Reduced-Search Source Decoders for Iterative Source-Channel Decoding

Laurent Schmalen[†], *Peter Vary*[†], *Marc Adrat*[⋆]

[†]Institut für Nachrichtengeräte und Datenverarbeitung (IND), RWTH Aachen, 52056 Aachen
E-Mail: {schmalen,vary}@ind.rwth-aachen.de
[⋆]*now with the* Research Establishment for Applied Sciences (FGAN FKIE/KOM), Wachtberg, Germany
E-Mail: adrat@fgan.de

Abstract

Iterative source-channel decoding (ISCD) exploits the residual redundancy of source codec parameters by using the Turbo principle. However, ISCD might require more computational complexity than available as the utilized soft decision source decoder (SDSD) can be computationally quite expensive. In this paper we propose a reduced-search SDSD, based on the M-algorithm known from channel decoding, which considerably reduces the complexity of the receiver. Furthermore, we show that by slightly modifying the quantization at the transmitter, the complexity can be further reduced without noticeable performance losses. Complexity figures are given for all approaches as well as a simulation example showing the performance of the complexity-reduced SDSD in an ISCD framework.

1 Introduction

With the discovery of Turbo codes, channel coding close to the Shannon limit has become possible with moderate computational complexity. In the past years, the Turbo principle of exchanging extrinsic information between separate channel decoders has also been extended to other receiver components. In a Turbo-like process the residual redundancy of source codec parameters such as scale factors or predictor coefficients for speech, audio, and video signals can be exploited by *iterative source-channel decoding* (ISCD) [1, 2]. This residual redundancy occurs due to imperfect source encoding resulting for instance from delay and complexity constraints. It can be utilized by a *soft decision source decoder* (SDSD) [3] which exchanges extrinsic reliabilities with a channel decoder.

The execution of the SDSD, however, can be computationally quite demanding, especially if large quantizer codebooks are employed. In non-iterative transmission systems, it is possible to execute the SDSD only for the most significant bits, as proposed in [4]. However, if such a source decoder is utilized in an ISCD transmission scheme, the source decoder can only generate extrinsic information for the most significant bits, leading to a sub-optimal system performance.

In order to reduce the complexity, we have proposed in [5] a transmitter modification called *conditional quantization* which allows to considerably reduce the number of operations carried out at the soft decision source decoder. However, the proposed modification also affects the quality of the reconstructed signal. Therefore, in this paper we propose a receiver-only approach called M-SDSD. This approach is similar to the well-known M-algorithm [6], [7], known from channel decoding. A similar approach has also been introduced in [8]. We show that the number of operations can be considerably reduced by only slightly affecting the overall system performance. Furthermore, by combining the proposed source decoders with the *conditional quantization* approach presented in [5], we show that the complexity can even further be reduced.

Parts of this work were financed by the European Union under Grant FP6-2002-IST-C 020023-2 FlexCode

2 System Model

In the following, we will briefly review the *iterative source-channel decoding* (ISCD) system. In Fig. 1 the baseband model of ISCD is depicted. At time instant t a source encoder generates a frame $\underline{u}_t = (u_1, \dots u_{K_S})$ of K_S unquantized source codec parameters u_κ, with $\kappa \in \{1, \dots, K_S\}$ denoting the position in the frame. Each value u_κ is individually mapped to a quantizer reproduction level $\bar{u}_\kappa$, with $\bar{u}_\kappa \in \mathbb{U} = \{\bar{u}^{(1)}, \dots, \bar{u}^{(Q)}\}$. The set $\mathbb{U}$ denotes the quantizer codebook with a total number of $|\mathbb{U}| = Q$ codebook entries. A unique bit pattern $\mathbf{x}_\kappa \in \mathbb{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(Q)}\}$ of w^* bits (i.e., $\mathbb{X} \subset \{0;1\}^{w^*}$), with $w^* \geq \lceil \log_2 Q \rceil \doteq w$, is assigned to each quantizer level $\bar{u}_\kappa$ according to the index assignment $\Gamma(\bar{u}^{(i)}) = \mathbf{x}^{(i)}$. The single bits of a bit pattern $\mathbf{x}_\kappa$ are indicated by $x_\kappa(m)$, $m \in \{1, \dots, w^*\}$. If $M^* > \log_2 Q$, the index assignment Γ is called *redundant index assignment* [9] and can be considered to be the composite function $\Gamma(\bar{u}) = \Gamma_\mathrm{R}(\Gamma_\mathrm{NB}(\bar{u}))$. The function Γ_NB performs a *natural binary* index assignment, i.e., the binary representation of the codebook index of $\bar{u}$ is assigned to $\Gamma_\mathrm{NB}(\bar{u})$. The function Γ_R can be regarded as being a (potentially non-linear) block code of rate $r^{\mathsf{IA}} = w/w^*$. After the index assignment, K_S bit patterns are grouped to a frame of bit patterns $\underline{\mathbf{x}} = (\mathbf{x}_1, \dots, \mathbf{x}_{K_S})$ consisting of $K_S \cdot w^*$ bits. The frame $\underline{\mathbf{x}}$ of bits is then rearranged by a bit interleaver π in a deterministic, pseudo-random like manner. The interleaved frame with $K_S \cdot w^*$ bits is denoted as $\underline{\check{\mathbf{x}}}$.

For channel encoding of a frame $\underline{\check{\mathbf{x}}}$, we use a convolutional code of constraint length $J+1$ and of rate r^{C}. In general, any channel code can be used as long as the respective decoder is able to provide the required *extrinsic reliabilities*. In this paper, we restrict ourselves to rate $r^{\mathsf{C}} = 1$, recursive, non-systematic convolutional codes as it has been shown [10] that the inner code of a serially concatenated system should be recursive in order to be

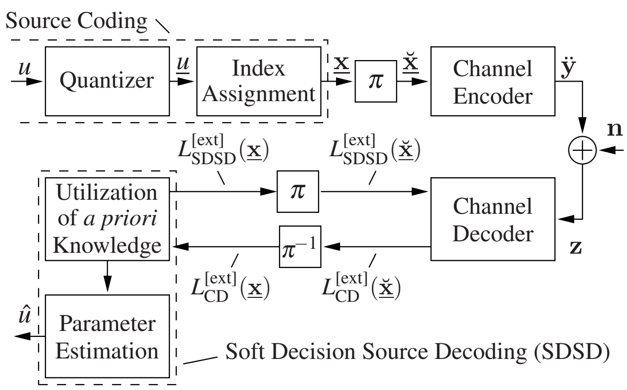

Figure 1: Baseband model of the utilized ISCD system

capacity achieving. For the termination of the code, J tail bits are appended to $\underline{\check{\mathbf{x}}}$. The encoded frame of length $K_S \cdot w^* + J$ is denoted by $\underline{\mathbf{y}}$. The bits y_k of $\underline{\mathbf{y}}$ are indexed by $k \in \{1, \dots, K_S \cdot w^* + J\}$. Prior to transmission over the channel, the encoded bits y_k are mapped to bipolar bits $\ddot{y}_k$ forming a sequence $\ddot{\mathbf{y}} \in \{\pm 1\}^{K_S \cdot w^* + J}$.

On the channel, the modulation symbols $\ddot{y}_k$ (with symbol energy $E_s = 1$) are subject to additive white Gaussian noise (AWGN) with known variance $\sigma_n^2 = N_0/2$. The received symbols z_k are transformed to L-values [11] prior to being evaluated in a Turbo process which exchanges *extrinsic* reliabilities between channel decoder (CD) and soft decision source decoder (SDSD). The channel decoder used in this paper is based on the LogMAP algorithm [12], [11]. For the derivations of the equations for computing the *extrinsic* probabilities of the SDSD, we refer the reader to the literature, e.g., [2], [3], [13]. In Section 3, we will briefly revise the SDSD equations and give expressions in the logarithmic domain in order to evaluate the complexity and the complexity savings of the proposed algorithms.

3 Soft Decision Source Decoding

The SDSD may be interpreted as a modification of the well-known BCJR algorithm [12], operating on a fully developed trellis diagram. For the derivation of the SDSD equations, we refer to the literature, e.g., [1], [2], [13]. The SDSD utilizes the remaining residual redundancy after source coding in the signal. In this paper, we assume that this residual redundancy is modelled as a Markov process of first order. The residual redundancy which we consider in this paper is so-called *inter-frame redundancy*, i.e., the parameters in one block are statistically independent but are correlated between ("inter-") different blocks, i.e., $P(\bar{u}_{\kappa,t}|\bar{u}_{\kappa-1,t}) = P(\bar{u}_{\kappa,t})$ but $P(\bar{u}_{\kappa,t}|\bar{u}_{\kappa,t-1}) \neq P(\bar{u}_{\kappa,t})$.

In the following, we briefly revise the equations for SDSD as they are needed to estimate the complexity of the algorithm. We give the equations in the logarithmic domain, as an implementation in the logarithmic domains offers several advantages, such as, e.g., better numerical stability [5].

The input to the soft decision source decoder (SDSD) are the (deinterleaved) extrinsic L-values generated by the channel decoder

$$L_{\text{SDSD}}^{[\text{input}]}(x_{\kappa,t}(m)) = L_{\text{CD}}^{[\text{ext}]}(x_{\kappa,t}(m)) = \ln\left(\frac{P_{\text{CD}}^{[\text{ext}]}(x_{\kappa,t}(m)=0)}{P_{\text{CD}}^{[\text{ext}]}(x_{\kappa,t}(m)=1)}\right). \quad (1)$$

The first step of the SDSD consists in determining the logarithmic input reliabilities $\tilde{\theta}(\mathbf{x}_{\kappa,t}^{(j)})$ for each distinct bit pattern $\mathbf{x}_{\kappa,t}^{(j)}$ $(j = 1, \dots, Q)$ for parameter position κ at time instant t with

$$\tilde{\theta}(\mathbf{x}_{\kappa,t}^{(j)}) = \sum_{m=1}^{w^*} \frac{\ddot{x}_{\kappa,t}^{(j)}(m)}{2} L_{\text{SDSD}}^{[\text{input}]}(x_{\kappa,t}(m)) \quad (2)$$

and $\ddot{x}_{\kappa,t}^{(j)}(m)$ denoting the bipolar representation of $x_{\kappa,t}^{(j)}(m)$, with $\ddot{x}_{\kappa,t}^{(j)}(m) = 1 - 2 \cdot x_{\kappa,t}^{(j)}(m)$.

After determination of the reliabilities $\tilde{\theta}(\mathbf{x}_{\kappa,t}^{(j)})$, the source decoder can compute the forward recursion

$$\tilde{\alpha}(\mathbf{x}_{\kappa,t}^{(j)}) = \tilde{\theta}(\mathbf{x}_{\kappa,t}^{(j)}) + \max_{i=1\dots Q}{}^{*}\left(\tilde{\alpha}(\mathbf{x}_{\kappa,t-1}^{(i)}) + \tilde{P}(\mathbf{x}_{\kappa,t}^{(j)}|\mathbf{x}_{\kappa,t-1}^{(i)})\right) \quad (3)$$

with the initialization $\tilde{\alpha}(\mathbf{x}_{\kappa,0}^{(\ell)}) = \tilde{P}(\mathbf{x}_{\kappa}^{(\ell)})$ and $\max^*(\delta_1, \delta_2) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_1 - \delta_2|})$. The factors $\tilde{P}(\mathbf{x}_{\kappa}^{(\ell)}) \doteq \ln P(\mathbf{x}_{\kappa}^{(\ell)})$ and $\tilde{P}(\mathbf{x}_{\kappa,t}^{(j)}|\mathbf{x}_{\kappa,t-1}^{(i)}) \doteq \ln P(\mathbf{x}_{\kappa,t}^{(j)}|\mathbf{x}_{\kappa,t-1}^{(i)})$ denote the logarithmic *a priori* probabilities. They can be computed offline and stored in the decoder memory. As only inter-frame redundancy is exploited, the decoder can only execute a single forward recursion if no further delay is allowed. If several blocks are stored at the receiver, an additional backward recursion can be carried out, leading to a slightly better decoding result. However, we do not consider this case here. For the equations of the backward recursion, see, e.g., [5].

Using the result of the forward recursion, the extrinsic information finally is calculated by

$$L_{\text{SDSD}}^{[\text{ext}]}(x_{\kappa,t}(m)) = \max_{\substack{j=1\dots Q \\ x_{\kappa,t}^{(j)}(m)=0}}{}^{*}\left(\tilde{\alpha}(\mathbf{x}_{\kappa,t}^{(j)}) - \frac{1}{2}L_{\text{SDSD}}^{[\text{input}]}(x_{\kappa,t}(m))\right) - \max_{\substack{j=1\dots Q \\ x_{\kappa,t}^{(j)}(m)=1}}{}^{*}\left(\tilde{\alpha}(\mathbf{x}_{\kappa,t}^{(j)}) + \frac{1}{2}L_{\text{SDSD}}^{[\text{input}]}(x_{\kappa,t}(m))\right). \quad (4)$$

3.1 Complexity of SDSD

The evaluation of (2) requires $Q \cdot w^*$ additions per parameter as the $\tilde{\theta}(\mathbf{x}_{\kappa,t}^{(j)})$ have to be determined for each possible bit pattern $\mathbf{x}_{\kappa,t}^{(j)} \in \mathbb{X}$. The factors $\frac{1}{2}L_{\text{SDSD}}^{[\text{input}]}(x_{\kappa,t}(m))$ can be calculated and stored (as they are needed a second time in the run-time of the algorithm) using w^* multiplications by a constant per parameter. As the multiplication by $\frac{1}{2}$ can be efficiently realized in hardware, we do not take into consideration this multiplication in the complexity evaluation. The multiplication by $\ddot{x}_{\kappa,t}^{(j)}(m)$ corresponds to a sign change only as $\ddot{x}_{\kappa,t}^{(j)}(m) \in \{\pm 1\}$. The evaluation of (3) requires Q^2 $\max^*$ operations as well as $Q + Q^2$ additions per parameter. Finally, the evaluation of (4) requires $w^* Q$ $\max^*$ operations as well as $w^*(Q+1)$ additions. Therefore, a total number of $Q^2 + Q(2w^* + 1) + w^*$ additions as well as $Q^2 + w^* Q$ $\max^*$ operations are required for carrying out the full-complexity SDSD.

4 Reduced Search Soft Decision Source Decoding

4.1 M-SDSD

In channel decoding of convolutional codes, the M-algorithm [7] can be successfully applied in order to reduce the complexity of the decoder. Another successful field of application is channel equalization of ISI-channels [6].

The SDSD in fact is a variant of the BCJR algorithm [12] operating on a fully developed trellis [13]. Each state corresponds to a quantizer reproduction level (or a bit pattern, respectively). The state transitions correspond to the possible transitions $\mathbf{x}_{\kappa,t-1}^{(i)} \to \mathbf{x}_{\kappa,t}^{(j)}$. At each trellis transition the M-SDSD determines the M states with the highest probability and only considers these states for computing the state transitions.

Equation (3) is performed for all states $\mathbf{x}_{\kappa,t}^{(j)}$ but only the M (saved) best states from the previous time instant $t-1$ are considered. Therefore, the complexity of (3) reduces to $Q+MQ$ (with $M<Q$) additions and MQ max* operations. After execution of the complexity-reduced version of (3), the M best states have to be determined, i.e., the $\tilde{\alpha}(\mathbf{x}_{\kappa,t}^{(j)})$ with the largest value. This can be done using a simple search with $MQ-\sum_{n=1}^{M} n = MQ-\frac{1}{2}(M^2+M)$ compare operations (states that have already been chosen don't need to be compared anymore). For determining the extrinsic information, only the M best $\tilde{\alpha}(\mathbf{x}_{\kappa,t})$ are utilized, and therefore the complexity of (4) reduces to $w^*(M+1)$ additions and w^*M max* operations.

4.2 M-SDSD with Conditional Quantization

A further complexity reduction of the source decoder can be achieved if the transmitter can be modified. In this case, the conventional scalar quantizer can be replaced by a so-called *conditional quantizer* (CQ). The conditional quantizer has been introduced in [5] and corresponds to a quantizer with memory: depending on the previously quantized sample $\bar{u}_{\kappa,t-1}^{(i)}$ a reduced codebook is utilized which considers only the entries which have a transition probability $P(\bar{u}_{\kappa,t}^{(j)}|\bar{u}_{\kappa,t-1}^{(i)}) > \mathscr{T}$, with $\mathscr{T}$ being the probability threshold of the quantizer. Depending on $\mathscr{T}$, the number of transitions in the trellis diagram is considerably reduced and thus also the number of operations. A drawback of conditional quantization is however the reduced reconstruction quality in error-free channel conditions. For details, we refer the reader to [5]. Note that if a CQ is employed at the transmitter, (3) has to be modified: the residual source redundancy $\tilde{P}(\mathbf{x}_{\kappa,t}^{(j)}|\mathbf{x}_{\kappa,t-1}^{(i)})$ has to be replaced by a modified probability $\tilde{P}_{\text{red}}(\mathbf{x}_{\kappa,t}^{(j)}|\mathbf{x}_{\kappa,t-1}^{(i)})$ as the source redundancy is modified by the conditional quantizer. For details, see [5].

The complexity of the M-SDSD is more difficult to determine as the number of transitions per state varies. Therefore, only a tight upper bound for the complexity can be given which however is needed for a hardware realization guaranteeing a certain throughput. The conditional quantizer uses a reduced codebook

$$\mathbb{U}_{\text{red},i} = \left\{ \bar{u}_{\kappa,t}^{(j)} : P\left(\bar{u}_{\kappa,t}^{(j)}|\bar{u}_{\kappa,t-1}^{(i)}\right) > \mathscr{T}, \forall \bar{u}_{\kappa,t}^{(j)} \in \mathbb{U} \right\}. \quad (5)$$

depending on the previously quantized sample. The number of entries of $|\mathbb{U}_{\text{red},i}|$ varies for different values of i. Let S_M denote the sum of the number of transitions of those M states (i.e., quantizer reproduction levels) having the largest number of entries $|\mathbb{U}_{\text{red},i}|$. This is a worst case for the number of transitions in the SDSD: the M states have been selected which lead to the highest number of transitions that have to be calculated by the SDSD.

Figure 2 shows an example of the reduction of states and state transitions in the SDSD trellis diagram. Figure 2-a) shows the fully developed trellis for $Q=16$ quantization levels. In Fig. 2-b) the number of of states is reduced by the M-SDSD with $M=6$. However, the number of state transitions ($Q=16$) per state is unchanged. This number can be reduced by applying conditional quantization leading to the trellis in Fig. 2-c). Note that Fig. 2-b) and Fig. 2-c) only show snapshots. The M selected states may vary for each trellis transition.

In the worst case, the complexity of (3) now reduces to $Q+S_M$ additions and S_M max* operations. The complexity for determining the M best states remains the same as in the case of the M-SDSD. Finally, the complexity of evaluating (4) by the conditional quantizer is not affected,

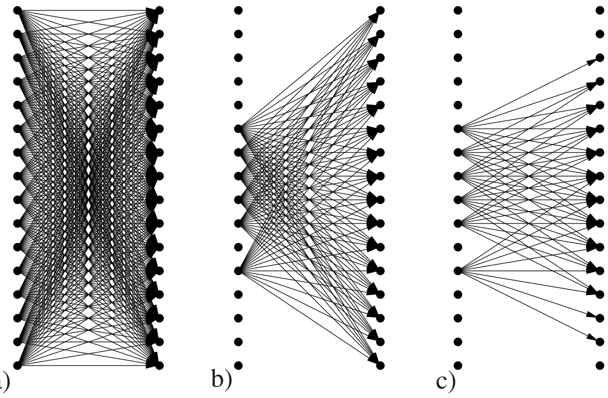

Figure 2: Trellis diagrams exploited at the source decoder in (3) for $Q=16$.
a) full trellis diagram
b) exemplarily trellis exploited at one stage of the M-algorithm with $M=6$
c) if additionally conditional quantization is exploited with $\mathscr{T}=10^{-2}$

therefore $w^*(M+1)$ additions and w^*M max* operations are required.

The total number of operations for the three different algorithms (standard SDSD, M-SDSD and M-SDSD with conditional quantization (CQ-M-SDSD) are summarized in Table 1.

5 Simulation Example

The capabilities of the complexity-reduced ISCD system are demonstrated by a simulation example. The *parameter signal-to-noise ratio* (SNR) between the originally generated parameters u and the estimated parameters $\hat{u}$ is used for quality evaluation. The parameter SNR is plotted for different values of E_s/N_0. The source is realized by K_S independent Gauss-Markov (autoregressive) processes with correlation coefficient ρ fixed to $\rho=0.9$. This autocorrelation value can be observed in typical speech and audio codecs, e.g., for the scale factors in CELP codecs or MP3. The quantization is performed using a $Q=16$ level Lloyd-Max codebook $\mathbb{U}$. The utilized block coded index assignment Γ_{R} is a repetition code [14] ($w^*=8$). The utilized channel code is a rate $r^{\mathsf{C}}=1$ recursive non-systematic convolutional code of constraint length $J=4$ with generator polynomial $G^{\mathsf{C}}(D) = \left(\frac{1}{1+D+D^2+D^3}\right)$. The non-iterative reference scheme as well as the hard-output channel decoding reference uses optimized components for non-iterative systems, i.e., a natural binary index assignment with $w^*=\log_2\lceil Q\rceil = 4$ and a rate $r^{\mathsf{C}}=\frac{1}{2}$ recursive, systematic convolutional code of constraint length $J=4$ with $G^{\mathsf{C}}(D) = \left(1, \frac{1+D^2+D^3}{1+D+D^3}\right)$.

We assume that the source exhibits inter-frame correlation, i.e., all the single elements u_κ of $\underline{u}_t$ are statistically independent from each other. The different samples u_κ are correlated with their counterpart from previous frames. A frame consists of $K_S=250$ parameters. In order not to introduce any additional delay, the forward-only SDSD as introduced in Sec. 3 is employed.

The simulation results are depicted in Fig. 3. At the receiver 15 iterations have been carried out in a first experiment. It can be seen that the utilization of the M-SDSD with $M=6$ does not cause any noteworthy performance losses for good channel conditions ($E_s/N_0 > -4$ dB). The application of conditional quantization further reduces the complexity at the expense of a slightly decreased param-

Table 1: Operations per parameters needed by the different complexity-reduced SDSD variations

	ADD	max*	CMP
Standard	$Q^2+(2w^*+1)Q+w^*$	Q^2+w^*Q	
M-SDSD	$(w^*+M+1)Q+w^*(M+1)$	$M(Q+w^*)$	$MQ-\frac{1}{2}(M^2+M)$
CQ-*M*-SDSD (upper bound)	$(w^*+1)Q+S_M+w^*(M+1)$	S_M+w^*M	$MQ-\frac{1}{2}(M^2+M)$

eter SNR in good channel conditions [5]. The fact that the CQ has a better performance in the waterfall region is explained in [5]. With $\mathscr{T}=10^{-2}$ and $M=6$, the factor S_M can be determined to $S_M=50$, by considering the 6 codebooks $\mathbb{U}_{\text{red},i}$ with the highest number of entries. The complexity per parameter of the three different utilized SDSD algorithms with the configuration of the simulation example are summarized in the following table:

	ADD	max*	CMP
Standard	536	384	
M-SDSD	296	144	75
CQ-*M*-SDSD (upper bound)	218	98	75

With $w^*=8$ and $r^{\mathsf{C}}=1$, the LogMAP decoder [11], [15] requires approximately 656 additions and 256 max* operations per parameter. If we assume in a first approximation that all three operations (additions, max*, and compares) require the same amount of computing power in a hardware realization, the total number of operations amounts to 19545 per parameter if 15 iterations are utilized and the *M*-SDSD with conditional quantization ($\mathscr{T}=10^{-2}$) is utilized. If the full SDSD is employed, only 10 iterations can be carried out if the above number of operations (19545) per parameter shall be fixed as an upper bound. The simulation result for 10 iterations is also given in Fig. 3. It can be seen that it is advantageous to utilize a complexity-reduced source decoder and a higher number of iterations if the total number of operations to be performed is limited.

6 Conclusion

In this paper, we have applied the *M*-algorithm known from channel decoding and channel equalization to the soft decision source decoder (SDSD) utilized in an iterative source-channel decoding. We have shown that the complexity can be considerably reduced by selecting the best states and only considering those in the computations. Furthermore, if the transmitter can be modified, a further complexity reduction can be achieved by employing conditional quantization. We have shown that the reduced-search source decoders show no noteworthy performance losses in the most interesting range of channel conditions by using less than half the number of operations of the standard SDSD. Furthermore, in a simulation example it has been shown that, if the number of available operations is fixed, it can be advantageous to employ reduced-search decoders and a higher number of iterations than the standard SDSD with a lower number of iterations.

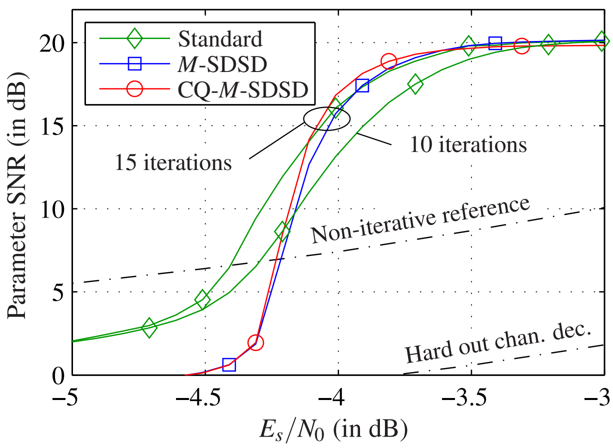

Figure 3: Parameter SNR performance for different reduced-search source decoders ($M=6$, $\mathscr{T}=10^{-2}$)

References

[1] M. Adrat, P. Vary, and J. Spittka, "Iterative Source-Channel Decoder Using Extrinsic Information from Softbit-Source Decoding," in *IEEE ICASSP*, Salt Lake City, USA, 2001.

[2] N. Görtz, "On the Iterative Approximation of Optimal Joint Source-Channel Decoding," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1662–1670, Sept. 2001.

[3] T. Fingscheidt and P. Vary, "Softbit Speech Decoding: A New Approach to Error Concealment," *IEEE Trans. Speech Audio Processing*, vol. 9, no. 3, pp. 240–251, Mar. 2001.

[4] X. Liu and S. N. Koh, "Simplification of Soft-Bit Speech Decoding and Application to MELP Encoded Speech," *Electronics Letters*, vol. 39, no. 3, pp. 332–333, Feb. 2003.

[5] L. Schmalen, P. Vary, M. Adrat, and T. Clevorn, "Complexity-Reduced Iterative Source-Channel Decoding by Conditional Quantization," in *Proc. of International Symposium on Turbo Codes & Related Topics*, Lausanne, CH, Sept. 2008.

[6] K. K. Y. Wong and P. J. McLane, "Bi-Directional Soft-Output M-Algorithm for Iterative Decoding," in *Proc. of IEEE ICC*, June 2004.

[7] V. Franz and J. B. Anderson, "Concatenated Decoding with a Reduced-Search BCJR Algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[8] M. Adrat, "Iterative Source-Channel Decoding for Digital Mobile Communications," Ph.D. dissertation, Aachener Beiträge zu Digitalen Nachrichtensystemen (ed. P. Vary), RWTH Aachen, 2003.

[9] T. Clevorn, M. Adrat, and P. Vary, "Turbo DeCodulation using Highly Redundant Index Assignments and Multi-Dimensional Mappings," in *Intl. Symposium on Turbo Codes & Related Topics*, Munich, Apr. 2006.

[10] J. Kliewer, A. Huebner, and D. J. Costello, Jr., "On the Achievable Extrinsic Information of Inner Decoders in Serial Concatenation," in *Proc. of ISIT*, Seattle, 2006.

[11] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symol Error Rate," *IEEE Trans. Inform. Theory*, vol. 20, Mar. 1974.

[13] M. Adrat, T. Clevorn, and L. Schmalen, "Iterative Source-Channel Decoding & Turbo-DeCodulation," in *Advances in Digital Speech Transmission*, R. Martin, U. Heute, and C. Antweiler, Eds. John Wiley & Sons, Ltd., Jan. 2008, ch. 13, pp. 365–398.

[14] T. Clevorn, L. Schmalen, P. Vary, and M. Adrat, "On Redundant Index Assignments for Iterative Source-Channel Decoding," *IEEE Commun. Lett.*, vol. 12, no. 7, July 2008.

[15] P. Robertson, E. Villebrun, and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Seattle, USA, 1995, pp. 1009–1013.