www.ietdl.org

Published in IET Communications
Received on 30th March 2011
Revised on 20th March 2012
doi: 10.1049/iet-com.2011.0268

ISSN 1751-8628

Symbol-based iterative decoding of convolutionally encoded multiple descriptions

C.-F. Wu W.-W. Chang

Institute of Communications Engineering, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsinchu, Taiwan
E-mail: cfwu.cm95@nctu.edu.tw

Abstract: Transmission of convolutionally encoded multiple descriptions over noisy channels can benefit from the use of iterative source-channel decoding. The authors first modified the BCJR algorithm in a way that symbol a posteriori probabilities can be derived and used as extrinsic information to improve the iterative decoding between the source and channel decoders. The authors also formulate a recursive implementation for the source decoder that processes reliability information received on different channels and combines them with inter-description correlation to estimate the transmitted quantiser index. Simulation results are presented for two-channel scalar quantisation of Gauss–Markov sources which demonstrate the error-resilience capabilities of symbol-based iterative decoding.

1 Introduction

With the rapid development of wireless multimedia communications, reliable transmission of speech and video signals over bandlimited noisy channels is becoming more and more widespread. Multiple description (MD) coding [1] is a method of representing a source with multiple correlated descriptions such that any subset of the descriptions can be used to decode the source with a fidelity that increases with the number of received descriptions. The output symbols of an MD encoder exhibit considerable residual redundancy in terms of both non-uniformity of distribution and their dependencies. This redundancy is because of the non-optimality of the practically designed source encoder in the presence of complexity and delay constraints, or by path diversity as a result of MD coding. The ability to exploit path diversity and source residual redundancy for error robustness makes MD coding an attractive option for the multimedia transmission over unreliable IP networks. A typical example is the MD scalar quantisation (MDSQ) [1, 2] which splits source samples into two descriptions by using scalar quantiser and index assignment. The index assignment can be represented by a mapping of each reproduction level of the scalar quantiser to a unique element in an index assignment matrix. The choice of the index assignment matrix determines the correlation between the descriptions and is key to realise an MDSQ. Design algorithms for good index assignments are presented in [2]. In these, the inter-description correlation is controlled by choosing the number of diagonals covered by the index assignment. The MDSQ has been extensively studied for noiseless channels with packet loss, assuming that there exists multiple independent channels that either provide error-free transmission or experience total failure. In many practical situations, however, MDs of the source signals are transmitted over channels subject to noise and packet loss. In this work, we account for these random noises by representing the non-erasure state of the channel using an additive white Gaussian noise (AWGN) channel. Some recent works have been presented in [3–5], where sequential Monte Carlo methods are used to perform signal processing in the non-Gaussian additive noise scenario.

For MD communication over noisy channels, a channel encoder may be used on each description to deal with random bit errors. When the MDSQ is concatenated with convolutional codes, iterative source-channel decoding (ISCD) [6, 7] inspired by turbo principle has been shown effective using the source residual redundancy and assisted with the reliability information provided by the soft-output channel decoder. In the so-called MD-ISCD schemes [8, 9], source residual redundancy and channel-code redundancy are exploited alternatively by exchanging extrinsic information between the constituent decoders. An iterative decoder consisting of two maximum a posteriori probability (MAP) detectors is proposed in [8] for joint decoding of MDSQ and convolutional codes. In [9], a cross decoding strategy was stated that exploits not only the reliability information of every bit in one description, but also the extrinsic information from the other description according to the chosen index assignment. In the decoding procedure, MAP detectors operating on soft channel outputs were used for each of the two descriptions in such a way that the output of one MAP detector is combined with inter-description correlation to compute the a priori information for the other detector.

With respect to an implementation of MD-ISCD, previous works in [8, 9] are expected to show two limitations. Firstly, as the source decoder uses two separate MAP detectors with each detector operating on one description, it may report

invalid codeword combinations corresponding to the empty cells of the index assignment matrix. In dealing with such situations, an invalid codeword combination is treated as an uncorrectable error and the mean of the source is reconstructed. Secondly, the major part of the iterative decoding process runs on bit-level, but the source decoder itself is realised on symbol level. This is in part due to the fact that binary convolutional codes are commonly used, so the soft-output channel decoding can be implemented efficiently by the BCJR algorithm [10, 11]. It causes the problem that only bitwise source a priori knowledge can be exploited by the channel decoder, since the BCJR algorithm is derived based on a bit-level code trellis. For the purpose of applicability, it requires the symbol-to-bit and bit-to-symbol probability conversion in each passing of the extrinsic information between the source and channel decoders. This processing step destroys the bit-correlations within a symbol, thus reducing the effectiveness of iterative decoding. Recognising this, we focus on symbol-based trellis decoding algorithms throughout this paper since they allows to exchange between the source and channel decoders the whole symbol extrinsic information. The first step toward realisation is to use sectionalised code trellises rather than bit-level trellises as the bases for soft-output channel decoding of binary convolutional codes. Performance is further improved by using a joint MAP source decoder that processes reliability information received on different channels and combines them with inter-description correlation to provide a better estimate of the transmitted quantiser index.

2 System description

The two-channel transmission of autocorrelated sources over AWGN channels is considered, in which an MDSQ is used for source coding and convolutional codes are used for channel coding of individual descriptions. Fig. 1 shows our model of a two-channel transmission system. The MDSQ encoder can be decomposed into a scalar quantiser followed by an index assignment. Suppose at time t, the input sample v_t is quantised by the M-bit index u_t that, after index assignment, is represented by two descriptions $u_{I,t} = \delta_I(u_t)$ and $u_{J,t} = \delta_J(u_t)$ at an average rate of R bits per symbol per channel. For the scalar quantiser, the reproduction level corresponding to an index $u_t = l$ is denoted by c_l, where $l \in \Gamma = \{0,1, \ldots, 2^M - 1\}$. We can generally assume that there is a certain amount of residual redundancy remaining in the index sequence due to delay and complexity constraints for the quantisation stage. In the following, the time-correlations of quantiser indexes are modelled by a first-order stationary Markov process with index-transition probabilities $P(u_t|u_{t-1})$. The two descriptions resulting from MDSQ can be interpreted as the row and column indexes of an $2^R \times 2^R$ matrix, in which the $(\delta_I(l), \delta_J(l))$th location is placed with a specified quantiser index $u_t = l$. Since $2^M < 2^{2R}$, these two descriptions contain redundancy and the correlation properties of each possible pair $(\delta_I(l), \delta_J(l))$ can be computed from the knowledge of index assignment. The amount of inter-description correlation decreases as M becomes larger and more diagonals are occupied by the quantiser indexes. In describing the index assignment matrix, let $R_k = \{l|\delta_I(l) = k\}$ and $C_m = \{l|\delta_J(l) = m\}$ represent the subset of quantiser indexes located in row k and in column m of the matrix, respectively. We will denote the output symbols of MDSQ by $u_{D,t}$, where for simplicity, $D \in \{I, J\}$ stands for one of the descriptions. After MDSQ encoding a block of T symbols of description D, written as $\boldsymbol{U}_{D,1}^{\mathrm{T}} = (u_{D,1}, \ldots, u_{D,t}, \ldots, u_{D,T})$, are interleaved by a symbol interleaver Φ. The interleaved symbol sequence, denoted by $\boldsymbol{X}_{D,1}^{\mathrm{T}} = (x_{D,1}, \ldots, x_{D,t}, \ldots, x_{D,T})$, is then processed by a binary convolutional channel encoder with a code rate of 1/2. If a systematic channel encoder is used, the codeword corresponding to each symbol $x_{D,t}$ can be written as $y_{D,t} = \{x_{D,t}, z_{D,t}\}$, where $x_{D,t}$ and $z_{D,t}$ represent the systematic and parity symbol of the code, respectively. The code sequences are modulated with a binary phase shift keying modulator and then transmitted over an AWGN channel. For brevity, denote the input and output sequence of the AWGN channel by $\boldsymbol{Y}_{D,1}^{\mathrm{T}} = \{\boldsymbol{X}_{D,1}^{\mathrm{T}}, \boldsymbol{Z}_{D,1}^{\mathrm{T}}\}$ and $\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}} = \{\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}\}$, respectively.

Goal of the MD-ISCD is to jointly exploit the channel information and source a priori information for improved estimation of the transmitted quantiser index. When MDSQ is concatenated with a channel coder, the turbo-like evaluation of source residual redundancy and of artificial channel-code redundancy makes step-wise quality gains possible by iterative decoding. As shown in Fig. 2, the receiver consists of two separate channel decoders and an MD source decoder with soft-inputs and soft-outputs (SISO). In the first step, identical for both descriptions, a channel decoder processes the received code sequence $\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}$ and combines them with source a priori information to compute the extrinsic information $L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t})$ on individual systematic symbol $x_{D,t}$. The MD-SISO source decoder combines the extrinsic information provided by the two channel decoders to provide a more accurate estimate of the transmitted quantiser index. Joint decoding of the two descriptions produces the a posteriori probability (APP) for each of possibly transmitted quantiser index u_t, which is denoted by $P(u_t|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}})$. Given the knowledge of the index assignment and source encoder statistics, it also generates a symbol-level extrinsic information $L_{\mathrm{SD}}^{[\mathrm{ext}]}(u_{D,t})$ for each description which will be used as a priori information of the corresponding channel

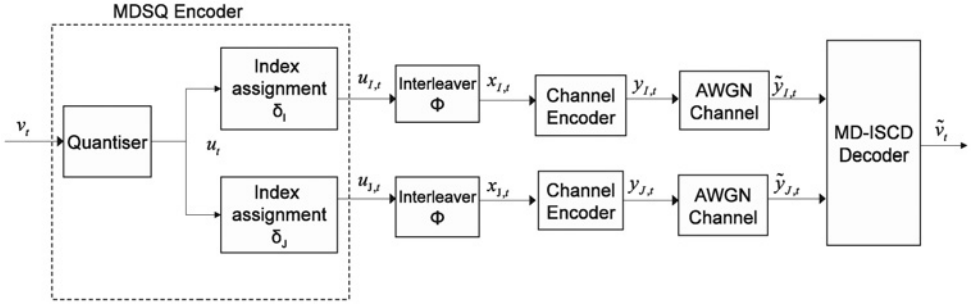

Fig. 1 *Block diagram of a two-channel MD communication system*

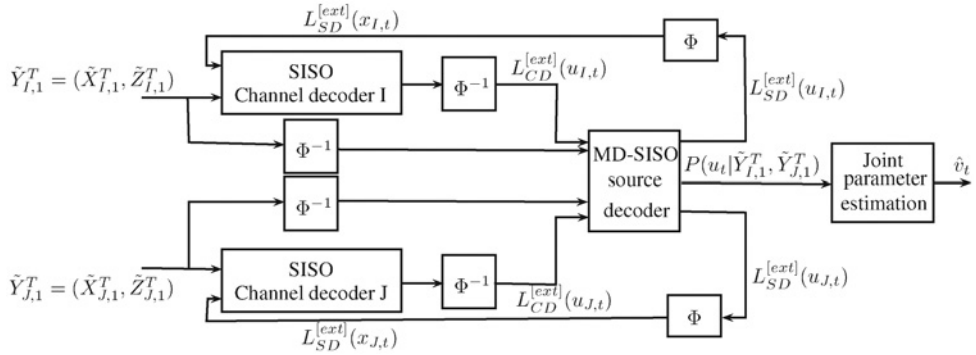

Fig. 2 *MD-ISCD scheme for the concatenation of MDSQ and convolutional codes*

decoder in the next iteration. Exchanging extrinsic information between the source and channel decoders is iteratively repeated until the reliability gain becomes insignificant. After the last iteration, the index APPs are used to determine the MAP signal estimates as follows

$$\hat{v}_t = c_{l^*}, \quad l^* = \max_{l \in \Gamma} P(u_t = l | \tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}} \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) \tag{1}$$

3 Symbol decoding of binary convolutional codes

For transmission schemes with channel coding, a SISO channel decoders may be used to provide both estimated bits and their reliability information for further processing. In many practical systems bit-based channel decoders are used for which the BCJR algorithm [10] is available for efficient decoding of binary convolutional codes. Proper sectionalisation of a bit-level code trellis [12] allows us to devise MAP symbol decoding algorithms which takes into account the symbol-level extrinsic information obtained from the source decoder. To proceed with this, we propose a modified BCJR algorithm which parses the received code-bit sequence into R-bit blocks and computes the APP on a symbol-by-symbol basis. By parsing the code-bit sequence into R-bit symbols, we are in essence merging R stages of the original bit-level code trellis into one. Having defining the trellis structure as such, there will be 2^R branches leaving and entering each state and each branch is associated with one APP corresponding to a particular symbol. For convenience, we say that the sectionalised trellis diagram forms a finite-state machine defined by its state transition function $F_s(x_{D,t}, s_t)$ and output function $F_p(x_{D,t}, s_t)$. Specifically, the code-symbol associated with the branch from state s_t to state $s_{t+1} = F_s(x_{D,t}, s_t)$ can be written as $y_{D,t} = (x_{D,t}, z_{D,t})$, where $z_{D,t} = F_p(x_{D,t}, s_t)$ is the parity symbol given state s_t and systematic symbol $x_{D,t}$.

We next apply sectionalised code trellises to formulate a recursive implementation for computing the APP of a systematic symbol $x_{D,t}$, given the received code sequence $\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}} = \{\tilde{y}_{D,1}, \tilde{y}_{D,2}, \ldots, \tilde{y}_{D,T}\}$. Let $l_D = \delta_D(l)$ represent the symbol of description D corresponding to a specified quantiser index $x_t = l$. Taking the trellis state s_t into consideration, we rewrite the symbol APP as follows

$$\begin{aligned} P(x_{D,t} = l_D | \tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}) &= C \sum_{s_t} P(x_{D,t} = l_D, s_t, \tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}) / P(\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}) \\ &= C \sum_{s_t} \alpha_t^x(l_D, s_t) \beta_t^x(l_D, s_t) / P(\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}) \end{aligned} \tag{2}$$

where $\alpha_t^x(l_D, s_t) = P(x_{D,t} = l_D, s_t, \tilde{\boldsymbol{Y}}_{D,1}^t)$ and $\beta_t^x(l_D, s_t) = P(\tilde{\boldsymbol{Y}}_{D,t+1}^{\mathrm{T}} | x_{D,t} = l_D, s_t, \tilde{\boldsymbol{Y}}_{D,1}^t)$. For the recursive implementation, the forward and backward recursions are to compute the following metrics:

$$\begin{aligned} \alpha_t^x(l_D, s_t) &= \sum_{s_{t-1}} \sum_{k_D} P(x_{D,t} = l_D, s_t, x_{D,t-1} = k_D, s_{t-1}, \tilde{y}_{D,t} \tilde{\boldsymbol{Y}}_{D,1}^{t-1}) \\ &= \sum_{s_{t-1}} \sum_{k_D} \alpha_{t-1}^x(k_D, s_{t-1}) \gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) \end{aligned} \tag{3}$$

$$\begin{aligned} \beta_t^x(l_D, s_t) &= \sum_{s_{t+1}} \sum_{k_D} P(\tilde{\boldsymbol{Y}}_{D,t+1}^T, s_{t+1}, x_{D,t+1} = k_D | x_{D,t} = l_D, s_t, \tilde{\boldsymbol{Y}}_{D,1}^t) \\ &= \sum_{s_{t+1}} \sum_{k_D} \beta_{t+1}^x(k_D, s_{t+1}) \gamma_{k_D,l_D}(\tilde{y}_{D,t+1}, s_{t+1}, s_t) \end{aligned} \tag{4}$$

where

$$\begin{aligned} &\gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) \\ &\quad = P(x_{D,t} = l_D, s_t, \tilde{y}_{D,t} | x_{D,t-1} = k_D, s_{t-1}, \tilde{\boldsymbol{Y}}_{D,1}^{t-1}) \\ &\quad = P(s_t | x_{D,t-1} = k_D, s_{t-1}) P(x_{D,t} = l_D | x_{D,t-1} = k_D) \\ &\qquad \times P(\tilde{y}_{D,t} | x_{D,t} = l_D, s_t) \end{aligned} \tag{5}$$

and

$$\begin{aligned} &\gamma_{k_D,l_D}(\tilde{y}_{D,t+1}, s_{t+1}, s_t) \\ &\quad = P(s_{t+1} | x_{D,t} = l_D, s_t) P(x_{D,t+1} = k_D | x_{D,t} = l_D) \\ &\qquad \times P(\tilde{y}_{D,t+1} | x_{D,t+1} = k_D, s_{t+1}) \end{aligned} \tag{6}$$

The computation of the branch metric $\gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1})$ can be further simplified as follows. First, making use of the merged code trellis, the value of $P(s_t | x_{D,t-1} = k_D, s_{t-1})$ is either one or zero depending on whether symbol k_D is associated with transition from state s_{t-1} to state $s_t = F_s(x_{D,t-1} = k_D, s_{t-1})$. The second term in (5) is reduced to $P(x_{D,t} = l_D)$ under the assumption that $x_{D,t}$ is uncorrelated with $x_{D,t-1}$, which is indeed the case as $x_{D,t}$ is the interleaved version of symbols $u_{D,t}$. For AWGN channels, the third term in (5) can be computed by

$$\begin{aligned} P(\tilde{y}_{D,t} | x_{D,t} = l_D, s_t) &= P(\tilde{x}_{D,t} | x_{D,t} = l_D) \\ &\quad \times P(\tilde{z}_{D,t} | z_{D,t} = F_p(x_{D,t} = l_D, s_t)) \end{aligned} \tag{7}$$

doi: 10.1049/iet-com.2011.0268

The MAP algorithm is likely to be considered too complex for real-time implementation in a practical system. To avoid the number of complicated operations and also numerical representation problems, realisations of the MAP algorithm in the logarithmic domain have been proposed in [13, 14]. We define the reliability of each non-zero symbol $x_{D,t} = l_D$, $l_D = 1, 2, \ldots, 2^R - 1$, with respect to $x_{D,t} = 0$, by considering log-likelihood ratio (LLR) of the following type

$$L(x_{D,t} = l_D | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}}) = \log \frac{P(x_{D,t} = l_D | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}})}{P(x_{D,t} = 0 | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}})} \tag{8}$$

This definition for the LLR values allow for easy conversion between the a posteriori LLRs and APPs. The next step is to reduce the large computational burden complexity which is required for computing the logarithmic values of the $\alpha_t^x(l_D, s_t)$ and $\beta_t^x(l_D, s_t)$ terms in (2). This task can be accomplished by using the Jacobian logarithm function [14] defined by the property

$$\log(\mathrm{e}^{\delta_1} + \mathrm{e}^{\delta_2}) = \max\{\delta_1, \delta_2\} + \log(1 + \mathrm{e}^{-|\delta_2 - \delta_1|}) \tag{9}$$

For brevity, we use the following shorthand notation $\max_j^*\{\delta_j\} = \log(\sum_{j=1}^{l} \mathrm{e}^{\delta_j})$. By taking the logarithm of $\alpha_t^x(l_D, s_t)$ in (3), we have

$$\begin{aligned}\hat{\alpha}_t^x(l_D, s_t) &= \log \alpha_t^x(l_D, s_t) \\ &= \max_{s_{t-1}}^* \max_{k_D}^* \{\hat{\alpha}_{t-1}^x(k_D, s_{t-1}) + \hat{\gamma}_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1})\}\end{aligned} \tag{10}$$

and similarly

$$\begin{aligned}\hat{\beta}_t^x(l_D, s_t) &= \log \beta_t^x(l_D, s_t) \\ &= \max_{s_{t+1}}^* \max_{k_D}^* \{\hat{\beta}_{t+1}^x(k_D, s_{t+1}) + \hat{\gamma}_{k_D,l_D}(\tilde{y}_{D,t+1}, s_{t+1}, s_t)\}\end{aligned} \tag{11}$$

$$\begin{aligned}\hat{\gamma}_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) &= \log \gamma_{l_D,k_D}(\tilde{y}_{D,t}, s_t, s_{t-1}) \\ &= \log P(s_t | x_{D,t-1} = k_D, s_{t-1}) \\ &\quad + \log P(x_{D,t} = l_D) \\ &\quad + \log P(\tilde{x}_{D,t} | x_{D,t} = l_D) \\ &\quad + \log P(\tilde{z}_{D,t} | z_{D,t} = F_p(x_{D,t} = l_D, s_t))\end{aligned} \tag{12}$$

An iterative process using the log-MAP channel decoder as a constituent decoder is realisable, if the a posteriori LLR $L(x_{D,t} = l_D | \tilde{Y}_{D,1}^{\mathrm{T}})$ can be separated into three additive terms: the a priori term $L_a(x_{D,t} = l_D) = \log[P(x_{D,t} = l_D)/P(x_{D,t} = 0)]$, the channel-related term $L_c(x_{D,t} = l_D) = \log[P(\tilde{x}_{D,t} | x_{D,t} = l_D)/P(\tilde{x}_{D,t} | x_{D,t} = 0)]$, and an extrinsic term $L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t} = l_D)$. Substituting (10) and (12) into (8) leads to

$$\begin{aligned}L(x_{D,t} = l_D | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}}) &= L_a(x_{D,t} = l_D) \\ &\quad + L_c(x_{D,t} = l_D) + L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t} = l_D)\end{aligned} \tag{13}$$

with the extrinsic LLR

$$\begin{aligned}&L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t} = l_D) \\ &= \max_{s_t}^* \{\log P(\tilde{z}_{D,t} | x_{D,t} = l_D, s_t) + \beta_t^x(l_D, s_t)\} \\ &\quad + \max_{s_{t-1}}^* \{\max_{k_D}^* \{\log P(s_t | x_{D,t-1} = k_D, s_{t-1}) + \alpha_{t-1}^x(k_D, s_{t-1})\}\} \\ &\quad - \max_{s_t}^* \{\log P(\tilde{z}_{D,t} | x_{D,t} = 0, s_t) + \beta_t^x(0, s_t)\} \\ &\quad + \max_{s_{t-1}}^* \{\max_{k_D}^* \{\log P(s_t | x_{D,t-1} = 0, s_{t-1}) + \alpha_{t-1}^x(k_D, s_{t-1})\}\}\end{aligned} \tag{14}$$

Notice that the a priori LLR in (13) is initialised to be $L_a(x_{D,t} = l_D)$ in terms of the source distribution $P(x_{D,t} = l_D)$. Within iterations the precision of the APP estimation can be enhanced by replacing $L_a(x_{D,t} = l_D)$ with the interleaved extrinsic LLR $L_{\mathrm{SD}}^{[\mathrm{ext}]}(x_{D,t} = l_D)$ provided by the source decoder. Therefore the extrinsic LLR resulting from the channel decoding can be calculated by

$$\begin{aligned}L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t} = l_D) &= L(x_{D,t} = l_D | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}}) \\ &\quad - L_{\mathrm{SD}}^{[\mathrm{ext}]}(x_{D,t} = l_D) - L_c(x_{D,t} = l_D)\end{aligned} \tag{15}$$

and used as new a priori information for the source decoder.

4 MD-SISO source decoder

Goal of the MD-SISO source decoder is to compute the APPs of transmitted quantiser indexes by jointly exploiting the channel information, the source residual redundancy and the inter-description correlation induced by the MDSQ. In previous work related to this problem [8, 9], the source decoder uses two separate MAP detectors with each detector operating on a single description $\tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}}$ to compute the APP $P(u_{D,t} | \tilde{\mathbf{Y}}_{D,1}^{\mathrm{T}})$ for a decoded systematic symbol $u_{D,t} = l_D$. Afterwards the source decoder makes an MAP decision on the two symbols $\{l_I^*, l_J^*\}$ and uses their combination to locate the corresponding quantiser index from the index assignment matrix. As the two MAP symbol estimates are decoded separately, it may report invalid codeword combinations corresponding to the empty cells of the index assignment matrix. To compensate for this shortage, we propose a joint MAP decoding algorithm which combines reliability information received on different channels and computes the APP for each of possibly transmitted quantiser index $u_t = l$. For the purpose of applicability, the algorithm of MD-SISO source decoding is separated into two parts. The first algorithmic step consists in the computation of the APP $P(u_t | \tilde{\mathbf{Y}}_{I,1}^{\mathrm{T}}, \tilde{\mathbf{Y}}_{J,1}^{\mathrm{T}})$ for a decoded quantiser index u_t, given the two received code-symbol sequences $\{\tilde{\mathbf{Y}}_{I,1}^{\mathrm{T}}, \tilde{\mathbf{Y}}_{J,1}^{\mathrm{T}}\}$. In the second step, identical for both descriptions, these index APPs are combined with a priori knowledge of the index assignment to extract the extrinsic information $L_{\mathrm{SD}}^{[\mathrm{ext}]}(u_{D,t})$ on every symbol $u_{D,t}$ of description D. It contains the new part of information resulting from MD-SISO source decoding and will be delivered back to the corresponding channel decoder as new a priori information for the next iteration.

The source decoding algorithm starts by computing the APP for a decoded quantiser index $u_t = l$ as follows

$$P(u_t = l|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) = P(u_t = l, \tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}})/P(\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) \tag{16}$$

Since the received sequence of systematic symbols are de-interleaved and then processed by the source decoder, we have $P(u_t = l, \tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) = P(u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}})$, where $\tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}} = \Phi^{-1}(\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})$. These probabilities can be further decomposed by using the Bayes theorem as

$$\begin{aligned}
&P(u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}}) \\
&\quad = P(u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}) P(\tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}}|u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}) \\
&\quad = \alpha_t^u(l)\beta_t^u(l) \prod_{D\in\{I,J\}} P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|u_{D,t} = l_D, \tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}})
\end{aligned} \tag{17}$$

where $\alpha_t^u(l) = P(u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{t}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{t}})$ and $\beta_t^u(l) = P(\tilde{\boldsymbol{U}}_{I,t+1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,t+1}^{\mathrm{T}}|u_t = l, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{t}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{t}})$. Using the Markov property of the indexes and the memoryless assumption of the channel, the forward-backward recursions of the algorithm in the logarithmic domain can be expressed as

$$\begin{aligned}
\hat{\alpha}_t^u(l) &= \log \alpha_t^u(l) \\
&= \log \sum_k P(u_t = l, u_{t-1} = k, \tilde{\boldsymbol{U}}_{I,1}^t, \tilde{\boldsymbol{U}}_{J,1}^t) \\
&= \log \sum_k P(\tilde{u}_{I,t}, \tilde{u}_{J,t}|u_t = l, u_{t-1} = k, \tilde{\boldsymbol{U}}_{I,1}^{t-1}, \tilde{\boldsymbol{U}}_{J,1}^{t-1}) \\
&\quad \times P(u_t = l|u_{t-1} = k, \tilde{\boldsymbol{U}}_{I,1}^{t-1}, \tilde{\boldsymbol{U}}_{J,1}^{t-1}) P(u_{t-1} = k, \tilde{\boldsymbol{U}}_{I,1}^{t-1}, \tilde{\boldsymbol{U}}_{J,1}^{t-1}) \\
&= \max_k{}^*\{\hat{\gamma}_{l,k}^t(\tilde{u}_{I,t}, \tilde{u}_{J,t}) + \hat{\alpha}_{t-1}^u(k)\}
\end{aligned} \tag{18}$$

$$\begin{aligned}
\hat{\beta}_t^u(l) &= \log \beta_t^u(l) \\
&= \log \sum_k P(u_t = l, u_{t+1} = k, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}})/ \\
&\quad \times P(u_t = l, \tilde{\boldsymbol{U}}_{I,1}^t, \tilde{\boldsymbol{U}}_{J,1}^t) \\
&= \max_k{}^*\{\hat{\gamma}_{k,l}^{t+1}(\tilde{u}_{I,t+1}, \tilde{u}_{J,t+1}) + \hat{\beta}_{t+1}^u(k)\}
\end{aligned} \tag{19}$$

and in (18)

$$\begin{aligned}
\hat{\gamma}_{l,k}^t(\tilde{u}_{I,t}, \tilde{u}_{J,t}) &= \log P(\tilde{u}_{I,t}|u_{I,t} = l_I) + \log P(\tilde{u}_{J,t}|u_{J,t} = l_J) \\
&\quad + \log P(u_t = l|u_{t-1} = k)
\end{aligned} \tag{20}$$

With these metrics, the a posteriori LLR corresponding to the index APP $P(u_t = l|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}})$ can be expressed as

$$\begin{aligned}
L(u_t = l|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) &= \hat{\alpha}_t^u(l) + \hat{\beta}_t^u(l) - \hat{\alpha}_t^u(0) - \hat{\beta}_t^u(0) \\
&\quad + \sum_{D\in\{I,J\}} \{L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t} = l_D) - L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t} = 0_D)\}
\end{aligned} \tag{21}$$

In the next step, the APP of each decoded symbol in every description is calculated from the temporary values of the index APPs and used for computing the extrinsic information of the source decoder. From the properties of the index assignment matrix, this task was accomplished by summing together the APPs of quantiser indexes being assigned to a certain description. For example, the APP for a decoded symbol $u_{I,t} = l_I$ of description I is given by

$$P(u_{I,t} = l_I|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) = \sum_{n\in R_{l_I}} P(u_t = n|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) \tag{22}$$

where $R_{l_I} = \{n|\delta_I(n) = l_I\}$ represents the subset of quantiser indexes located in column l_I of the matrix. Substituting (18) and (20) into (22) leads to

$$\begin{aligned}
&\log P(u_{I,t} = l_I|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) \\
&\quad = \log P(\tilde{u}_{I,t}|u_{I,t} = l_I) + \log P(\tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}|u_{I,t} = l_I, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}}) \\
&\qquad + \max_{n\in R_{l_I}}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = n_J) \\
&\qquad + \log P(\tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}}|u_{J,t} = n_J, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}) \\
&\qquad + \hat{\beta}_t^u(n) + \max_k{}^*\{\log P(u_t = n|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\}
\end{aligned} \tag{23}$$

This allows us to decompose the a posteriori LLR $L(u_{I,t} = l_I|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}})$ into three additive terms: a priori term $L_{\mathrm{a}}(u_{I,t})$, the channel-related term $L_{\mathrm{c}}(u_{I,t} = l_I)$, and an extrinsic term $L_{\mathrm{SD}}^{[\mathrm{ext}]}(u_{I,t} = l_I)$. In order to determine each of the three terms, we rewrite (23) in log-likelihood algebra as

$$\begin{aligned}
L(u_{I,t} = l_I|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}) &= \log \frac{\sum_{n\in R_{l_I}} P(u_t = n|\tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T})}{\sum_{m\in R_0} P(u_t = m|\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}})} \\
&= L_{\mathrm{a}}(u_{I,t} = l_I) + L_{\mathrm{c}}(u_{I,t} = l_I) + L_{\mathrm{SD}}^{[\mathrm{ext}]}(u_{I,t} = l_I)
\end{aligned} \tag{24}$$

where

$$\begin{aligned}
&L_{\mathrm{a}}(u_{I,t} = l_I) \\
&\qquad = \log [P(\tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}|u_{I,t} = l_I, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}})/P(\tilde{\boldsymbol{Z}}_{I,1}^{\mathrm{T}}|u_{I,t} = 0, \tilde{\boldsymbol{U}}_{I,1}^{\mathrm{T}})]
\end{aligned} \tag{25}$$

$$L_{\mathrm{c}}(u_{I,t} = l_I) = \log [P(\tilde{u}_{I,t}|u_{I,t} = l_I)/P(\tilde{u}_{I,t}|u_{I,t} = 0)] \tag{26}$$

and

$$\begin{aligned}
&L_{\mathrm{SD}}^{[\mathrm{ext}]}(u_{I,t} = l_I) \\
&\quad = \max_{n\in R_{l_I}}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = n_J) + \log P(\tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}}|u_{J,t} = n_J, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}) \\
&\qquad + \hat{\beta}_t^u(n) + \max_k{}^*\{\log P(u_t = n|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\} \\
&\qquad - \max_{m\in R_0}{}^*\{\log P(\tilde{u}_{J,t}|u_{J,t} = m_J) + \log P(\tilde{\boldsymbol{Z}}_{J,1}^{\mathrm{T}}|u_{J,t} = m_J, \tilde{\boldsymbol{U}}_{J,1}^{\mathrm{T}}) \\
&\qquad + \hat{\beta}_t^u(m) + \max_k{}^*\{\log P(u_t = m|u_{t-1} = k) + \hat{\alpha}_{t-1}^u(k)\}\}
\end{aligned} \tag{27}$$

doi: 10.1049/iet-com.2011.0268

As shown in the Appendix, the a priori LLR in (25) is equal to the de-interleaved sequence of extrinsic information resulting from the channel decoding, that is, $L_a(u_{I,t} = l_I) = L_{CD}^{[ext]}(u_{I,t} = l_I)$. The extrinsic LLR $L_{SD}^{[ext]}(u_{I,t} = l_I)$ contains the new part of information which has been determined by the source decoder by exploiting the residual source redundancy as well as the inter-description correlation induced by the MDSQ. With respect to (24), the extrinsic LLR resulting from the source decoding can be calculated by

$$L_{SD}^{[ext]}(u_{I,t} = l_I) = L(u_{I,t} = l_I | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T}) - L_{CD}^{[ext]}(u_{I,t} = l_I) - L_c(u_{I,t} = l_I) \quad (28)$$

which is used after interleaving as a priori information in the next channel decoding round. Finally, we summarise the proposed MD-ISCD scheme as follows:

1. Initialisation: Set the extrinsic information of source decoding to $L_{SD}^{[ext]}(x_{I,t}) = L_{SD}^{[ext]}(x_{J,t}) = 0$. Set the iteration counter to $n = 0$ and define an exit condition n_{max}.
2. Read series of received sequences $\tilde{\boldsymbol{Y}}_{D,1}^{T}$ and map all received systematic symbols $\tilde{x}_{D,t}$ to channel-related LLR $L_c(x_{D,t})$.
3. Perform log-MAP channel decoding on each description to compute the extrinsic LLR $L_{CD}^{[ext]}(x_{D,t})$ using (15).
4. Perform MD-SISO source decoding by inserting the de-interleaved extrinsic LLR $L_{CD}^{[ext]}(u_{I,t})$ and $L_{CD}^{[ext]}(u_{J,t})$ into (21) to compute the index a posteriori LLR $L(u_t | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T})$ and into (24) to compute the symbol a posteriori LLR $L(u_{I,t} | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T})$. Then, the extrinsic LLR $L_{SD}^{[ext]}(u_{I,t})$ is computed by (28) and is forwarded to the channel decoder as a priori information. Joint decoding of the two received sequences to extract the extrinsic LLR $L_{SD}^{[ext]}(u_{J,t})$ of description J operates in a similar manner.
5. Increase the iteration counter $n \leftarrow n + 1$. If the exit condition $n = n_{max}$ is fulfilled, then continue with step 6, otherwise proceed with step 3.
6. Compute the APP for each decoded index $u_t = l$ as follows

$$P(u_t = l | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T}) = e^{L(u_t = l | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T})} / \sum_{j=0}^{2^M - 1} e^{L(u_t = j | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \tilde{\boldsymbol{Y}}_{J,1}^{T})} \quad (29)$$

7. Estimate the decoder output signals $\hat{v}_t$ by (1) using the index APPs obtained from step 6.

5 Experimental results

Computer simulations were conducted to compare the performance of various MD-ISCD schemes for transmission of convolutionally encoded MDs over AWGN channels. First a bit-level iterative decoding scheme MD-ISCD1 [9] is considered for error mitigation using the classical BCJR algorithm for soft-output channel decoding and assisted with the bit reliability information provided by the soft-bit source decoding [15]. For the MD-ISCD1 scheme with bit interleaving, the source decoder applies two separate MAP detectors and performs turbo cross decoding to exploit the inter-description correlation [9]. Two approaches to symbol-level iterative decoding, denoted by MD-ISCD2 and MD-ISCD3, are presented and investigated. They both applied a symbol interleaver and performed log-MAP symbol decoding of binary convolutional codes based on sectionalised code trellises. Unlike the MD-ISCD1 and MD-ISCD2 which use two MAP detectors with each detector decoding one description, the MD-ISCD3 applies a joint MAP source decoder to improve the estimation of transmitted quantiser indexes by combining reliability information received on different channels. Specifically, the APP to be computed for the MD-ISCD3 is $P(u_t | \tilde{\boldsymbol{Y}}_{I,1}^{T}, \boldsymbol{Y}_{J,1}^{T})$ in (15), and $\{P(u_{I,t} | \tilde{\boldsymbol{Y}}_{I,1}^{T}), P(u_{J,t} | \tilde{\boldsymbol{Y}}_{J,1}^{T})\}$ for the other two schemes. The input signals considered here include are first order Gauss–Markov sources described by $v_t = \rho v_{t-1} + w_t$, where w_t is a zero-mean, unit-variance white Gaussian noise, with correlation coefficients of $\rho = 0.8$ and 0.95. As indicated in [16], a value of $\rho = 0.95$ can be found for scale factors determined in the MPEG audio codec for digital audio broadcasting. On the other hand, $\rho = 0.8$ provides a good fit to the long-time-averaged autocorrelation function of 8 kHz-sampled speech that is bandpass-filtered to the range (300 and 3400 Hz) [17]. A total of 3 000 000 input samples is processed by a scalar M-bit Lloyd–Max quantiser and each quantiser index is mapped to two descriptions, each with R bits per symbol per channel. For each of the two descriptions, the bitstreams were spread by an interleaver of length 300 bits and afterwards they were channel encoded by a rate-1/2 recursive systematic convolutional code with a memory order 2 and generator polynomial $\boldsymbol{G}(D) = (1, (1 + D^2)/(1 + D + D^2))$.

A preliminary experiment was first performed to examine the step-wise quality gains because of the turbo-like evaluation of channel-code and source-code redundancies. For quality evaluation we consider the parameter signal-to-noise ratio (SNR), calculated according to $\sum_t v_t^2 / \sum_t (v_t - \tilde{v}_t)^2$. Here, v_t represents the input sample and $\tilde{v}_t$ represents the systems's output sample. The variation of parameter SNR as a function of the channel SNR E_s/N_0 for MD-ISCD3 simulation of Gauss–Markov sources with $\rho = 0.95$ and $(M, R) = (5, 3)$ is shown in Fig. 3. The results indicate that a turbo-like refinement of the extrinsic information from both constituent decoders makes substantial quality improvements possible. The full gain in parameter SNR is reached after three iterations. The

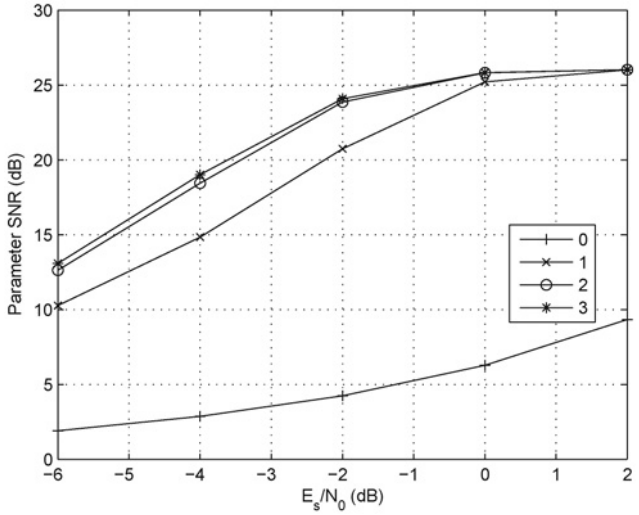

Fig. 3 *MD-ISCD3 performance for Gauss–Markov sources with $\rho = 0.95$ and $(M, R) = (5, 3)$*

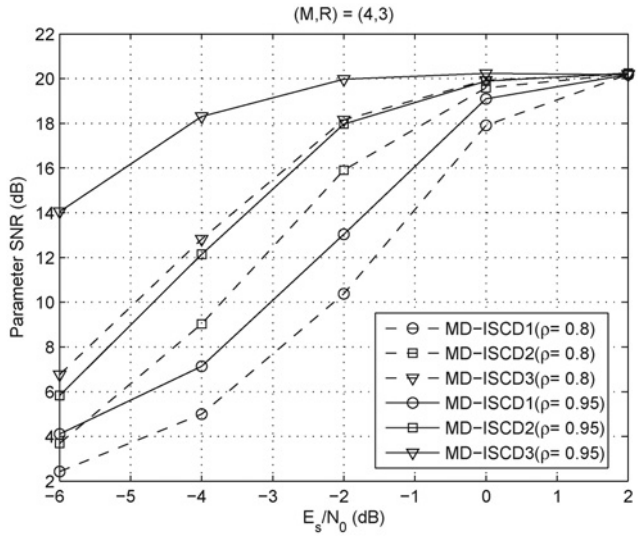

Fig. 4 *SNR performance of different decoders for (M, R) = (4, 3) and Gauss–Markov sources (ρ = 0.8, 0.95)*

investigation further showed that the improved performance achievable using MD-ISCD3 is more noticeable for lower channel SNR. To elaborate further, SNR performances of various MD-ISCD schemes were examined for Gauss–Markov sources with $\rho = 0.8$ and 0.95. We provide results for experiments on MDSQ with $(M, R) = (4, 3)$ and (5, 3) in Figs. 4 and 5, respectively. Three iterations of the algorithm were performed by each decoder as further iterations did not result in a significant improvement. The results clearly demonstrate the improved performance achievable using symbol decoders MD-ISCD2 and MD-ISCD3 in comparison to that of bit-based MD-ISCD1. Furthermore, the improvement has a tendency to increase for lower channel SNR and for more heavily correlated Gaussian sources. This indicates that the extrinsic information between source and channel decoders is better to be exploited at the symbol level. The investigation further showed that there is a considerable gap between the MD-ISCD2 and MD-ISCD3 schemes. Moreover, the performance gain achievable using MD-ISCD3 increases as more and more diagonals are included in the index

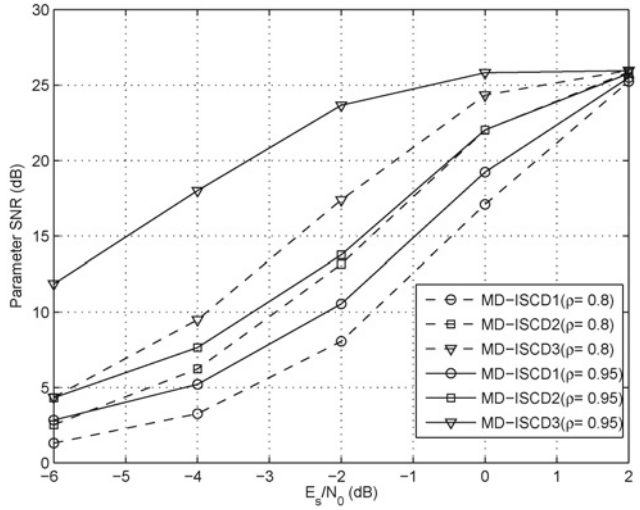

Fig. 5 *SNR performance of different decoders for (M, R) = (5, 3) and Gauss–Markov sources (ρ = 0.8, 0.95)*

assignment. For the case of $(M, R) = (4, 3)$ and $\rho = 0.95$, the MD-ISCD3 yields about 1.97 dB improvement at $E_b/N_0 = -2$ dB relative to the MD-ISCD2. For the case of $(M, R) = (5, 3)$, the parameter SNR can further be improved by up to 9.91 dB. The difference between them is due to the fact that MD-ISCD2 only accounts for the information received on a single description through the knowledge of symbol APP $P(u_{D,t}|\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})$. On the other hand, the MD-ISCD3 uses in its APP computation the total channel outputs $\{\tilde{\boldsymbol{Y}}_{I,1}^{\mathrm{T}}, \tilde{\boldsymbol{Y}}_{J,1}^{\mathrm{T}}\}$ and makes the final decision by incorporating the inter-description correlation as a result of the MDSQ.

6 Conclusions

This study presents a new MD-ISCD technique which allows to exploit the source residual redundancy as well as the inter-description correlation to the fullest extent. First a log-MAP symbol decoding scheme is proposed to decode binary convolutional codes and is shown to be superior to the bit-level BCJR algorithm. Performance of the MD-ISCD is further enhanced by exchanging between its constituent decoders the whole symbol extrinsic information. Also proposed is a joint MAP source decoder which processes the total channel outputs and combines them with inter-description correlation to improve the estimation of transmitted quantiser indexes. Experimental results indicate that the combined use of a symbol-based channel decoder and a joint MAP source decoder allows the proposed MD-ISCD scheme to achieve high robustness against channel noises.

7 Acknowledgment

This study was supported by the National Science Council, Republic of China, under contract NSC 98-2221-E-009-090-MY3.

8 References

1 Goyal, V.K.: 'Multiple description coding: compression meets the network', *IEEE Signal Process. Mag.*, 2001, **18**, (5), pp. 74–93
2 Vaishampayan, V.A.: 'Design of multiple description scalar quantizers', *IEEE Trans. Inf. Theory*, 1993, **39**, pp. 821–834
3 Dong, B., Wang, X., Doucet, A.: 'A new class of soft MIMO demodulation algorithms', *IEEE Trans. Commun.*, 2003, **51**, (11), pp. 2752–2763
4 Doucet, A., Wang, X.: 'Monte Carlo methods for signal processing: a review in the statistical signal processing context', *IEEE Signal Process. Mag.*, 2005, **22**, (6), pp. 152–170
5 Cappe, O., Godsill, S.J., Moulines, E.: 'An overview of existing methods and recent advances in sequential Monte Carlo', *Proc. IEEE*, 2007, **95**, (5), pp. 899–924
6 Gortz, N.: 'A generalized framework for iterative source-channel decoding'. Annals of Telecommunications, Special Issue on Turbo Codes, July/August 2001, pp. 435–446
7 Adrat, M., Vary, P., Spittka, J.: 'Iterative source-channel decoder using extrinsic information from softbit source decoding'. Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Salt Lake City, UT, USA, May 2001, vol. 4, pp. 2653–2656
8 Srinivasan, M.: 'Iterative decoding of multiple descriptions'. Proc. IEEE ICC, March 1999, pp. 3–12
9 Barros, J., Hagenauer, J., Gortz, N.: 'Turbo cross decoding of multiple descriptions'. Proc. IEEE ICC, 2002, vol. 3, pp. 1398–1402
10 Bahl, L.R., Cocke, J., Jelinek, F., Raviv, J.: 'Optimal decoding of linear codes for minimizing symbol error rate', *IEEE Trans. Inf. Theory*, 1974, **IT-20**, pp. 284–287
11 Lin, S., Costello, D.J.: 'Error control coding' (Prentice-Hall, New Jersey, 2004)
12 Liu, Y., Lin, S., Fossorier, M.P.C.: 'MAP algorithms for decdoing linear block codes based on sectionalised trellis diagrams', *IEEE Trans. Commun.*, 2000, **48**, pp. 577–587

doi: 10.1049/iet-com.2011.0268

13 Erfanian, J.A., Pasupathy, S., Gulak, G.: 'Reduced complexity symbol detectors with parallel structures for ISI channels', *IEEE Trans. Commun.*, 1994, **42**, pp. 1661–1671
14 Robertson, P., Villebrun, E., Hoeher, P.: 'A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain'. Proc. IEEE Int. Conf. on Communication, June 1995, vol. 2, pp. 1009–1013
15 Fingscheidt, T., Vary, P.: 'Softbit speech decoding: a new approach to error concealment', *IEEE Trans. Speech Audio Process.*, 2011, **9**, (3), pp. 240–251
16 Gortz, N.: 'On the iterative approximation of optimal joint source-channel decoding', *IEEE J. Select. Areas Commun.*, 2001, **19**, (9), pp. 1662–1670
17 Jayant, N.S., Noll, P.: 'Digital coding of waveforms' (Prentice-Hall, Englewood Cliffs, NJ, 1984)

9 Appendix

In this Appendix we shall show that the a priori LLR in (24) is equal to the de-interleaved sequence of extrinsic information provided by the SISO channel decoder, that is, $L_{\mathrm{a}}(u_{I,t}=l_I)=L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{I,t}=l_I)$. The APP of a systematic symbol $x_{D,t}=l_D$, given the received code sequences $\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}}=(\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}},\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}})$, can be decomposed by using the Bayes theorem as

$$\begin{aligned}
&P(x_{D,t}=l_D|\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})\\
&\quad=P(x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})\\
&\qquad\cdot P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})/P(\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})\\
&\quad=P(\tilde{x}_{D,t}|x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{t-1},\tilde{\boldsymbol{X}}_{D,t+1}^{\mathrm{T}})\\
&\qquad\cdot P(x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{t-1},\tilde{\boldsymbol{X}}_{D,t+1}^{\mathrm{T}})\\
&\qquad\times P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})/P(\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})\\
&\quad=C\cdot P(\tilde{x}_{D,t}|x_{D,t}=l_D)\cdot P(x_{D,t}=l_D)\\
&\qquad\cdot P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})
\end{aligned}\tag{30}$$

where $C=P(\tilde{\boldsymbol{X}}_{D,1}^{t-1},\tilde{\boldsymbol{X}}_{D,t+1}^{\mathrm{T}})/P(\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})$. We rewrite (29) in log-likelihood algebra as

$$\begin{aligned}
&L(x_{D,t}=l_D|\tilde{\boldsymbol{Y}}_{D,1}^{\mathrm{T}})\\
&\quad=L_a(x_{D,t}=l_D)+L_c(x_{D,t}=l_D)+L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t}=l_D)
\end{aligned}\tag{31}$$

with

$$L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t}=l_D)=\log\frac{P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|x_{D,t}=l_D,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})}{P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|x_{D,t}=0,\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})}\tag{32}$$

Since the de-interleaved sequence of $L_{\mathrm{CD}}^{[\mathrm{ext}]}(x_{D,t}=l_D)$ is used by the source decoder, we have

$$L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t}=l_D)=\log\frac{P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|u_{D,t}=l_D,\tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}})}{P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|u_{D,t}=0,\tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}})}\tag{33}$$

where $\tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}}=\Phi^{-1}(\tilde{\boldsymbol{X}}_{D,1}^{\mathrm{T}})$. Once the LLR $L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t}=l_D)$ has been determined, we can compute the probability as follows

$$P(\tilde{\boldsymbol{Z}}_{D,1}^{\mathrm{T}}|u_{D,t}=l_D,\tilde{\boldsymbol{U}}_{D,1}^{\mathrm{T}})=\mathrm{e}^{L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t}=l_D)}/\sum_{j=0}^{2^R-1}\mathrm{e}^{L_{\mathrm{CD}}^{[\mathrm{ext}]}(u_{D,t}=j)}\tag{34}$$