Low Complexity Syndrome Based Decoding of Turbo Codes

Jan Geldmacher, Klaus Hueske, and Jürgen Götze
Information Processing Lab, TU Dortmund University
Otto-Hahn-Strasse 4, 44227 Dortmund, Germany.
Correspondence: jan.geldmacher@ieee.org

Martin Kosakowski
Research In Motion Deutschland GmbH
Universitätsstr. 140
44799 Bochum, Germany.

***Abstract*—A new syndrome trellis based decoding approach for Turbo coded data is described in this paper. Based on estimating error symbols instead of code symbols, it inherently features options for reducing the computational complexity of the decoding process. After deriving required transition metrics, numerical results in terms of block error rate and required equivalent iterations are presented to demonstrate the efficiency of the approach.**

I. Introduction

Forward error correction based on Turbo coding has been adopted in numerous applications since its initial presentation [1]. Especially in mobile communication systems like the Long Term Evolution (LTE) system their error correcting performance is an integral part for enabling high data throughput. On the downside, however, is the high computational complexity involved with the decoder's iterative implementation, which makes it one of the most complex and power consuming parts in the receiver baseband signal processing. Therefore it is important to keep the computational effort as small as possible and to avoid unnecessary iterations of the decoder. This can be achieved by using early termination (ET) schemes, which stop the iteration process if a block is not decodable (Low SNR ET) or already successfully decoded (High SNR ET). Several methods have been proposed to achieve this, see for example [2]–[4]. An effective ET scheme leaves the highest number of iterations to the medium SNR range, the waterfall region of the code. Unfortunately this is also often the desired working point of the system – in the LTE system for example, the CQI reporting adaptively keeps the system at a target block error rate (BLER) of about 10%, which corresponds to the waterfall region for common block lengths.

In this paper a syndrome trellis based MAP decoding algorithm is proposed. While its trellis complexity and performance in terms of the generated soft information is equivalent to the conventional encoder trellis based MAP algorithm, it can be modified to reduce the computational effort during Turbo decoding. Especially in the medium SNR range, where many iterations are required to decode a received block, a significant reduction can be achieved, such that the proposed approach can complement High and Low SNR ET schemes.

A key property of the underlying syndrome based decoding algorithm are the unbalanced probabilities of the trellis states, which tend to the all-zero state if the input sequence contains fewer errors. Combining this with a syndrome based state identification [5] allows to estimate possible error-free subblocks in the input sequence. The MAP algorithm then only needs to process the remaining erroneous blocks.

The basic idea of this so-called block syndrome decoder (BSD) principle has been described previously for Viterbi decoding [6] and Turbo equalization [7]. The extension to Turbo decoding as presented in this work requires a modification of the original syndrome based MAP decoding algorithm: In order to achieve a meaningful syndrome based state estimation, we propose a precorrection scheme, which corrects hard decision errors in the systematic and parity part with ongoing iterations. The transition metric involved with this modification is derived in the following Sec. II. The resulting Turbo decoding framework is described in Sec. III. Sec. IV briefly reviews ET and the BSD principle and its application in context of Turbo decoding. And finally numerical results and conclusions are given in Sec. V and Sec. VI, respectively.

II. Syndrome Based MAP Decoding

A. Derivation of Transition Metric

Given a k/n-rate binary convolutional code $\mathcal{C}$, then $\mathbf{H}^T$ is a syndrome former of $\mathcal{C}$, if $\mathbf{G}\mathbf{H}^T = \mathbf{0}$, where $\mathbf{G}$ represents an encoder of $\mathcal{C}$. Let $\mathbf{r}$ denote the binary hard decision of a received sequence,

$$\mathbf{r} = \mathbf{u}\mathbf{G} \oplus \boldsymbol{\epsilon}_c = \mathbf{v} \oplus \boldsymbol{\epsilon}_c, \tag{1}$$

where $\mathbf{u}$ denotes the original information sequence and $\boldsymbol{\epsilon}_c$ the hard decision of the channel error. Applying an arbitrary sequence $\mathbf{x}$ to $\mathbf{r}$ and using $\mathbf{H}^T$ yields the syndrome $\mathbf{b}$,

$$\mathbf{b} = (\mathbf{r} \oplus \mathbf{x})\mathbf{H}^T. \tag{2}$$

Then each sequence $\mathbf{e}$ corresponding to each path in the trellis representation of $\mathbf{H}^T$ subject to $\mathbf{b}$ represents an admissible error sequence for $(\mathbf{r} \oplus \mathbf{x})$, i.e. $(\mathbf{r} \oplus \mathbf{x} \oplus \mathbf{e}) \in \mathcal{C}$, and $\mathbf{x} \oplus \mathbf{e} = \boldsymbol{\epsilon}$ denotes an admissible error sequence for $\mathbf{r}$. The objective here is to find an estimate for $\boldsymbol{\epsilon}$, given $\mathbf{r}$, an arbitrary $\mathbf{x}$ and the trellis of $\mathbf{H}^T$ subject to $\mathbf{b}$.

More specifically, the proposed syndrome based MAP decoder maximizes the *a posteriori* probability $P(\mathbf{e}_t|\tilde{\mathbf{r}}, \mathbf{x})$ of an error symbol $\mathbf{e}_t$ given the received soft decision sequence $\tilde{\mathbf{r}}$ and a binary sequence $\mathbf{x}$. $\mathbf{e}_t$ is an element of $\mathbf{e}$ at time instant

t. The probability $P(\mathbf{e}_t = \mathbf{e}_l|\tilde{\mathbf{r}}, \mathbf{x})$ for some error symbol $\mathbf{e}_l$, $l = 1 \ldots 2^n$, can be expressed using all transitions $\mathcal{T}_t^{\mathbf{e}_l}$ in the trellis of $\mathbf{H}^T$ that lead to an error symbol $\mathbf{e}_l$ at time instant t:

$$P(\mathbf{e}_t = \mathbf{e}_l|\tilde{\mathbf{r}}, \mathbf{x}) = \sum_{(p,q)\in\mathcal{T}_t^{\mathbf{e}_l}} P(\Psi_t = p, \Psi_{t+1} = q|\tilde{\mathbf{r}}, \mathbf{x}) \quad (3)$$

In (3), $\Psi_t = p$ and $\Psi_{t+1} = q$ denote trellis states at time instants t and $t+1$, respectively. It is well known that $P(\Psi_t = p, \Psi_{t+1} = q|\tilde{\mathbf{r}}, \mathbf{x})$ can be efficiently computed by performing forward and backward recursions on the trellis [8]. Thus, using the forward and backward state probabilities $\alpha_t(p)$ and $\beta_{t+1}(q)$ of states p and q yields

$$P(\Psi_t = p, \Psi_{t+1} = q|\tilde{\mathbf{r}}, \mathbf{x}) \sim \alpha_t(p)\gamma_t(p, q)\beta_{t+1}(q). \quad (4)$$

The probability $\gamma_t(p, q)$ of a transition from state p to state q at time instant t is given as

$$\begin{aligned}\gamma_t(p, q) &= p(\tilde{\mathbf{r}}_t|\Psi_t = p, \Psi_{t+1} = q, \mathbf{x}_t)P(\Psi_t = p|\Psi_{t+1} = q)\\ &= p(\tilde{\mathbf{r}}_t|\mathbf{e}_{(p,q)}, \mathbf{x}_t)P(\mathbf{e}_t = \mathbf{e}_{(p,q)}, \mathbf{x}_t). \quad (5)\end{aligned}$$

The error symbol associated with the transition from p to q is denoted by $\mathbf{e}_{(p,q)}$ and $p(\tilde{\mathbf{r}}_t|\mathbf{e}_{(p,q)}, \mathbf{x}_t)$ is determined by the channel, while $P(\mathbf{e}_t = \mathbf{e}_{(p,q)}, \mathbf{x}_t)$ denotes the *a priori* probability of $\mathbf{e}_t$. The "precorrection" symbol is denoted by $\mathbf{x}_t$. Further factorization yields

$$\gamma_t(p, q) = \prod_{i=1}^{n} p(\tilde{r}_t^{(i)}|e_{(p,q)}^{(i)}, x_t^{(i)}) \prod_{j=1}^{k} P(e_t^{(j)} = e_{(p,q)}^{(j)}, x_t^{(j)}), \quad (6)$$

where we assume that an error symbol consists of n bit, and that *a priori* information is available for the first k bit.

In the following, a transmission over an AWGN channel with noise power σ_n^2 and energy per coded bit E_c is assumed. Also, to simplify the notation, $p(\tilde{r}_t^{(i)}|e_{(p,q)}^{(i)}, x_t^{(i)}) =: p(\tilde{r}|e, x)$ is used in the following. In order to find $p(\tilde{r}|e, x)$, the probabilities of the different combinations of $\tilde{r} > 0$, $\tilde{r} \leq 0$, $e = 0$, $e = 1$, $x = 0$ and $x = 1$ have to be combined to one expression. For example, letting $e = 0$ yields the probabilities of the corresponding four combinations:

$$p(\tilde{r} > 0|e = 0, x = 0) \sim \exp(-(\tilde{r} - \sqrt{E_c})^2/(2\sigma_n^2)) \quad (7)$$
$$p(\tilde{r} \leq 0|e = 0, x = 0) \sim \exp(-(\tilde{r} + \sqrt{E_c})^2/(2\sigma_n^2)) \quad (8)$$
$$p(\tilde{r} > 0|e = 0, x = 1) \sim \exp(-(\tilde{r} + \sqrt{E_c})^2/(2\sigma_n^2)) \quad (9)$$
$$p(\tilde{r} \leq 0|e = 0, x = 1) \sim \exp(-(\tilde{r} - \sqrt{E_c})^2/(2\sigma_n^2)) \quad (10)$$

Combining (7), (8), and (9), (10), results in

$$p(\tilde{r}|e = 0, x = 0) \sim \exp(-(-|\tilde{r}| + \sqrt{E_c})^2/(2\sigma_n^2)) \quad (11)$$
$$p(\tilde{r}|e = 0, x = 1) \sim \exp(-(|\tilde{r}| + \sqrt{E_c})^2/(2\sigma_n^2)). \quad (12)$$

Further, using $\tilde{x} = 2x - 1$ simplifies (11) and (12) to

$$p(\tilde{r}|e = 0, x) \sim \exp(-(\tilde{x}|\tilde{r}| + \sqrt{E_c})^2/(2\sigma_n^2)). \quad (13)$$

In the same way, the probability $p(\tilde{r}|e = 1, x)$ of recieving $\tilde{r}$ given $e = 1$ and x, can be derived as

$$p(\tilde{r}|e = 1, x) \sim \exp(-(\tilde{x}|\tilde{r}| - \sqrt{E_c})^2/(2\sigma_n^2)). \quad (14)$$

Finally combining (13) and (14) using $\tilde{e} = 2e - 1$ results in

$$p(\tilde{r}|e, x) \sim \exp(-(\tilde{x}|\tilde{r}| - \sqrt{E_c}\tilde{e})^2/(2\sigma_n^2)). \quad (15)$$

Under the assumption, that the *a priori* probability $P(\epsilon)$ of an error is supplied in the form of an LLR $L_A(\epsilon)$,

$$L_A(\epsilon) = \log \frac{P(\epsilon = 1)}{P(\epsilon = 0)}, \quad (16)$$

the *a priori* probabilities for the cases $e = 0$, $e = 1$, $x = 0$ and $x = 1$ can be combined to the following expression

$$\begin{aligned}P(e, x) &= \frac{\exp(-L_A(\epsilon)/2)}{1 + \exp(-L_A(\epsilon))} \exp(-\tilde{x}\tilde{e}L_A(\epsilon)/2)\\ &\sim \exp(-\tilde{x}\tilde{e}L_A(\epsilon)/2). \quad (17)\end{aligned}$$

Putting (15) and (17) into (6) yields the transition probability

$$\begin{aligned}\gamma_t(p, q) \sim &\prod_{i=1}^{n} \exp(-(\tilde{x}_t^{(i)}|\tilde{r}_t^{(i)}| - \sqrt{E_c}\tilde{e}_{(p,q)}^{(i)})^2/(2\sigma_n^2))\\ &\cdot \prod_{j=1}^{k} \exp\left(-\tilde{x}_t^{(j)}\tilde{e}_{(p,q)}^{(j)}L_A(\epsilon_t^{(j)})/2\right). \quad (18)\end{aligned}$$

Note that some scaling factors have been skipped in the derivation such that suitable normalization would be required in order to get true probabilities.

Based on (18) a log domain formulation is readily found as

$$\begin{aligned}\log \gamma_t(p, q) \sim &-\frac{1}{2\sigma_n^2} \sum_{i=1}^{n} (\tilde{x}_t^{(i)}|\tilde{r}_t^{(i)}| - \sqrt{E_c}\tilde{e}_{(p,q)}^{(i)})^2\\ &- \sum_{j=1}^{k} \tilde{x}_t^{(j)}\tilde{e}_{(p,q)}^{(j)}L_A(\epsilon_t^{(j)})/2, \quad (19)\end{aligned}$$

and using the fact that terms independent of $\tilde{e}_{(p,q)}^{(j)}$ and $\tilde{x}_t^{(j)}$ have no influence on a maximization, the final transition metric $\Gamma_t(p, q)$ from state p to state q at time instant t becomes

$$\Gamma_t(p, q) = \frac{\sqrt{E_c}}{\sigma_n^2} \sum_{i=1}^{n} \tilde{x}_t^{(i)}|\tilde{r}_t^{(i)}|\tilde{e}_{(p,q)}^{(i)} - \frac{1}{2} \sum_{j=1}^{k} \tilde{x}_t^{(j)}\tilde{e}_{(p,q)}^{(j)}L_A(\epsilon_t^{(j)}). \quad (20)$$

B. Relation to Other Work

Schalkwijk et. al. have proposed syndrome decoding for convolutional codes [9], where the trellis of $\mathbf{H}^T$ is constructed subject to $\mathbf{b} = \mathbf{r}\mathbf{H}^T$. Tajima et. al. have derived soft decision transition metrics for this approach [10], [11]. On the other hand, constructing the trellis of $\mathbf{H}^T$ subject to $\mathbf{b} = \mathbf{0}$ [12]–[14], results in a trellis where each path represents a element of $\mathcal{C}$, and where the conventional MAP transition metric can be adopted. The main objective of this approach is a possible reduction of trellis complexity for high code rates.

Both methods can be found as special cases of the approach proposed in this paper, by selecting $\mathbf{x} = \mathbf{0}$ or $\mathbf{x} = \mathbf{r}$. However, instead of selecting a fixed value for $\mathbf{x}$, we propose to select $\mathbf{x}$ as an estimate of the channel error $\boldsymbol{\epsilon}_c$. Therefor, if $\mathbf{x}$ is a suitable "precorrection" for $\mathbf{r}$, the trellis state probabilities are dominated by the all zero state. This property can be exploited to achieve a reduction of computational complexity as will be described in Sec. IV-B.

III. Turbo Decoding

In the following subsection the operation of one constituent decoder is summarized, while Subsection III-B provides an overview of the resulting Turbo decoding framework with precorrection. A binary Turbo code based on two parallel concatenated recursive systematic convolutional codes is assumed. Puncturing may be employed on transmitter side. In this case a depuncturing is realized on receiver side by placing zero reliability softbits at the punctured positions. The energy per transmitted bit is set to $E_c = 1$, for simplicity of notation.

A. Implementation of Constituent Decoder

A constituent decoder takes the received soft decision sequence $\tilde{\mathbf{r}}$, a precorrection sequence $\mathbf{x}$ and *a priori* LLRs of the errors $L_A(\boldsymbol{\epsilon})$ in the systematic part as input. For the systematic part, it generates an *a posteriori* hard decision estimate $\hat{\boldsymbol{\epsilon}}$ of the error in the hard decision $\mathbf{r}$ of $\tilde{\mathbf{r}}$ and extrinsic LLRs $L_E(\boldsymbol{\epsilon})$. The decoder performs the following steps:

1) **Precorrection and syndrome computation** The precorrection sequence $\mathbf{x}$ is applied to the hard decision $\mathbf{r}$ of $\tilde{\mathbf{r}}$, and the syndrome $\mathbf{b}$ is computed as

$$\mathbf{b} = (\mathbf{r} \oplus \mathbf{x})\mathbf{H}^T. \tag{21}$$

2) **Trellis operation** Given the trellis of $\mathbf{H}^T$ subject to $\mathbf{b}$, the LogMAP or MaxLog algorithm is applied to generate an *a posteriori* estimate $L(\boldsymbol{\epsilon})$:
The initial state for the forward recursion is set to the zero state, while the initial state for the backward recursion is selected according to the final state of the syndrome former from (21). The LogMAP or MaxLog algorithm based on the transition metric (20) is applied, resulting in the *a posteriori* LLR $L(\mathbf{e})$. Note that in case of the MaxLog algorithm, the noise power σ_n^2 can be neglected in (20), because the result of the MaxLog algorithm is independent of any scaling factor.
In order to generate the LLR $L(\boldsymbol{\epsilon})$ of the absolute error in $\mathbf{r}$, the sign of $L(\mathbf{e})$, which is an estimate of the error in $\mathbf{r} \oplus \mathbf{x}$, has to be flipped according to $\mathbf{x}$,

$$L(\epsilon_t^{(j)}) = -\tilde{x}_t^{(j)} L(e_t^{(j)}) \text{ for } j = 1 \dots n. \tag{22}$$

The hard decision $\hat{\boldsymbol{\epsilon}}$ of $L(\boldsymbol{\epsilon})$ delivers the *a posteriori* estimate of the channel error $\boldsymbol{\epsilon}_c$. Applying the systematic part of $\hat{\boldsymbol{\epsilon}}$ to the systematic part of $\mathbf{r}$ results in the estimate $\hat{\mathbf{u}}$ of the original information bits $\mathbf{u}$.

3) **Generation of Extrinsic LLR** Extrinsic LLRs $L_E(\boldsymbol{\epsilon})$ are generated by removing the *a priori* information $L_A(\boldsymbol{\epsilon})$ and the received softbits from $L(\epsilon_t^{(j)})$,

$$L_E(\epsilon_t^{(j)}) = \begin{cases} L(\epsilon_t^{(j)}) + \frac{2}{\sigma_n^2}|\tilde{r}_t^{(j)}|, & j > k \\ L(\epsilon_t^{(j)}) + \frac{2}{\sigma_n^2}|\tilde{r}_t^{(j)}| - L_A(\epsilon_t^{(j)}), & \text{else.} \end{cases}$$

The systematic extrinsic LLRs are (de-)interleaved and become the *a priori* LLR of the other constituent decoder.

B. Turbo Decoder with Precorrection

Like in the conventional Turbo decoder, both constituent decoders use the systematic and their according parity part of the received softbits, and, as priors, the interleaved extrinsic LLRs of the systematic part generated by the other decoder. Additionally, the precorrection sequence $\mathbf{x}$ is used.

Generally an arbitrary sequence can be selected for $\mathbf{x}$, without changing the absolute values of the generated LLRs. However, as mentioned before, here it is desired to make $\mathbf{b}$ a function of the remaining errors in $\mathbf{r}$. This can be achieved by selecting $\mathbf{x}$ as an extrinsic estimate of the channel error $\boldsymbol{\epsilon}_c$, which in context of the Turbo decoding framework can be done as follows:

- The systematic part of $\mathbf{x}$ is set to the hard decision of the *a priori* values $L_A(\boldsymbol{\epsilon})$. This estimate stems from the other constituent decoder.
- The parity part of $\mathbf{x}$ is set to the extrinsic estimate of the parity error from the previous iteration, from the same constituent decoder.

As a result the sequence $\mathbf{x}$ depends on both constituent decoders. In case of convergence, it therefore leads to a decreasing hamming weight of the syndrome sequence $\mathbf{b}$ with ongoing iterations.

It is important to note that up to this point the described syndrome based Turbo decoder is identical to the conventional decoder in terms of decoding performance and trellis complexity. Additional effort is required to compute the sequence $\mathbf{x}$ and the syndrome $\mathbf{b}$. However, both are low complexity, binary operations.

IV. Reduction of Decoding Complexity

In order to reduce the computational complexity of the decoder, ET for high and low SNR scenarios can be adopted. Furthermore, we propose the BSD approach to reduce the average number of iterations for medium to high SNR. Both approaches will now be reviewed briefly in context of the syndrome based Turbo decoder.

A. Early Termination

Several ET schemes have been proposed, which may be classified into

- hard or soft decision based criteria,
- threshold-based or adaptive criteria, and
- Low and/or High SNR criteria.

In practice a low complexity criterion capable of Low and High SNR ET is preferred. One might also want to avoid a threshold, because it can be difficult to select suitable values for all possible states of a system.

In this work a modified version of the improved hard decision aided (IHDA) criterion [15] is applied. Although originally only described for High SNR ET, it can be easily extended to also cover Low SNR ET. For the syndrome based Turbo decoder it may be implemented in the following way:

- Count the sign differences Δ_i between the *a posteriori* LLRs $L_1(\boldsymbol{\epsilon})$ and $L_2(\boldsymbol{\epsilon})$ of the first and second constituent decoder, respectively, after each full iteration i.

- Terminate the decoding process after the i-th iteration for $i > 1$ if
  - $\Delta_i = 0$ (High SNR ET),
  - $\Delta_i \geq \Delta_{i-1}$ (Low SNR ET), or
  - $i = i_{max}$ (max. number of iterations i_{max} reached).

This criterion performs well for low and high SNR, is independent of a threshold and keeps the implementation overhead small. Note however, that any other criterion, which has been proposed for conventional MAP decoding can be adopted for syndrome based decoding as well.

B. Block Syndrome Decoding

Based on the described syndrome based Turbo decoder, the BSD concept [6], [7] can be applied to achieve a reduction of decoding effort. Because of the precorrection, the syndrome sequence $\mathbf{b}$ of a constituent decoder shows subsequences of consecutive zeros, whose length and number increase with ongoing iterations in case of convergence. As described in Sec. I and in [6], [7], this can be exploited to separate the input sequence into subblocks that are considered to be erroneous and subblocks that are considered to be error-free. Consequently, a reduction of decoding effort can be achieved by only processing the erroneous subblocks and neglecting the supposedly error-free subblocks.

More precisely the syndrome based Turbo decoding algorithm from Sec. III is extended as follows:

1) **Preprocessing of syndrome** Identify subsequences of length $\geq \ell_{min}$ zeros in $\mathbf{b}$. Consider the corresponding subblocks in $\tilde{\mathbf{r}}$, except a padding of $\lfloor \ell_{min}/2 \rfloor$ at the beginning and end of each subblock, as error-free and the remaining subblocks as erroneous.
2) **Processing of blocks**
   a) *Erroneous blocks* The erroneous subblocks are processed by the syndrome based MAP decoder, which generates extrinsic values $L_E(\epsilon_t^{(j)})$ and the estimated error $\hat{\epsilon}_t^{(j)}$ for all t in these subblocks as described in Sec. III-A. Note that these blocks can be considered to be terminated in the zero state, because the zero state is the most likely state in the preceding and succeeding error-free blocks.
   b) *Error-free blocks* No processing is required for the supposedly error-free blocks. Instead, the extrinsic LLR is set to a sufficiently large value $c > 0$,

$$L_E(\epsilon_t^{(j)}) = \tilde{x}_t^{(j)} c, \ j = 1 \ldots n, \qquad (23)$$

   and the estimated error $\hat{\boldsymbol{\epsilon}}$ is set according to the precorrection sequence

$$\hat{\epsilon}_t^{(j)} = x_t^{(j)}, \ j = 1 \ldots n. \qquad (24)$$

   A reasonable choice for c is the largest value in the quantization range of the LLR values.

The choice of the design parameter ℓ_{min} affects the achievable reduction of decoding effort and the possible loss in decoding performance due to falsely classified error-free blocks. Given an acceptable loss in BLER or BER, it may be selected heuristically. It is important to note that ℓ_{min} is dependent on the underlying code (and puncturing scheme), and less dependent on other system parameters like block length or modulation type. Thus it is not prohibitive to select it in advance for the considered system and its code rates.

V. Numerical Results

This section shows simulation results for performance and computational complexity in terms of BLER and the average number of required iterations.

A. Simulation Parameters

The following simulation results are based on a binary Turbo code using parallel concatenation of two UMTS/LTE compatible recursive systematic encoders $\mathbf{G}(D) = \left[1, \ \frac{1+D+D^3}{1+D^2+D^3}\right]$. A corresponding syndrome former[1] is $\mathbf{H}^T(D) = \left[1+D+D^3, \ 1+D^2+D^3\right]^T$. Two code rates are evaluated: $R = 1/3$ and $R = 1/2$, where the latter is generated by puncturing odd and even parity bits of the first and second encoder, respectively. A pseudo-random interleaver of length 6144 is used, along with BPSK modulated transmission over an AWGN channel. The decoder is based on the MaxLog algorithm and maximum number of $i_{max} = 8$ iterations.

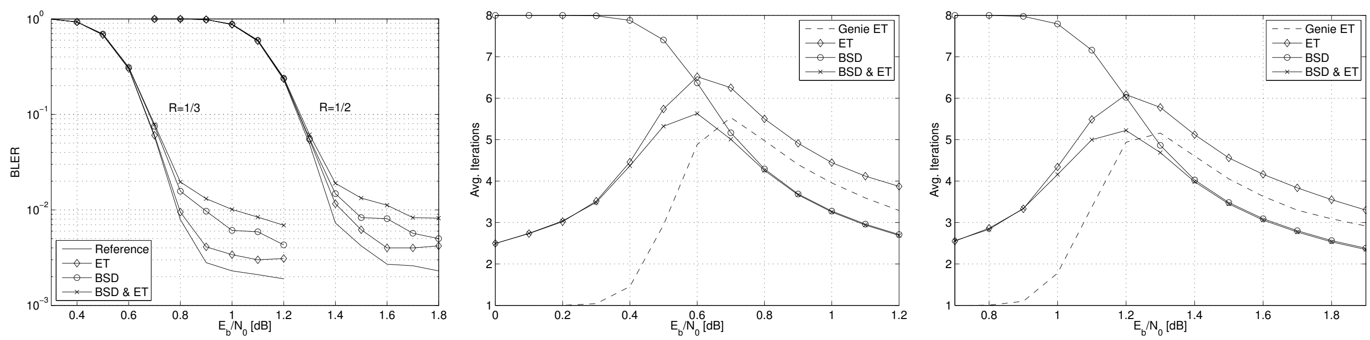
B. BLER and Average Iterations

For $R = 1/2$ and $R = 1/3$ Fig. 1 compares BLER and the average number of iterations for the following four cases:

- **Reference / Genie ET** This setting serves as a reference. The reference BLER is shown in Fig. 1(a) for the case where no ET is done and where the decoder always executes $i_{max} = 8$ iterations. The lowest possible number of iterations without BLER degradation is plotted in Figs. 1(b) and 1(c) and termed Genie ET. Here it is assumed that the decoder could perfectly detect undecodable blocks after the first full iteration (Low SNR ET scenario) and successfully decoded blocks after each full iteration (High SNR ET scenario).
- **ET** In this case only Low and High SNR ET is applied as described in Sec. IV-A.
- **BSD** This is the result of the BSD approach (Sec. IV-B).
- **BSD & ET** This is the result of complementing Low and High SNR ET with the BSD approach.

The design parameter ℓ_{min} has been chosen individually for the different code rates as $\ell_{min} = 31$ and $\ell_{min} = 25$ for $R = 1/2$ and $R = 1/3$, respectively. The choice of ℓ_{min} is a tradeoff between the required BLER performance and the resulting reduction of computational complexity. In this case, is has been selected such that there is negligible degradation for BLERs around 10%. This is a typical working point in the LTE system, but has also been shown to be a reasonable choice in general if automatic retransmissions are involved [17].

[1] A syndrome former for $R = 1/2$ encoders of the form $\mathbf{G}(D) = [G_1(D), \ G_2(D)]$ or $\mathbf{G}(D) = [1, \ G_2(D)/G_1(D)]$ can be directly seen to be $\mathbf{H}^T(D) = [G_2(D), \ G_1(D)]^T$. For higher codes rates it can be computed using the invariant factor decomposition of $\mathbf{G}(D)$ [16, Sec. 2.2 and Sec. 2.9].



(a) BLER for $R = 1/3$ and $R = 1/2$. (b) Average Iterations for $R = 1/3$. (c) Average Iterations for $R = 1/2$.

Fig. 1. Performance results for BSD with High and Low SNR ET in terms of BLER and average iterations.

As shown in Fig. 1(a) the BLER around the 10% BLER working point is practically identical in all cases. For smaller BLERs there is a visible degradation: For example at 1% BLER the combination of BSD & ET shows a loss of about 0.2dB to the reference.

Figs. 1(b) and 1(c) show the average number of iterations. In case of ET this is measured by just averaging over the iterations carried out by the decoder until termination or until i_{max} is reached. In case of BSD, each iteration is weighted by the percentage of erroneous blocks before the average is computed. For both code rates and BSD&ET a significant reduction of the number of iterations is visible: For example at the working point, the reduction against using only ET is about 1 full iteration (around 20%) without loss in BLER performance. For higher SNR the reduction is even larger than for Genie ET, because the BSD processes subblocks and the conventional ET scheme always processes the whole block.

It can also be observed, that for High SNR, the number of iterations is determined by the BSD. On the other hand in the low SNR range the average number of iterations is dominated by the ET scheme only. The reason for the latter is clearly that the constituent decoders do not converge to a common solution and thus no suitable precorrection can be found. In the medium range, both schemes complement each other such that the combination of both yields more reduction than each approach can achieve separately.

VI. Conclusions

A syndrome based MAP decoding approach for convolutional codes has been presented in this paper and its application to Turbo codes has been described. The extension with the so called BSD principle for reduction of computational complexity has been described. Further it has been demonstrated that the combination with an ET scheme can further reduce the average number of iterations, with negligible performance loss around a given typical working point.

Besides the BSD approach it should be noted that the described syndrome decoder with precorrection also yields scarce state transitions, which cannot be directly realized using the conventional MAP decoding approach [18]. Moreover, in case of true high rate codes, well-known trellis complexity reduction methods can be realized as well (cf. [19]).

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE ICC 93*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] A. Worm, H. Michel, F. Gilbert, G. Kreiselmaier, M. Thul, and N. Wehn, "Advanced implementation issues of turbo-decoders," in *2nd International Symposium on Turbo-Codes and Related Topics*, Brest, 2000.

[3] F. Gilbert, F. Kienle, and N. Wehn, "Low complexity stopping criteria for UMTS turbo-decoders," in *VTC 2003-Spring*, April 2003.

[4] F.-M. Li and A.-Y. Wu, "On the new stopping criteria of iterative turbo decoding by using decoding threshold," *IEEE Transactions on Signal Processing*, vol. 55, no. 11, pp. 5506–5516, 2007.

[5] A. H. Vinck, P. Dolezal, and Y. Kim, "Convolutional encoder state estimation," *IEEE Trans. Inf. Th.*, vol. 44, no. 4, pp. 1604–1609, 1998.

[6] J. Geldmacher, K. Hueske, and J. Götze, "An adaptive and complexity reduced decoding algorithm for convolutional codes and its application to digital broadcasting systems," in *Int. Conf. on Ultra Modern Telecommunications (ICUMT2009)*, St. Petersburg, Russia, Oct. 2009.

[7] J. Geldmacher, K. Hueske, S. Bialas, and J. Götze, "Adaptive low complexity MAP decoding for turbo equalization," in *6th Int. Symp. on Turbo Codes & Iterative Inf. Proc. (ISTC2010)*, Brest, France, 2010.

[8] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284 – 287, March 1974.

[9] J. Schalkwijk and A. Vinck, "Syndrome decoding of binary rate-1/2 convolutional codes," *IEEE Trans. Comm.*, vol. 24, no. 9, Sept. 1976.

[10] M. Tajima, "Metrics for syndrome decoding of convolutional codes," *Electr. and Comm. in Japan*, vol. 79, no. 12, pp. 22–30, 1996.

[11] M. Tajima, K. Shibata, and Z. Kawasaki, "Relation between encoder and syndrome former variables and symbol reliability estimation using a syndrome trellis," *IEEE Trans. Comm.*, vol. 51, no. 9, Sept. 2003.

[12] T. Yamada, H. Harashima, and H. Miyakawa, "A new maximum likelihood decoding of high rate convolutional codes using a trellis," *Trans. IEICE Electr. and Comm. in Japan*, vol. J66-A, no. 7, 1983.

[13] L. Lee, D. Tait, P. Farrell, and P. Leung, "Novel scarce-state-transition syndrome-former error-trellis decoding of (n, n-1) convolutional codes," in *IEEE International Symposium on Information Theory*, 1995, p. 219.

[14] T. Minowa and H. Imai, "Decoding of high-rate turbo codes using a syndrome trellis," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 1, Jun. 2001, pp. 74 –78 vol.1.

[15] T. Ngatched and F. Takawira, "Simple stopping criterion for turbo decoding," *Electr. Letters*, vol. 37, no. 22, pp. 1350 –1351, Oct 2001.

[16] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. IEEE press, 1999.

[17] P. Wu and N. Jindal, "Coding versus arq in fading channels: How reliable should the phy be?" *IEEE Trans. Comm.*, vol. 59, no. 12, Dec 2011.

[18] H. Liu, C. Jego, E. Boutillon, M. Jezequel, and J.-P. Diguet, "Scarce state transition turbo decoding based on re-encoding combined with a dummy insertion," *Electr. Letters*, vol. 45, no. 16, pp. 846 – 848, 2009.

[19] M. Tajima, K. Okino, and T. Miyagoshi, "Minimal code(error)-trellis module construction for rate-k/n convolutional codes: Extension of yamada-harashima-miyakawas construction," *IEICE Transactions on Fundamentals*, vol. E90-A, no. 11, pp. 2629–2634, 2007.