit redundancy after source encoding is additionally used to enhance the error correcting capability of the joint decoder. In 1995, Hagenauer [3] proposed a JSC coding scheme with Viterbi decoding for PCM transmission used in the full rate GSM speech codec, where modifications are made to the Viterbi decoding algorithm to take the redundancy after source encoder into account.

Discovery of the turbo codes, followed by the re-discovery of the LDPC codes, has invoked a new concept of the JSC code design. In [1], [4], [5], Turbo code is used as JSC code while LDPC code in [6]–[9], where source statistic is exploited in the decoding process, and hidden markov model is used to describe the source. Instead of making modifications on the decoding algorithms, [10]–[13] modifies the encoder based on the source statistics and use standard maximum a posteriori probability (MAP) algorithm (it is well known that for convolutional code, MAP decoding can be efficiently performed by the BCJR algorithm) at the receiver. Quite recently, Multiple Label Mapping (MLM) is investigated in [14] to eliminate the boundary problem due to the variable length source coding, and the use of Burrows-Wheeler Transform (BWT) is investigated in [15], both with the aim of achieving efficient JSC code design.

In our proposed system, the source memory is utilized during the decoding process, in the same way as in [1], [4], [5]. Especially, [1] presents a JSC coding scheme using turbo code for binary markov source which achieves considerable gains over standard turbo code. However, the technique presented in [1] requires relatively large memory length constituent codes for the turbo code used. Furthermore, the extrinsic information obtained by the second decoder needs to be modified before feeding it back to the first constituent decoder, which requires higher decoding complexity than that with the standard BCJR algorithm.

In this paper, we propose a JSC coding technique using

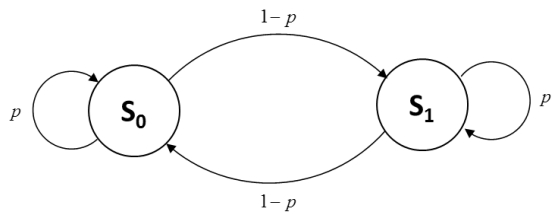


Fig. 1. symmetric state emitting Markov source. For state S_i, i is transmitted, $i \in \{0, 1\}$.

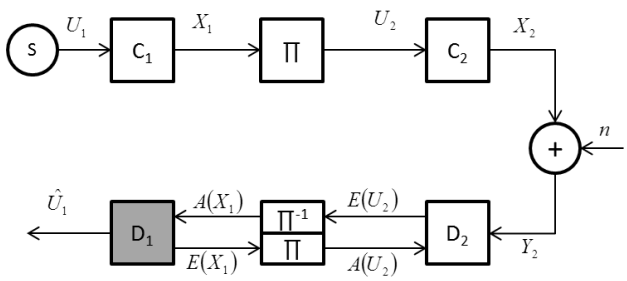


Fig. 2. Block diagram of the system.

serially concatenate convolutional code, where, since the code optimality is the issue of the EXIT curve matching, not the strength of the code itself, it is shown through EXIT analysis that very short memory codes can achieve almost equivalent or even better performance than the technique shown in [1]. More specially, memory-1 outer code and memory-2 inner code are used in this paper. The standard BCJR algorithm is modified to take into account the memory structure of the Markov source, and used for decoding the outer code, while the standard BCJR algorithm is used for decoding inner code, which enables directly exchange of the extrinsic information between the inner and outer decoders without requiring any modifications. As a whole, the decoding complexity of the proposed JSC technique is much lower than that required by [1].

The reminder of this paper is organized as follow. In section II, we present a brief description of the system model. Section III makes modifications on the BCJR algorithm to exploit the memory structure of the Markov source. Section IV shows results of simulations conducted to demonstrate the superiority of the proposed technique. Finally, Section V concludes this paper with some concluding remarks.

II. System Model

The source we considered in this paper is a stationary state emitting binary Markov source $\{U_t\}, 1 \leq t \leq T$, with the property that the current binary value is determined only by its previous value, as

$$\Pr\{U_t | U_{t'}, 1 \leq t' < i\} = \Pr\{U_t | U_{t-1}\}, \tag{1}$$

of which can be conveniently described by using the transition matrix:

$$A = [a_{i,j}] = \begin{bmatrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{bmatrix}, \tag{2}$$

with the transition probability defined by

$$a_{i,j} = \Pr\{U_t = j | U_{t-1} = i\}, \quad i, j \in \{0, 1\}. \tag{3}$$

If $p_1 = p_2 = p$, the source is referred to as symmetric Markov source, as shown in Fig. 1, where S_0 is the state that emits "0" and S_1 the state that emits "1". The entropy rate of stationary Markov source [16] is

$$H(S) = -\sum_{i,j \in \{0,1\}} \mu_i a_{i,j} \log a_{i,j}, \tag{4}$$

where $\{\mu_i\}$ is the stationary distribution. For symmetric Markov source, $\mu_0 = \mu_1 = 0.5$ holds, which yields

$$H(S) = -p \log p - (1-p) \log(1-p). \tag{5}$$

If $p_1 \neq p_2$, the source is referred to as asymmetric Markov source. The entropy rate can also be easily obtained by (4). For the sake of simplicity, we only consider symmetric Markov source in this paper, and the result can easily be extended to the asymmetric case [1].

The block diagram of our proposed JSC coding scheme is illustrated in Fig. 2. 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 $(= r_1 r_2 = r_1)$. The information bits from the binary Markov source are fed directly to C_1. The output of C_1, including systematic bits and parity bits, are then bit-interleaved by π and encoded by C_2. Since C_2 is a rate 1 code, only the encoded bits of C_2, not including its systematic input, are BPSK modulated and then transmitted over the channel suffering form zero mean Additive White Gaussian Noise (AWGN) with variance σ^2.

At the receiver side, iterative decoding is invoked between two soft input soft output (SISO) decoders D_1 and D_2, according to the turbo decoding process, where the standard BCJR decoding algorithm and its modified version, as described in the next section, are performed for decoding of C_2 and C_1, respectively. Extra gains in terms of extrinsic mutual information can be achieved by utilizing the modified BCJR algorithm in decoder D_1 that exploits knowledge about the Markovianity of the source.

III. Decoder Design for Binary Markov Source

The goal of this section is to derive the modified version of the BCJR algorithm that takes into account the source memory. Hence, in this section, we ignore momentarily the serially concatenated structure, even though it is the core topic of this paper, and only focus on the decoding process of one decoder using the BCJR algorithm. For the sake of notation simplicity, the same notations to describe the transmitted and the received symbols, X_k and Y_k, respectively, are used as in Fig. 2.

A. Standard BCJR algorithm

First of all, we briefly describe the standard BCJR decoding algorithm, originally developed to perform the MAP algorithm [17] for convolutional code (CC). For a CC with memory length v, there are 2^v states in its trellis diagram, which is indexed by $m,\ m = 0, 1, \cdots, 2^v - 1$. The input to the encoder is denoted as $\{U_t\}$, of which length is L. Let's assume that $r_1 = 1/2$ for the simplicity. Then, 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 is transmitted over a AWGN channel. 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 during time duration from t_1 to t_2 is denoted as $Y_{t_1}^{t_2} = Y_{t_1}, Y_{t_1+1}, \ldots, Y_{t_2}$. When summarizing and making modifications on the BCJR algorithm, we do not assume that the code is either systematic or non-systematic. Because the BCJR decoder can calculate the extrinsic $log\text{-}likelihood\ ratio$ (LLR) of coded and uncoded (information) bits by giving labeling properly in the trellis diagram of the code corresponding to its input-output relationship [18].

The BCJR algorithm evaluates the conditional LLR for $\{X_t^{p1}\}$ based on the whole received sequence Y_1^L, which is defined by

$$\begin{aligned} L(X_t^{p1}) &= \log \frac{\Pr\{X_t^{p1} = 1 | Y_1^L\}}{\Pr\{X_t^{p1} = 0 | Y_1^L\}} \\ &= \log \frac{\Pr\{X_t^{p1} = 1, Y_1^L\}}{\Pr\{X_t^{p1} = 0, Y_1^L\}}. \end{aligned} \tag{6}$$

To compute the LLR of X_t^{p1}, we use the joint probability

$$\sigma_t(m', m) = \Pr\{S_{t-1} = m', S_t = m, Y_1^L\}, \tag{7}$$

and rewrite (6) as

$$\begin{aligned} L(X_t^{p1}) &= \log \frac{\Pr\{X_t^{p1} = 1, Y_1^L\}}{\Pr\{X_t^{p1} = 0, Y_1^L\}} \\ &= \log \frac{\sum\limits_{(m',m) \in B_t^1} \sigma_t(m', m)}{\sum\limits_{(m',m) \in B_t^0} \sigma_t(m', m)}, \end{aligned} \tag{8}$$

where B_t^j denotes the set of transitions $S_{t-1} = m' \rightarrow S_t = m$ such that the output on that transition is $X_t^{p1} = j, j \in (0, 1)$. In order to compute (7), three parameters indicating the probabilities defined as below have to be introduced:

$$\begin{aligned} \alpha_t(m) &= \Pr\{S_t = m, Y_1^t\}, \\ \beta_t(m) &= \Pr\{Y_{t+1}^L | S_t = m\}, \\ \gamma_t(m', m) &= \Pr\{S_t = m, Y_t | S_{t-1} = m'\}, \end{aligned} \tag{9}$$

Now we have

$$\sigma_t(m', m) = \alpha_{t-1}(m') \gamma_t(m', m) \beta_t(m). \tag{10}$$

It is easy to show that $\alpha_t(m)$ and $\beta_t(m)$ can be computed via the following recursive formulae

$$\begin{aligned} \alpha_t(m) &= \sum_{m'} \alpha_{t-1}(m') \gamma_t(m', m), \\ \beta_t(m) &= \sum_{m'} \beta_{t+1}(m') \gamma_{t+1}(m, m'). \end{aligned} \tag{11}$$

Since the encoder always starts form zero state, the appropriate boundary condition for α is $\alpha_0(0) = 1$ and $\alpha_0(m) = 0,\ m \neq 0$. The boundary conditions for β depends on whether the trellis diagram is terminated by transmitting the tail bits or not. If we leave the encoder unterminated, the corresponding condition for β is $\beta_L(m) = 1/2^v,\ m = 0, 1, \cdots, 2^v - 1$; otherwise, $\beta_L(0) = 1$ and $\beta_L(m) = 0,\ for\ all\ m \neq 0$. In our system, we use long and random enough interleaver, so that $LLRs$ can be regarded as statistically independent.

From the above descriptions, it is found that γ plays a crucial role in computing the $LLRs$. Because Y_k^{p1} and Y_k^{p2} are statistically independent, γ can be computed by

$$\begin{aligned} \gamma_t(m', m) = \Pr\{S_t = m | S_{t-1} = m'\} \Pr\{Y_t^{p1} | X_t^{p1} = x_t^{p1}\} \\ \Pr\{Y_t^{p2} | X_t^{p2} = x_t^{p2}\}, \end{aligned} \tag{12}$$

The first term $\Pr\{S_t = m | S_{t-1} = m'\}$ is determined by the statistic information of the both input and output bits, as:

$$\begin{aligned} \Pr\{S_t = m | S_{t-1} = m'\} = \Pr\{U_t = u_t\} \Pr\{X_t^{p1} = x_t^{p1}\} \\ \Pr\{X_t^{p2} = x_t^{p2}\}, \end{aligned} \tag{13}$$

where the input/output bits, $u_t/(x_t^{p1}, x_t^{p2})$, are associated with trellis branch of $S_{t-1(m')} \rightarrow S_t(m)$. It should be mentioned that $\Pr\{U_t = 1\} = \Pr\{U_t = 0\} = 1/2$. Then we can rewrite (8) as

$$L(X_t^{p1}) = L_{ap}(X_t^{p1}) + L_{ch}(X_t^{p1}) + L_{ex}(X_t^{p1}), \tag{14}$$

where

$$L_{ap}(X_t^{p1}) = \log \frac{\Pr\{X_t^{p1} = 1\}}{\Pr\{X_t^{p1} = 0\}}, \tag{15}$$

$$L_{ch}(X_t^{p1}) = \log \frac{\Pr\{Y_t^{p1} | X_t^{p1} = 1\}}{\Pr\{Y_t^{p1} | X_t^{p1} = 0\}}, \tag{16}$$

and $L_{ex}(X_t^{p1})$ is defined as (17) at the top the next page, which are called the a priori LLR, the channel LLR, and the extrinsic LLR, respectively. If the decoder is not connected to the channel such as outer code of the serially concatenated code, it can not get any information about coded bits from the channel, which means $L_{ch} = 0$ and $L(X_t^{p1}) = L_{ap}(X_t^{p1}) + L_{ex}(X_t^{p1})$. The same result can be obtained for X_t^{p2}.

In iterative decoding, $L_{ex}(X_t^{p1})$ and $L_{ex}(X_t^{p2})$ are rearranged by the interleaver and fed into another decoder.

B. Modified BCJR Algorithm

In the original BCJR algorithm, the information bits are assumed to be memoryless. With the presence of the source correlation, the BCJR algorithm can well be modified to best

$$L_{ex}(X_t^{p1}) = \log \frac{\sum_{(m',m)\in B_t^1} \alpha_{t-1}(m') \Pr\{U_t = u_t\} \Pr\{X_t^{p2} = x_t^{p2}\} \Pr\{Y_t^{p2}|X_t^{p2} = x_t^{p2}\}\beta_t(m)}{\sum_{(m',m)\in B_t^0} \alpha_{t-1}(m') \Pr\{U_t = u_t\} \Pr\{X_t^{p2} = x_t^{p2}\} \Pr\{Y_t^{p2}|X_t^{p2} = x_t^{p2}\}\beta_t(m)} \tag{17}$$

$$L'_{ex}(X_t^{p1}) = \log \frac{\sum_{(m',m)\in B_t^1} \sum_{i',i} \alpha_{t-1}(m',i') a_{i',i} \Pr\{U_t = i\} \Pr\{X_t^{p2} = x_t^{p2}\} \Pr\{Y_t^{p2}|X_t^{p2} = x_t^{p2}\}\beta_t(m,i)}{\sum_{(m',m)\in B_t^0} \sum_{i',i} \alpha_{t-1}(m',i') a_{i',i} \Pr\{U_t = i\} \Pr\{X_t^{p2} = x_t^{p2}\} \Pr\{Y_t^{p2}|X_t^{p2} = x_t^{p2}\}\beta_t(m,i)} \tag{27}$$

ultilize the redundancy inherent within the Markov source. To achieve this goal, variables α, β and γ have to be modified as

$$\begin{aligned} \alpha_t(m,i) &= \Pr\{S_t = m, U_t = i, Y_1^t\}, \\ \beta_t(m,i) &= \Pr\{Y_{t+1}^L | S_t = m, U_t = i\}, \end{aligned} \tag{18}$$

$$\begin{aligned} &\gamma_t(m',i',m,i) \\ &\quad = \Pr\{S_t = m, U_t = i, Y_t | S_{t-1} = m', U_{t-1} = i'\}, \end{aligned} \tag{19}$$

with which α and β can also be calculated in the same way as in the standard BCJR algorithm, as

$$\begin{aligned} \alpha_t(m,i) &= \sum_{m',i'} \alpha_{t-1}(m',i')\gamma_t(m',i',m,i), \\ \beta_t(m,i) &= \sum_{m',i'} \beta_{t+1}(m',i')\gamma_{t+1}(m',i',m,i), \end{aligned} \tag{20}$$

Then the joint probability σ can then be derived as

$$\sigma_t(m',m) = \sum_{i',i} \alpha_{t-1}(m',i')\gamma_t(m',i',m,i)\beta_t(m,i). \tag{21}$$

Next we show how to compute $\gamma_t(m',i',m,i)$. It can be decomposed as

$$\begin{aligned} &\gamma_t(m',i',m,i) \\ &= \Pr\{S_t = m, U_t = i, Y_t | S_{t-1} = m', U_{t-1} = i'\} \\ &= \Pr\{S_t = m, U_t = i | S_{t-1} = m', U_{t-1} = i'\} \\ &\qquad \Pr\{Y_t^{p1} | X_t^{p1} = x_t^{p1}\} \Pr\{Y_t^{p2} | X_t^{p2} = x_t^{p2}\}, \end{aligned} \tag{22}$$

The first term is a "joint" probability reflecting the two factors: input/output relationship corresponding to the state transition $S_{t-1} \rightarrow S_t$, specified by the encoder structure, of which influence appears in the statistics of the a priori information, and the other the transition probability depending on memory structure of the Markov source. We approximate this term by

$$\begin{aligned} &\Pr\{S_t, U_t | S_{t-1}, U_{t-1}\} \approxeq \Pr\{U_t = i | U_{t-1} = i'\} \\ &\qquad \Pr\{U_t = i\} \Pr\{X_t^{p1} = x_t^{p1}\} \Pr\{X_t^{p2} = x_t^{p2}\}. \end{aligned} \tag{23}$$

Now, let us compare (22) and (23) with (12) and (13). It is now found that the transition probability $\Pr\{U_t = i | U_{t-1} = i'\}$ is invoked in the computation of γ. Thereby, we can now obtain the LLR for $\{X_t^{p1}\}$ as

$$L'(X_t^{p1}) = L'_{ap}(X_t^{p1}) + L'_{ch}(X_t^{p1}) + L'_{ex}(X_t^{p1}) \tag{24}$$

with

$$L'_{ap}(X_t^{p1}) = \log \frac{\Pr\{X_t^{p1} = 1\}}{\Pr\{X_t^{p1} = 0\}}, \tag{25}$$

$$L'_{ch}(X_t^{p1}) = \log \frac{\Pr\{Y_t^{p1} | X_t^{p1} = 1\}}{\Pr\{Y_t^{p1} | X_t^{p1} = 0\}}, \tag{26}$$

and $L'_{ex}(X_t^{p1})$ is defined as (27), located at the top of this page.

Comparing the expressions described above with the standard BCJR algorithm, it is found that with the modified BCJR algorithm, the a priori LLR and the channel LLR (actually this term is 0 for the outer code of serially concatenated codes) stay the same as in the standard BCJR algorithm, respectively. The statistical structure of the source is exploited inherently within forward-backward calculations, resulting in improved extrinsic LLR in the presence of source memory, which can be fed directly into the other constituent decoder via interleaver without involving any other computations. It should be noticed that if we apply this modified algorithm for memoryless source, since $a_{i',i} = 0.5,\ for\ i', i \in (0,1)$, the extrinsic LLR of the modified BCJR algorithm, given by (27), is the same as that of the standard BCJR. Moreover, as the source correlation becomes larger, the extrinsic LLR will also becomes larger, which will help the decoder recover the information bits even at lower Signal to Noise Ratio (SNR) value range. The expected impact of using the modified BCJR algorithm is verified in next section through computer simulations.

IV. Numerical Results

In this section, we present results of EXIT analysis [19], [20] conducted to identify the impact of source correlation on outer coder, as well as the convergence property of the proposed system. Then the BER performance of our scheme is evaluated and compared with the JSC coding technique shown in [1] in terms of the gap in E_b/N_0 to the theoretical limit derived from the Shannon capacity.

A. EXIT Analysis

1) Outer Coder: From the description in Section III-B, we know that the source memory helps increase the output extrinsic information of outer decoder. As described in Section II, the correlation of the source can be parameterized by the Markov state transition probability p, $0 < p < 1$.

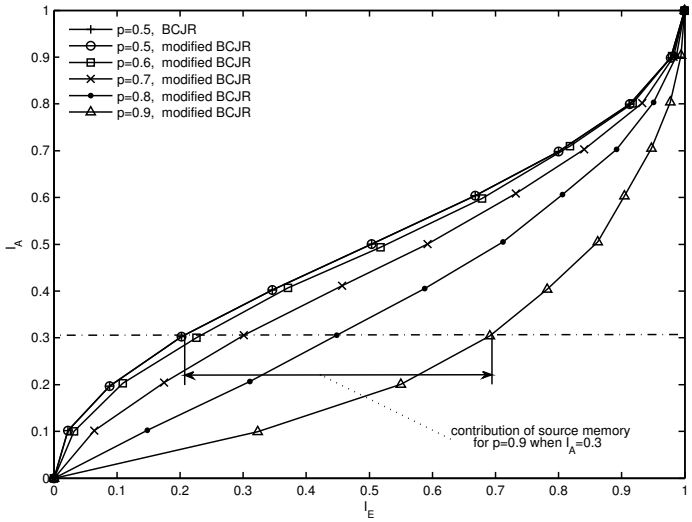


Fig. 3. Extrinsic information transfer characteristic of outer coder, $(G_r, G) = (3, 2)_8$

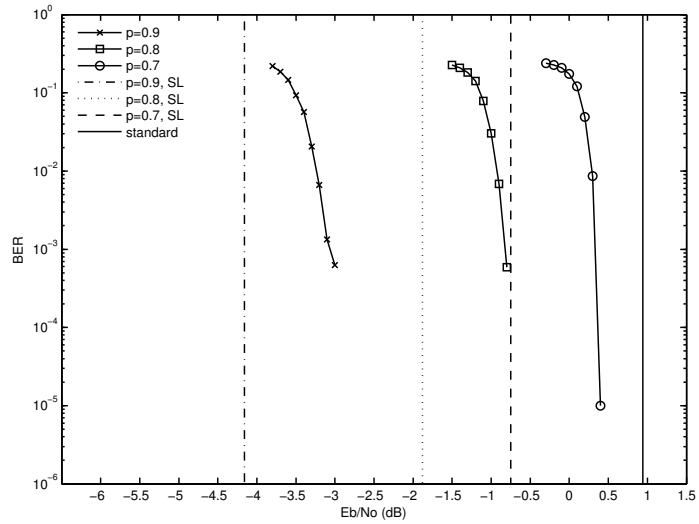


Fig. 5. BER performance of proposed system for Markov sources with different correlation

$p = 0.5$ indicates the memoryless source while $p > 0.5$ (or equivalently $p < 0.5$) indicates source with memory. Since the correlation is symmetric on p, we only consider the case $0.5 \leq p < 1$. The EXIT curves with standard BCJR and with modified BCJR exploiting source with different transition probabilities are illustrated in Fig. 3. The code we used in the simulations is memory-1 recursive systematic convolutional code with the generator polynomial $(G_r, G) = (3, 2)_8$.

As shown in Fig 2, the decoder D_1 of C_1 exploits a priori LLR $A(X_1)$. By using the modified BCJR algorithm, it generates extrinsic LLR $E(X_1)$. Hence the EXIT function of D_1 is defined as:

$$I_E(X_1) = T_{X_1}(I_A(X_1)), \tag{27}$$

where function $T_{X_1}(\cdot)$ was obtained by the histogram measurement [20].

For the source with $p = 0.5$, the EXIT curves with the standard BCJR decoder and our modified BCJR decoders are the same. For the Markov sources with different p ($p = 0.6, 0.7, 0.8, 0.9$), the EXIT curves obtained by using the modified BCJR decoder are pushed down and shifted to the right as p increase, indicating that larger extrinsic information can be obtained. These results are consistent with the consideration provided in Section III-B. It is also worth noticing that the contribution of source memory represented by the increase in extrinsic mutual information is larger when a priori input $I_A < 0.5$ than when $I_A > 0.5$, and the contribution becomes negligible when $I_A > 0.9$. It can also bee seen that for $p = 0.6$, the improvement is quite limited. Thus we will not consider this case in the following discussion.

2) Inner Code: D_2 calculates its extrinsic LLR int the same way as D_1, except that it has a direct connection to the Channel. Hence, its EXIT function is defined as:

$$I_E(U_2) = T_{U_2}(I_A(U_2), E_b/N_0). \tag{28}$$

Here we used memory-2 recursive convolutional code with the generator polynomial $(7, 4)_8$ for numerical evaluation. To obtain $T_{U_2}(\cdot)$, we also use the LLR histogram measurement.

B. Trajectory

Because the mutual information does not change after interleavering/deinterleavering, the following equality holds

$$I_A^{(l)}(X_1) = I_E^{(l-1)}(U_2), \tag{29}$$

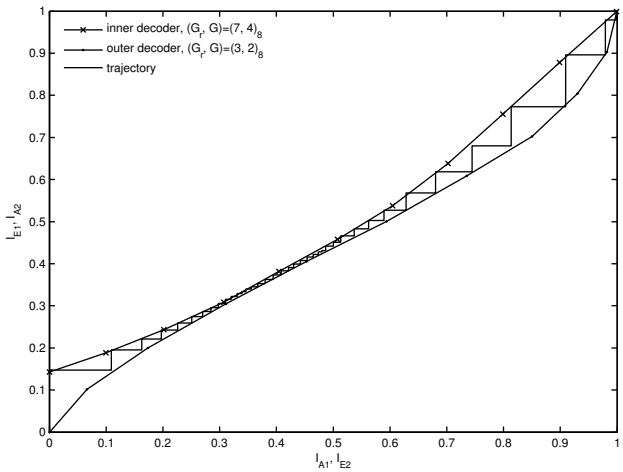
$$I_A^{(l)}(U_2) = I_E^{(l-1)}(X_1), \tag{30}$$

where l is the iteration index, i.e., the extrinsic information generated by the first decoder is used as the a priori information for the second decoder. In the chain simulation, we evaluated the extrinsic mutual information, iteration-by-iteration, and plotted the obtained mutual information, according to (29) and (30).

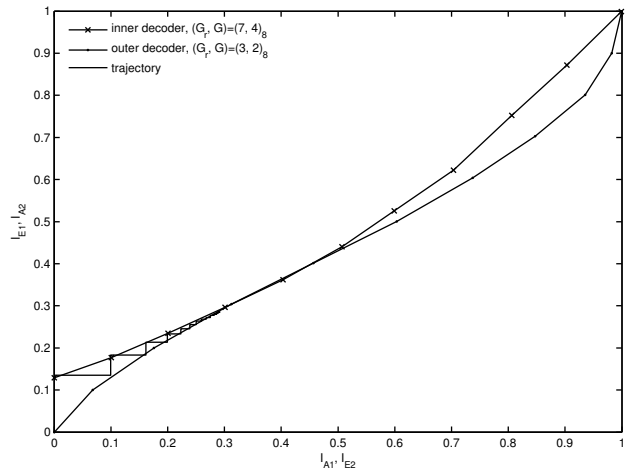
The EXIT chart and trajectory of our system for Markov source with different p values as a parameter are shown in Fig. 4 for $(2, 3)_8$ and $(7, 4)_8$ outer and inner codes, respectively. It is found from Fig. 4(a) that with $p = 0.7$ and $E_b/N_0 = 0.1$ dB, the convergence tunnel is still open until a point very close to $(1, 1)$ mutual information point, and the trajectory reaches the convergence point. The convergence behavior with $p = 0.7$ and $E_b/N_0 = 0$ dB is presented in Fig. 4(b), where E_b/N_0 is only 0.1 dB lower than the case shown in Fig. 4(a). It is found that the trajectory gets stuck with $E_b/N_0 = 0$ dB. This observation suggests that the convergence threshold is 0.1 dB for $p = 0.7$. The same observation can be drawn with $p = 0.8$ and $p = 0.9$, where the convergence threshold is -1.1 dB and -3.4 dB, respectively.

C. BER Performance Evaluation

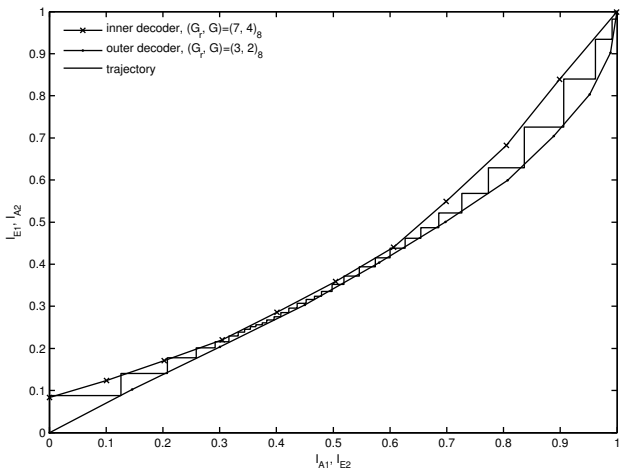
A series of simulations were conducted to evaluate bit error rate (BER) performance of our proposed technique. The length of the bit sequence transmitted through the channel is 10000



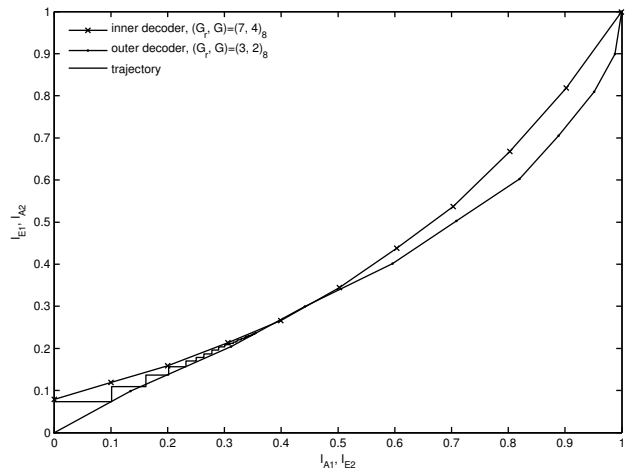
(a) $p = 0.7,\ E_b/N_0 = 0.1$ dB



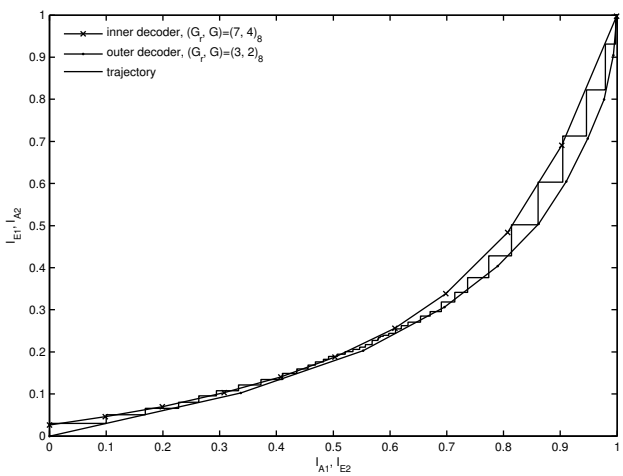
(b) $p = 0.7,\ E_b/N_0 = 0$ dB



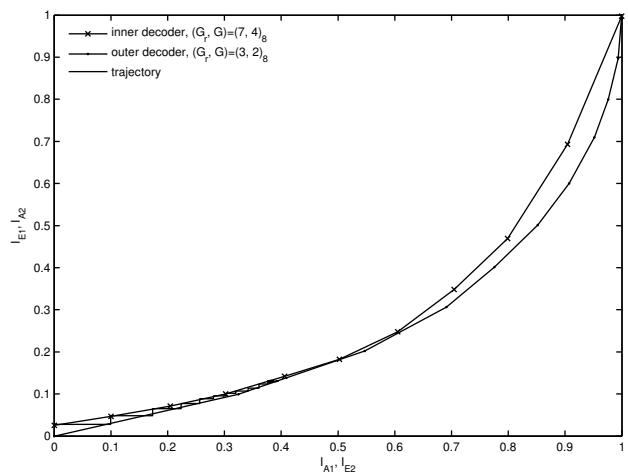
(c) $p = 0.8,\ E_b/N_0 = -1.1$ dB



(d) $p = 0.8,\ E_b/N_0 = -1.2$ dB



(e) $p = 0.9,\ E_b/N_0 = -3.4$ dB



(f) $p = 0.9,\ E_b/N_0 = -3.5$ dB

Fig. 4. EXIT chart of proposed system for Markov source with different p values.

bits, and 1000 different blocks were transmitted for the sake of keeping reasonable accuracy. From the convergence analysis provided in the previous subsection, we can see that when the E_b/N_0 is even a little larger than the threshold, narrow convergence tunnel opens until a point very close to the (1,1) mutual information point, as shown in Fig. 4(e). The iteration times were determined, depending on the trajectory simulation results shown in Fig. 4.

BER performance of the proposed JSC coding technique for Markov source with different correlation over AWGN channel is shown in Fig. 5. It can be seen that there are no error floors. Furthermore, the E_b/N_0 value, with which the turbo cliff happens, is exactly consistent with the convergence threshold presented in the previous subsection.

For the source without memory, the threshold E_b/N_0 for the serially concatenated codes with the same code parameters ($(3,2)_8$ inner and $(7,4)_8$ outer codes) is 0.94 dB [19], if it is decoded by the standard BCJR algorithm. For the source with memory, the limit $(E_b/N_0)_{lim}$ is determined from the condition $H(S)R \leq C$, where with the source-channel separation assumption, R specifies the code rate, $H(S)$ the entropy of the source, and $C = \frac{1}{2}\log_2(1 + \frac{2E_bR}{N_0})$ is the capacity of the AWGN channel in bits per channel use. These conditions yields

$$(\frac{E_b}{N_0})_{lim} = \frac{2^{2H(S)R} - 1}{2R}. \tag{31}$$

Using (31), we can obtain the Shannon limit for $p = 0.7, 0.8, 0.9$ and $R = 1/2$, which are -0.75 dB, -1.88 dB, and -4.16 dB, respectively. These limits are also depicted in Fig. 5.

It can be clearly observed from the threshold analysis shown in Fig. 4(e) that when $p = 0.9$, our system offers a gain of 4.34 $(= 0.94 - (-3.4))$ dB over the concatenated convolutional code with the same parameter decoded by the standard BCJR algorithm (Threshold is $E_b/N_0 = 0.94$ dB, as described above). The gap to the Shannon limit described above is 0.76 $(= -3.4 - (-4.16))$ dB. It should be noticed that the gains from the standard BCJR decoder and the gaps to the Shannon limits are different, depending on the correlation parameter p. The results are provided in Table I, togather with the results of JSCTC [1]. It can be found from the table that substaintial gains can be achieved for different p values for both systems over original parallel/serial concatenated codes, proposed in [1] and in this paper, respectively, indicating that exploiting the source memory in JSC scheme provides us with significant advantage. Intuitively, if the correlation of the source is stronger (p is larger), the gains will also becomes larger. This observation is consistent with Table I. However, for JSCTC, as p increase, the gaps to the Shannon limits also become larger, reflecting the fact that JSCTC can not fully exploit the source memory structure when the source correlation is strong. While our proposed system is more suitable in such high correlation scenarios. In general, besides the gap in the case of $p = 0.7$, our system outperforms JSCTC in terms of both gains over the standard BCJR decoding and gaps to the Shannon limits.

TABLE I
COMPARISON BETWEEN PROPOSED SYSTEM AND JSCTC FOR BER PERFORMANCE

Source Correlation	JSCTC		Our system	
	Gain(dB)	Gap(dB)	Gain(dB)	Gap(dB)
p=0.7	0.45	0.73	0.84	0.85
p=0.8	1.29	0.94	2.04	0.78
p=0.9	3.03	1.36	4.34	0.76

It should also be noticed here that our system employs a memory-1 outer code and a memory-2 inner code, and the extrinsic information obtained by the decoders can be exchanged directly between two decoder, while JSCTC shown in [1] uses two memory-4 constituent codes and extrinsic LLR obtained by the decoders have to be modified before being fed into another decoder. Hence, our proposed system requires much less complexity compared with the JSCTC technique shown in [1].

V. CONCLUSION

In this paper we have investigated the design of serially concatenated JSC codes for symmetric binary Markov source. To fully exploit the Markov source memory, the standard BCJR algorithm is modified, and the modified version of the BCJR algorithm is used in the outer decoder, while the standard BCJR algorithm is used in the inner decoder. The extrinsic LLR obtained by the inner/outer decoder can be exchanged between the constituent decoders via interleaving/deinterleavering, just in the same way as the decoding of serially concatenated codes is performed. We have also verified the superiority of the proposed technique over the JSCTC proposed in [1] by EXIT analysis as well as trajectory and BER simulations. It has been shown that our system provides significant improvements in both gains over the standard BCJR decoder and the gaps to the Shannon limits by requiring only minor increase in computational burden.

REFERENCES

[1] G. Zhu and F. Alajaji, "Joint source-channel turbo coding for binary markov sources," *IEEE Transactions on Wireless Communications*, vol. 5, no. 5, pp. 1065–1075, May 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1633359

[2] F. Hekland, "A review of joint source-channel coding," Technical Report, Norwegian University of Science and Technology, Tech. Rep., 2004.

[3] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–2457, Sep. 1995. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=412719

[4] J. Garcia-Frias and J. Villasenor, "Combining hidden markov source models and parallel concatenated codes," *IEEE Communications Letters*, vol. 1, no. 4, pp. 111–113, Jul. 1997. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=602600

[5] ——, "Joint turbo decoding and estimation of hidden markov sources," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, pp. 1671–1679, Sep. 2001. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=947032

[6] L. Yin, J. Lu, and Y. Wu, "LDPC-based joint source-channel coding scheme for multimedia communications," in *Proceedings of the The 8th International Conference on Communication Systems - Volume 01*, ser. ICCS '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 337–341. [Online]. Available: http://portal.acm.org/citation.cfm?id=1259590.1260052

[7] ——, "Combined hidden markov source estimation and low-density parity-check coding: a novel joint source-channel coding scheme for multimedia communications," *Wireless Communications and Mobile Computing*, vol. 2, no. 6, pp. 643–650, Sep. 2002. [Online]. Available: http://doi.wiley.com/10.1002/wcm.86

[8] M. Fresia, F. Pérez-Cruz, and H. V. Poor, "Optimized concatenated ldpc codes for joint source-channel coding," in *Proceedings of the 2009 IEEE international conference on Symposium on Information Theory - Volume 3*, ser. ISIT'09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 2131–2135. [Online]. Available: http://portal.acm.org/citation.cfm?id=1701116.1701256

[9] I. Ochoa, P. M. Crespo, and M. Hernaez, "LDPC codes for non-uniform memoryless sources and unequal energy allocation," *IEEE Communication Letters*, vol. 14, pp. 794–796, September 2010. [Online]. Available: http://dx.doi.org/10.1109/LCOMM.2010.09.100662

[10] G. Zhu and F. Alajaji, "Turbo codes for nonuniform memoryless sources over noisy channels," *IEEE Communications Letters*, vol. 6, no. 2, pp. 64–66, Feb. 2002. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=984695

[11] F. Cabarcas, R. Souza, and J. Garcia-Frias, "Source-controlled turbo coding of non-uniform memoryless sources based on unequal energy allocation," in *International Symposium on Information Theory, 2004. ISIT 2004. Proceedings.*, Chicago, Illinois, USA, 2004, pp. 164–164. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1365201

[12] G. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Transmission of nonuniform memoryless sources via nonsystematic turbo codes," *IEEE Transactions on Communications*, vol. 52, no. 5, pp. 855–855, May 2004. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1299079

[13] F. Cabarcas, R. Souza, and J. Garcia-Frias, "Turbo coding of strongly nonuniform memoryless sources with unequal energy allocation and PAM signaling," *IEEE Transactions on Signal Processing*, vol. 54, no. 5, pp. 1942–1946, May 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1621424

[14] V. Tervo, T. Matsumoto, and J. Karjalainen, "Joint Source-Channel coding using multiple label mapping," in *2010 IEEE 72nd Vehicular Technology Conference - Fall*, Ottawa, ON, Canada, Sep. 2010, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5594170

[15] J. D. Ser, P. M. Crespo, I. Esnaola, and J. Garcia-Frias, "Joint Source-Channel coding of sources with memory using turbo codes and the Burrows-Wheeler transform," *IEEE Transactions on Communications*, vol. 58, no. 7, pp. 1984–1992, Jul. 2010. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5504599

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

[17] 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, 1974.

[18] W. E. Ryan, "Concatenated convolutional codes and iterative decoding," *Wiley Encyclopedia in Telecommunications*, 2003.

[19] S. ten Brink, "Code characteristic matching for iterative decoding of serially concatenated codes," *Annals of Telecommunications*, vol. 56, no. 7, pp. 394–408, 2001.

[20] ——, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.