Graph-Based Turbo DeCodulation with LDPC Codes

Birgit Schotsch, Laurent Schmalen, and Peter Vary
Institute of Communication Systems and Data Processing
RWTH Aachen University, Aachen,Germany
{schotsch,schmalen,vary}@ind.rwth-aachen.de

Thorsten Clevorn
COM Development Center NRW
Infineon Technologies AG, Duisburg, Germany
thorsten.clevorn@infineon.com

***Abstract*—Turbo DeCodulation is the combination of iterative demodulation and iterative source-channel decoding in a multiple Turbo process. The receiver structures of bit-interleaved coded modulation with iterative decoding (BICM-ID) and iterative source-channel decoding (ISCD) are merged to one joint Turbo system, which we further enhance in this paper by using a low-density parity check (LDPC) code for channel coding, resulting in a third iterative loop. We propose to use a special LDPC code structure with short sub-codes, which can be implemented very effectively in parallel. The quadripartite Tanner graph of the Turbo DeCodulation is presented, showing that the processing of all receiver nodes could be parallelized. Simulation results including an EXIT chart analysis demonstrate the excellent capabilities of Turbo DeCodulation with its performance gain exceeding the combined gain of BICM-ID and ISCD.**

I. Introduction

Turbo codes [1, 2] and low-density parity check (LDPC) codes [3, 4] have significantly changed the design paradigms for communication systems. The Turbo principle of exchanging extrinsic information can be extended to other receiver components, e.g., to a combination of a demodulator and a channel decoder, which constitutes the so-called bit-interleaved coded modulation with iterative decoding (BICM-ID) [5], or to a combination of a channel decoder and a soft decision source decoder (SDSD) [6, 7], which is known as iterative source-channel decoding (ISCD) [8, 9]. In [10, 11] Turbo DeCodulation is presented, which combines BICM-ID and ISCD to a receiver performing joint source decoding, channel decoding, and demodulation in two iterative loops.

In this paper, we replace the convolutional code used so far in [10, 11] by an LDPC code and thus add a third iterative loop. Since also other components take part in the iterative processing a special LDPC coding scheme as proposed in [12] can be applied. The frame is segmented into short sub-frames, each encoded by a separate LDPC code through which a high flexibility in frame size can be achieved.

In Section II we describe the proposed communication system and present the emerging quadripartite Tanner graph of the new Turbo DeCodulation receiver. The graph structure and the messages are detailed. The excellent performance of the proposed Turbo DeCodulation is demonstrated by simulation in Section III. For an additional in depth analysis of the convergence behavior we employ a three-dimensional extrinsic information transfer (EXIT) chart [11, 13]. Finally, we outline the advantages of the proposed scheme regarding a possible implementation on a chip in Section IV. With its separate graph nodes and the independent short LDPC sub-codes a high degree of parallelization in processing can be achieved.

This work has been conducted within UMIC, Cluster of Excellence at RWTH Aachen University which is funded by the German Research Council (DFG).

II. The Turbo DeCodulation System

In Fig. 1 the baseband model of the proposed Turbo DeCodulation system is depicted. The inner iterative loop in the receiver corresponds to a BICM-ID system [5], while the outer iterative loop is similar to an ISCD system [8, 9]. By connecting these two loops, e.g., by a common channel code, we obtain a Turbo DeCodulation receiver which performs joint source decoding, channel decoding, and demodulation [10, 11].

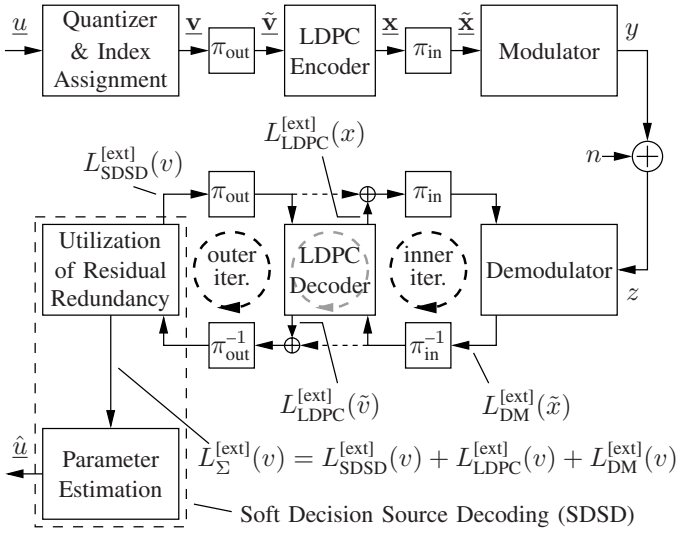

Fig. 1. Baseband model of the Turbo DeCodulation system.

A. Transmitter

At time instant τ a source encoder determines a frame $\underline{u}_\tau$ of K_S source codec parameters $u_{\kappa,\tau}$ with $\kappa = 1, \dots K_S$ denoting the position in the frame. The single elements $u_{\kappa,\tau}$ of $\underline{u}_\tau$ are assumed to be statistically independent of each other. Each value $u_{\kappa,\tau}$ is individually mapped to a quantizer reproduction level $\bar{u}_\kappa^{(\xi)}$, $\xi = 1, \dots 2^{M_\kappa}$. To each quantizer reproduction level $\bar{u}_\kappa^{(\xi)}$ selected at time instant τ a unique bit pattern $\mathbf{v}_{\kappa,\tau}$ of M_κ bits is assigned according to the index assignment Γ, $\mathbf{v}_{\kappa,\tau} = \Gamma(\bar{u}_\kappa^{(\xi)})$. For simplicity, we assume $M_\kappa = M$ for all κ. The single bits of a bit pattern $\mathbf{v}_{\kappa,\tau}$ are indicated by $v_{\kappa,\tau}^{(m)}$, $m = 1, \dots M$. The frame of bit patterns is denoted by $\underline{\mathbf{v}}_\tau$.

The first and outer bit interleaver π_{out} scrambles the incoming frame $\underline{\mathbf{v}}_\tau$ of data bits to $\underline{\tilde{\mathbf{v}}}_\tau$ in a deterministic manner. In the following, for simplicity, we consider only a single time instant τ to be jointly interleaved and channel coded. Both routines can also be realized for several consecutive frames $\underline{\mathbf{v}}$.

For channel encoding any channel code can be used as long as the respective decoder is able to provide the required extrinsic probabilities. In this paper we consider LDPC codes [3, 4], as they belong to the most powerful codes known today. Furthermore, we employ a special LDPC coding scheme as presented in [12], which segments a complete frame $\underline{\tilde{\mathbf{v}}}_\tau$ of data bits v into numerous short sub-frames, each of which is encoded by a separate LDPC code. The results in [12] show, that approximately the same performance as with a single large LDPC code can be achieved if the LDPC decoder can iteratively exchange information with other receiver components. One advantage of this scheme is, e.g., the flexibility in overall frame size as the number of sub-frames can be easily adjusted and does not require a new LDPC code or other structural changes. After encoding the sub-frames are concatenated again to a large frame $\underline{\mathbf{x}}_\tau$ of encoded bits x.

The second and inner bit-interleaver π_{in} permutes this codeword $\underline{\mathbf{x}}_\tau$ to $\underline{\tilde{\mathbf{x}}}_\tau$ The interleaved codeword $\underline{\tilde{\mathbf{x}}}_\tau$ is divided into bit patterns $\tilde{\mathbf{x}}_{k,\tau}$, $k=1,\dots K_C$, with I single bits $\tilde{x}^{(i)}_{k,\tau}$, $i=1,\dots I$. In case the last bit pattern $\tilde{\mathbf{x}}_{K_C,\tau}$ is not completely filled, the remaining positions are padded with zeros. The modulator maps each pattern $\tilde{\mathbf{x}}_{k,\tau}$ to a complex modulated symbol $y_{k,\tau}$ from the signal constellation set $\mathbb{Y}$ according to a mapping rule μ, $y_{k,\tau}=\mu(\tilde{\mathbf{x}}_{k,\tau})$. The modulated symbols are normalized to an average energy of $\mathrm{E}\{\|y_{k,\tau}\|^2\}=1$.

On the channel complex additive white Gaussian noise (AWGN) $n_{k,\tau}=n'_{k,\tau}+jn''_{k,\tau}$ with a known power spectral density of $\sigma_n^2=N_0$ is applied, i.e., $z_{k,\tau}=y_{k,\tau}+n_{k,\tau}$.

B. Receiver

The received symbols $z_{k,\tau}$ are evaluated in a multiple Turbo process, in which extrinsic reliabilities are exchanged between demodulator and LDPC channel decoder in the inner iterations, and between LDPC channel decoder and soft decision source decoder in the outer iterations. Furthermore, the LDPC decoder internally executes LDPC iterations. The reliability information can either be evaluated in terms of probabilities $P(\cdot)$ or L-values [2]. We will use both variants in parallel and at each point choose the one more suited for the explanations.

1) Demodulator (DM): The demodulator computes the extrinsic probabilities $P^{[\text{ext}]}_{\text{DM}}(\tilde{x})$ of each bit $\tilde{x}^{(i)}_{k,\tau}=b$, $b\in\{0,1\}$ according to (for details see e.g., [5])

$$P^{[\text{ext}]}_{\text{DM}}(\tilde{x}^{(i)}_{k,\tau}=b)\sim\sum_{\hat{y}\in\mathcal{Y}^i_b}P(z_{k,\tau}|\hat{y})\prod_{j=1,j\neq i}^{I}P^{[\text{apri}]}_{\text{DM}}\left(\tilde{x}^{(j)}_{k,\tau}=\bar{\mu^{-1}}(\hat{y})^{(j)}\right). \quad (1)$$

Each $P^{[\text{ext}]}_{\text{DM}}(\tilde{x})$ consists of the sum over all possible channel symbols $\hat{y}$ for which the i^{th} bit of the corresponding bit pattern $\tilde{\mathbf{x}}=\bar{\mu^{-1}}(\hat{y})$ is b. These channel symbols form the subset $\mathcal{Y}^i_b$ with $\mathcal{Y}^i_b=\{\mu([\tilde{x}^{(1)},\dots\tilde{x}^{(i)}])|\tilde{x}^{(i)}=b\}$. The conditional probability density $P(z_{k,\tau}|\hat{y})$ is given by $P(z_{k,\tau}|\hat{y})=(1/\pi\sigma_n^2)\exp(-\|z_{k,\tau}-\hat{y}\|^2/\sigma_n^2)$. $P^{[\text{apri}]}_{\text{DM}}$ is the extrinsic information provided by the other two receiver components. Expressed as L-value this is simply

$$L^{[\text{apri}]}_{\text{DM}}(x)=L^{[\text{ext}]}_{\text{SDSD}}(x)+L^{[\text{ext}]}_{\text{LDPC}}(x)\,, \quad (2)$$

where $L^{[\text{ext}]}_{\text{SDSD}}(x)=0$ for the non-systematic bits. In Fig. 1 this is indicated by the dashed line.

2) LDPC Decoder: For LDPC decoding we employ standard belief propagation with the sum-product algorithm [4, 14]. With our special scheme of small sub-frames with separate LDPC codes the decoding is done in parallel for the sub-frames. As a priori information the variable nodes $\mathcal{V}$ get

$$L^{[\text{apri}]}_{\text{LDPC}}(\mathcal{V})=L^{[\text{apri}]}_{\text{LDPC}}(x)=L^{[\text{ext}]}_{\text{SDSD}}(x)+L^{[\text{ext}]}_{\text{DM}}(x)\,, \quad (3)$$

where again $L^{[\text{ext}]}_{\text{SDSD}}(x)=0$ for the non-systematic bits. The LDPC decoder supplies the other receiver components with the *extrinsic* information $L^{[\text{ext}]}_{\text{LDPC}}$ of each variable node $\mathcal{V}$, which is the sum of the messages from the connected check nodes $\mathcal{C}$

$$L^{[\text{ext}]}_{\text{LDPC}}(\mathcal{V})=L^{[\text{apri}]}_{\text{LDPC}}(x)=\sum_{\mathcal{C}\text{ connected to }\mathcal{V}}L^{[\text{ext}]}(\mathcal{C})\,. \quad (4)$$

Of course, in the messages sent internally from a variable node $\mathcal{V}$ to a check node $\mathcal{C}$, the incoming message from this specific check node is excluded. For more details we refer to the literature [4, 12, 14].

3) Soft Decision Source Decoder (SDSD): The algorithm the SDSD uses to compute the extrinsic probabilities $P^{[\text{ext}]}_{\text{SDSD}}(v)$ is based on a fully connected trellis of the parameters. It has been derived in [7–9] and reads as

$$P^{[\text{ext}]}_{\text{SDSD}}(v^{(m)}_{\kappa,\tau}=b)\sim \quad (5)$$
$$\sum_{\mathbf{v}^{[\text{ext},m]}_{\kappa,\tau}}\gamma(\mathbf{v}^{[\text{ext},m]}_{\kappa,\tau})\sum_{\mathbf{v}_{\kappa,\tau-1}}P(\mathbf{v}^{[\text{ext},m]}_{\kappa,\tau}|\mathbf{v}_{\kappa,\tau-1},v^{(m)}_{\kappa,\tau}=b)\cdot\alpha_{\tau\text{-}1}(\mathbf{v}_{\kappa,\tau-1})\,.$$

This algorithm utilizes the a priori knowledge $P(\mathbf{v}_{\kappa,\tau}|\mathbf{v}_{\kappa,\tau-1})$ and the present *extrinsic* trellis edge transition probabilities $\gamma(\mathbf{v}^{[\text{ext},m]}_{\kappa,\tau})$, $\mathbf{v}_{\kappa,\tau}=\{v^{(m)}_{\kappa,\tau},\mathbf{v}^{[\text{ext},m]}_{\kappa,\tau}\}$. The probabilities $\alpha_{\tau\text{-}1}(\mathbf{v}_{\kappa,\tau-1})$ are computed in a forward recursion. Again, we refer to the literature for details [6, 8, 9]. The a priori information from the other receiver components for the systematic bits v,

$$L^{[\text{apri}]}_{\text{SDSD}}(v)=L^{[\text{ext}]}_{\text{LDPC}}(v)+L^{[\text{ext}]}_{\text{DM}}(v) \quad (6)$$

is used in the forward recursion.

4) Parameter Estimation: In the final step the available extrinsic information of all three components is added,

$$L^{[\text{ext}]}_{\Sigma}(v)=L^{[\text{ext}]}_{\text{SDSD}}(v)+L^{[\text{ext}]}_{\text{LDPC}}(v)+L^{[\text{ext}]}_{\text{DM}}(v)\,. \quad (7)$$

After combining the bit-wise *extrinsic* information $L^{[\text{ext}]}_{\Sigma}(v)$ to parameter-wise knowledge $P(\mathbf{v}|z_1,\dots z_\tau)$, the individual estimates $\hat{u}$ are computed by [6]

$$\hat{u}_{\kappa,\tau}=\sum_{\xi}\bar{u}^{(\xi)}_{\kappa}\cdot P(\mathbf{v}_{\kappa,\tau}\hat{=}\xi|z_1,\dots z_\tau) \quad (8)$$

according to the minimum mean square error (MMSE) fidelity criterion.

C. Tanner Graph of Turbo DeCodulation

An LDPC code is often described by its bipartite Tanner graph, which is also a graphical interpretation of its parity check matrix $\mathbf{H}_{\text{LDPC}}$. This Tanner graph contains two types of nodes, the variable nodes $\mathcal{V}$, representing the coded bits, and the check nodes $\mathcal{C}$, representing the parity check equations. If we consider the equations of the utilization of residual

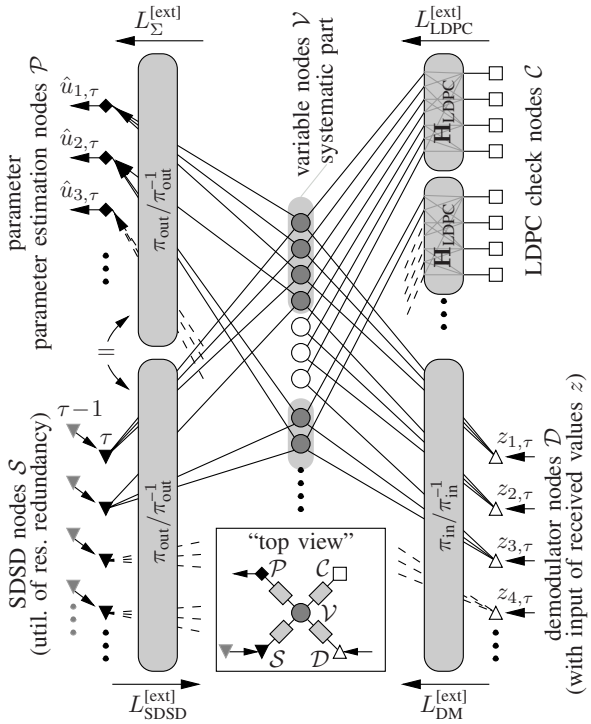

Fig. 2. Tanner graph of the Turbo DeCodulation system.

redundancy and the parameter estimation in the SDSD and the equation of the demodulator also as nodes $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{D}$, respectively, we can represent the complete Turbo DeCodulation receiver by a single Tanner graph at time instance τ as shown in Fig. 2.

This Tanner graph is now quadripartite and arranged around the variable nodes $\mathcal{V}$. Each $\mathcal{V}$ represents an LDPC encoded bit x and the $\mathcal{V}$ of systematic bits implicitly also the corresponding data bit v. All other types of nodes are solely connected to the variable nodes $\mathcal{V}$. For convenience, the messages passed on all edges connected to the variable nodes $\mathcal{V}$ shall be L-values of the respective bit.

The demodulator nodes $\mathcal{D}$ in the lower right part of Fig. 2 represent (1). Besides the L-values from the variable nodes $\mathcal{V}$ the demodulator nodes $\mathcal{D}$ get the respective received values $z_{k,\tau}$ as input. Each variable node $\mathcal{V}$ is connected to a demodulator node $\mathcal{D}$, because every encoded bit x obviously must be modulated. Note, we have drawn the edges between the $\mathcal{D}$ and $\mathcal{V}$ nodes sequentially to keep Fig. 2 clear. The interleaver block $\pi_{\text{in}}/\pi_{\text{in}}^{-1}$ provides for the permutation of the edges. The messages transmitted by the demodulator nodes $\mathcal{D}$ to the variable nodes $\mathcal{V}$ are the L-values $L_{\text{DM}}^{[\text{ext}]}$.

In the upper right part of Fig. 2 there are the check nodes $\mathcal{C}$ of the LDPC code(s). We can observe the special LDPC code structure with numerous short LDPC codes instead of one large monolithic LDPC code [12]. Via the other node types of the quadripartite Tanner graph the short LDPC codes can exchange information and thus indirectly form a single large LDPC code. This explains why this scheme achieves approximately the same performance as one large monolithic LDPC code. Again, the actual Tanner graph topologies are embodied in the blocks labeled with the parity check matrix $\mathbf{H}_{\text{LDPC}}$. The operation carried out at the check nodes $\mathcal{C}$ is classical belief propagation decoding [4, 14], e.g., by the sum-product algorithm. The messages sent from the check nodes $\mathcal{C}$ to the variable nodes $\mathcal{V}$ are the L-values $L_{\text{LDPC}}^{[\text{ext}]}$, comprising the complete extrinsic information generated from all check nodes $\mathcal{C}$ connected to a variable node $\mathcal{V}$. Note, every variable node $\mathcal{V}$ is connected to several check nodes $\mathcal{C}$. For lucidity we have drawn only a single edge between variable nodes $\mathcal{V}$ and check nodes $\mathcal{C}$. Actually the complex graph embodied by $\mathbf{H}_{\text{LDPC}}$ directly connects the $\mathcal{V}$ and $\mathcal{C}$ nodes.

In the lower left part of Fig. 2 we have the SDSD nodes $\mathcal{S}$, which represent (5). Since the SDSD acts only on the data bits v, only the systematic variable nodes $\mathcal{V}$ (filled circles) have a connection to an SDSD node $\mathcal{S}$ each. Due to the assumed residual redundancy in form of an autocorrelation, i.e., using $P(\mathbf{v}_{\kappa,\tau}|\mathbf{v}_{\kappa,\tau-1})$ in (5), we also need an edge to the corresponding SDSD node $\mathcal{S}$ of the previous time instance $\tau-1$ at the SDSD nodes $\mathcal{S}$. The messages from these preceding nodes are basically the respective right sum in (5). The interleaver block $\pi_{\text{out}}/\pi_{\text{out}}^{-1}$ embodies the edge permutation and the messages by the SDSD nodes $\mathcal{S}$ are the L-values $L_{\text{SDSD}}^{[\text{ext}]}$ of the data bits x.

In the upper left part of Fig. 2 we have the parameter estimation nodes $\mathcal{P}$, which represent (8). Note, the parameter estimation is actually part of SDSD as seen in Fig. 1, but we will consider it separately here because it is based on a different equation and accepts other messages, yielding a new type of graph node. Similarly to the SDSD nodes $\mathcal{S}$ the nodes $\mathcal{P}$ are only connected to the systematic variable nodes $\mathcal{V}$ and the edges are permuted by an interleaver block $\pi_{\text{out}}/\pi_{\text{out}}^{-1}$. Using (8) the nodes $\mathcal{P}$ finally deliver the estimated $\hat{u}_{\kappa,\tau}$. They are processed only at the end and get the L-values $L_{\Sigma}^{[\text{ext}]}$ as messages from the variable nodes $\mathcal{V}$.

At each point the variable nodes $\mathcal{V}$ reflect the so far obtained extrinsic information $L_{\Sigma}^{[\text{ext}]}(v)$ of their respective bits v (or x) according to (7). The variable nodes $\mathcal{V}$ also coordinate the iterative processing and deliver the respective a priori information to the demodulator nodes $\mathcal{D}$, check nodes $\mathcal{C}$, and SDSD nodes $\mathcal{S}$ (see (2), (3), and (6)).

III. Simulation Results

The capabilities of the proposed Turbo DeCodulation scheme shall be demonstrated by simulation. Instead of using any specific speech, audio, or video encoder, we model $K_S = 330$ statistically independent source codec parameters u by K_S independent 1st order Gauss-Markov processes with autocorrelation $\rho = 0.9$, a typical value for some parameters of source codecs. We assume that the receiver knows this value perfectly. A mismatch between the actual (at the transmitter) and the assumed (at the receiver) residual redundancy, e.g., autocorrelation, can have significant effects on the performance [15]. Each parameter $u_{\kappa,\tau}$ is scalarly quantized by a Lloyd-Max quantizer using $M = 3$ bits/parameter. The index assignment Γ is either natural binary (if SDSD stand-alone)

or parameter SNR optimized [16] (if SDSD in the loop). The latter one is optimized for the used ρ.

The bit-interleaved frame $\underline{\tilde{\mathbf{v}}}_\tau$ is channel encoded using a (21,11)-DSC (difference set cyclic) code [17] as short LDPC code, i.e., the $K_S M = 990$ interleaved bits of $\tilde{\mathbf{v}}_\tau$ are segmented in $990/11 = 90$ sub-frames and each sub-frame is separately encoded by the (21,11)-DSC code. It has been shown that DSC codes are suitable LDPC codes for such a short frame length [18]. With this (21,11)-DSC code the code rate of $r_{\text{LDPC}} = 11/21$ is close to the typical $r = 1/2$.

The modulator maps $I = 3$ bits to one channel symbol by either 8PSK-Gray (if demodulator stand-alone) or 8PSK-SP (set partitioning) mapping [5] (if demodulator in the loop). In [5] other mappings are presented, which exhibit a better asymptotic performance than 8PSK-SP. However, with the SDSD providing good extrinsic information to the channel decoder already at relatively low E_b/N_0 the 8PSK-SP mapping is the better choice for the given settings due to its waterfall region being at a likewise low E_b/N_0 [5].

At the Turbo DeCodulation receiver Ξ_{outer} outer iterations and Ξ_{in} inner iterations are performed. For the order of inner and outer iterations there exist lots of possibilities. For simplicity we set $\Xi_{\text{TD}} = \Xi_{\text{in}} = \Xi_{\text{out}}$ and restrict ourselves to the case in which demodulator, channel decoder and SDSD are executed sequentially in this order. This results in single inner and outer iterations being performed alternately. This can be seen as both iterative loops being executed in parallel, i.e., the LDPC decoder is feeding both loops simultaneously. To efficiently benefit from the LDPC decoding we execute $\Xi_{\text{LDPC}} = 3$ internal belief propagation iterations each time the LDPC decoder is activated. This small Ξ_{LDPC} is sufficient due to the small size of the short sub-frames. When considering the Tanner graph view of the receiver of Fig. 2, in each overall iteration first the demodulator nodes $\mathcal{D}$, then three times the check nodes $\mathcal{C}$, and finally, the SDSD nodes $\mathcal{S}$ are activated.

Beside the Turbo DeCodulation system and its non-iterative baseline variant, we additionally define an ISCD and a BICM-ID configuration for comparison, which only employ outer *or* inner iterations, respectively. In Table I an overview of the most important differences in the simulation settings is given. The parameter signal-to-noise ratio (SNR) between the originally generated parameters $u_{\kappa,\tau}$ and the reconstructed estimates $\hat{u}_{\kappa,\tau}$ is used for quality evaluation. The simulation results are depicted in Fig. 3.

We can observe that already employing soft decision source decoding in a non-iterative system yields a noticeable gain compared to a hard decision after channel decoding, which is typically done in today's communication systems. When adding a single iterative loop with BICM-ID or ISCD an additional gain of $\Delta_{E_b/N_0} = 1.7$ dB respectively $\Delta_{E_b/N_0} = 2.1$ dB can be achieved at an arbitrary reference parameter SNR of 13 dB. We assume that below this reference parameter SNR the quality is not sufficient and other measures must be taken.

With the proposed Turbo DeCodulation scheme exploiting the iterative possibilities of the complete Tanner graph of Fig. 2 (i.e., three iterative loops in Fig. 1) the gain with respect

TABLE I
SIMULATION SETTINGS FOR FIG. 3

Configuration	Index Assignment Γ	Mapping μ	Ξ_{out}	Ξ_{in}
Turbo DeCod.	Parameter SNR opt.	8PSK-SP	10	10
ISCD	Parameter SNR opt.	8PSK-Gray	10	1
BICM-ID	Natural Binary	8PSK-SP	1	10
non-iterative	Natural Binary	8PSK-Gray	1	1

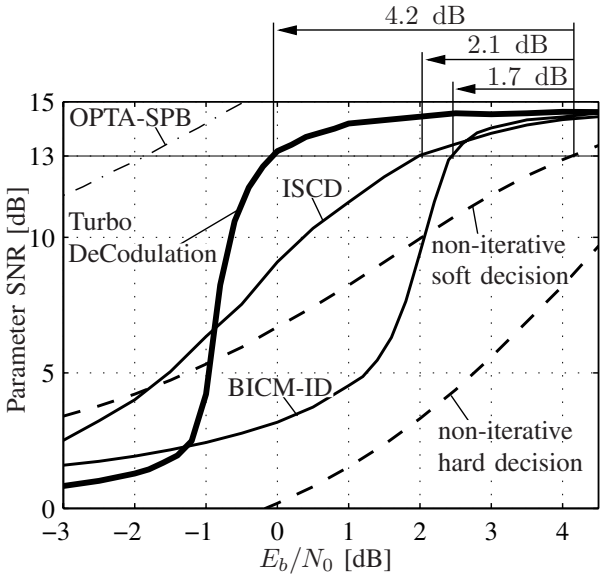

Fig. 3. Parameter SNR Simulation Results with Gauss-Markov Source

to the non-iterative soft decision case increases significantly further to $\Delta_{E_b/N_0} = 4.2$ dB. This is more than the combined gain of BICM-ID and ISCD, proving the effectiveness of the presented graph-based Turbo DeCodulation. The dash-dotted line in Fig. 3 depicts the theoretical limit for all simulated schemes [19]. This optimum performance theoretically attainable (OPTA) is obtained by combining the channel capacity with the rate-distortion function. Additionally, the sphere packing bound (SPB) is considered to incorporate the effects of the finite frame size. The proposed Turbo DeCodulation exploits a major part of the gap between the previously known systems BICM-ID and ISCD and the OPTA-SPB limit.

To analyze the convergence behavior of the Turbo DeCodulation system we can employ a three-dimensional extrinsic information transfer (EXIT) chart [11, 13]. The EXIT characteristics $\mathcal{T}$ give the maximum extrinsic mutual information $\mathcal{I}^{[\text{ext}]}$ of a bit that can be generated by a receiver component using the a priori extrinsic mutual information $\mathcal{I}^{[\text{ext}]}$ from the other components. The tunnel (and its length) between the EXIT characteristics indicates the potential for iterative performance improvements which can be exploited by the decoding trajectory of the Turbo process of the combined receiver. In Fig. 4 the EXIT chart for the Turbo DeCodulation system in Fig. 3 is depicted at $E_b/N_0 = 0$ dB and a narrow, but long tunnel can be observed. After several iterations the decoding trajectory reaches the end of the tunnel. This stopping intersection occurs at relatively high $\mathcal{I}^{[\text{ext}]}$, especially $\mathcal{I}^{[\text{ext}]}_{\text{LDPC}}$. Note that $\mathcal{T}(\text{SDSD})$ is only outlined between $\mathcal{T}(\text{DM})$

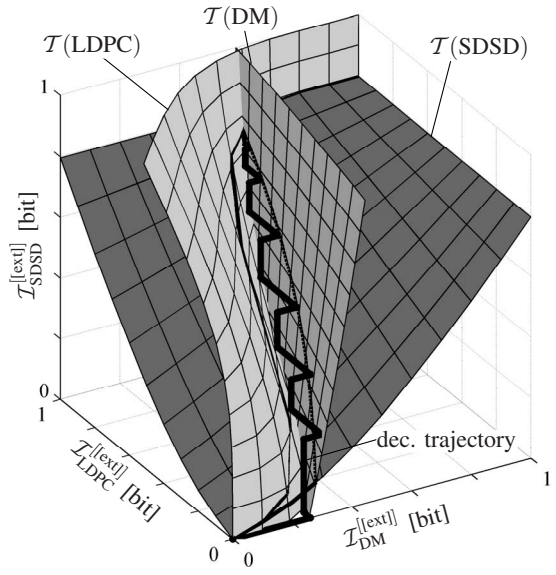

Fig. 4. EXIT Chart for Turbo DeCodulation in Fig. 3 at $E_b/N_0 = 0$ dB.

and $\mathcal{T}(\mathrm{LDPC})$ to allow a view on the decoding trajectory and for the LDPC code we use an overall EXIT characteristic (taking Ξ_{LDPC} into account) and not one of the "internal" EXIT characteristics used in the analysis of stand-alone LDPC codes.

IV. Implementational Aspects

When considering the Tanner graph in Fig. 2 some advantageous aspects regarding the implementation of the proposed graph-based Turbo DeCodulation can be derived. Obviously, parallel processing of all nodes (of one type or even of all types) is possible as long as we manage to distribute the messages via the variable nodes correctly. In contrast, with a convolutional code sequential processing of the complete frame (or potentially some large sub-frames with a window-based convolutional decoder) in channel decoding is required. The Tanner graph view also emphasizes the parallel nature of demodulator and SDSD nodes.

The parallel processing of the check nodes is of course also possible with a large monolithic LDPC code. But with the used separation into short sub-codes we have additionally numerous identical sub-blocks and we can decide how many instances of these short LDPC decoders we actually implement on a chip. Furthermore, a high degree of flexibility in the total frame size is obtained, as it can be easily modified in the granularity of the sub-frame size. Only the interleavers need to be adapted. A single large LDPC code usually has to be completely redesigned for a different frame size and typically, there do not exist such identical sub-blocks, even though LDPC codes in practical systems existing today, such as IEEE 802.16 and DVB-S2, are not completely random in order to simplify the implementation. Note, the separation into short sub-codes is only useful if an interaction by a Turbo process with at least a second receiver component exists [12]. This way the graphs of the sub-codes are connected to one large graph.

V. Conclusion

We presented the graph-based Turbo DeCodulation, which uses an LDPC code as channel code in the Turbo DeCodulation scheme of combined joint demodulation, channel decoding and source decoding. We employed a special LDPC coding scheme partitioned in short sub-codes, which offers a high degree of flexibility for the overall frame size and the usage of several decoder instances to parallelize the processing. Considering the presented quadripartite Tanner graph of Turbo DeCodulation we observed that all nodes (demodulator nodes, check nodes, and SDSD nodes) could be processed in parallel. In a simulation example we demonstrated the excellent capabilities of the proposed receiver design. The gain of Turbo DeCodulation surpasses the combined gain of the sub-systems BICM-ID and ISCD and we can approach the theoretical OPTA-SPB limit. The desired convergence behavior was confirmed by a three dimensional EXIT chart.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding," in *IEEE ICC*, Geneva, 1993.

[2] J. Hagenauer, E. Offer, and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes," *IEEE Trans. Comm.*, pp. 429–445, Mar. 1996.

[3] R. G. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, pp. 21–28, Jan. 1962.

[4] D. J. C. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Trans. Inform. Theory*, pp. 399–431, Mar. 1999.

[5] X. Li, A. Chindapol, and J. A. Ritcey, "Bit-Interleaved Coded Modulation With Iterative Decoding and 8PSK Signaling," *IEEE Trans. Comm.*, pp. 1250–1257, Aug. 2002.

[6] T. Fingscheidt and P. Vary, "Softbit Speech Decoding: A New Approach to Error Concealment," *IEEE Trans. Speech Audio Proc.*, Mar. 2001.

[7] P. Vary and R. Martin, *Digital Speech Transmission - Enhancement, Coding & Error Concealment*. Wiley, 2006.

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

[9] N. Görtz, "On the Iterative Approximation of Optimal Joint Source-Channel Decoding," *IEEE J. Select. Areas Commun.*, Sept. 2001.

[10] T. Clevorn, J. Brauers, M. Adrat, and P. Vary, "Turbo DeCodulation: Iterative Combined Demodulation and Source-Channel Decoding," *IEEE Comm. Lett.*, pp. 820–822, Sept. 2005.

[11] T. Clevorn, J. Brauers, M. Adrat, and P. Vary, "EXIT Chart Analysis of Turbo DeCodulation," in *IEEE PIMRC*, Berlin, Germany, Sept. 2005.

[12] T. Clevorn, F. Oldewurtel, and P. Vary, "Combined Iterative Demodulation and Decoding using Very Short LDPC Codes and Rate-1 Convolutional Codes," in *CISS*, Baltimore, MD, USA, Mar. 2005.

[13] S. ten Brink, "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes," *IEEE Trans. Comm.*, pp. 1727–1737, Oct. 2001.

[14] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient Implementations of the Sum-Product Algorithm for Decoding LDPC Codes," in *IEEE Globecom*, San Antonio, TX, USA, Nov. 2001.

[15] B. Schotsch, "Mismatch of First Order A Priori Information in Iterative Source-Channel Decoding," in *Poster*, Prague, Czech Republic, May 2007.

[16] T. Clevorn, P. Vary, and M. Adrat, "Parameter SNR Optimized Index Assignments and Quantizers based on First Order A Priori Knowledge for Iterative Source-Channel Decoding," in *CISS*, Princeton, NJ, USA, Mar. 2006.

[17] E. J. Weldon, "Difference-Set Cyclic Codes," *Bell Syst. Tech. J.*, pp. 1045–1055, Sept. 1966.

[18] R. Lucas, M. Fossorier, Y. Kou, and S. Lin, "Iterative Decoding of One-Step Majority Logic Decodable Codes based on Belief Propagation," *IEEE Trans. Comm.*, pp. 931–937, June 2000.

[19] T. Clevorn, L. Schmalen, P. Vary, and M. Adrat, "On the Optimum Performance Theoretically Attainable for Scalarly Quantized Correlated Sources," in *ISITA*, Seoul, Korea, Oct. 2006.