On Fault Tolerant Decoding of Turbo Codes

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

***Abstract*—Decoding of Turbo Codes requires buffer memories to store the received values and the extrinsic information that is exchanged between the constituent decoders. In this paper, the effect of unreliable buffer memories on the decoding performance is analyzed. The buffer is modeled as a discrete memoryless channel, which introduces spatially independent and uniform bit errors on the binary representation of the stored values. This leads to a strong performance degradation if a conventional Turbo decoding algorithm is employed. It is however shown that suitable modification of quantizer, index assignment, and of the transition metrics of the MAP algorithm can effectively compensate for these errors.**

I. Introduction

Iterative processing using for example Turbo Codes, Turbo equalization or iterative demapping, is a common way to improve the performance in baseband signal processing. Based on exchanging reliabilities about the transmitted information, these techniques usually require buffer memories to store these reliabilities, in addition to a commonly adopted receive buffer. It is known that the length of the processed blocks is a governing factor for the performance gain of iterative processing, which leads to large buffers, limited only by the latency requirements of the transmission system. An example is the Turbo decoding as specified in the Long Term Evolution (LTE) mobile communication standard, where the decoder has to support storage of up to 6144 reliability values in the form of Log Likelihood Ratios (LLRs) in an LLR buffer [1].

Consequently in such a system, memory not only takes a major amount of area in a hardware implementation. It also strongly impacts the power consumption of the device. Aggressive voltage scaling, i.e. scaling memory supply voltage below the required threshold, has been suggested to reduce power consumption of memory access [2]. It will, however, lead to memory failures and distortions of the stored values. Under a more general perspective, unreliable memory may also be the result of process-, time- or environment-dependent parameter variations of the involved integrated circuits [3].

In these cases, a co-design of the involved signal processing is required to compensate for the resulting memory failures. Under the assumption that the memory errors are uniformly and independently distributed, this co-design has been demonstrated for Viterbi decoding and iterative decoding based on a modified branch metric in [4] and [5], respectively. With similar preconditions, the authors have described a fault tolerant Turbo equalizer in [6]. In these works, the focus is on customizing the trellis transition probabilities under the assumption that quantized values are stored using a two's complement representation. Then a convenient analytical expression of the transition probability can be found and the effect of unreliable memory can be compensated in part.

In this paper, the problem of Turbo decoding in case of unreliable receive and LLR buffers is considered. Unreliable buffers are modeled as discrete memoryless channels (DMCs), such that the combination of transmission channel and quantizer and consecutive buffer forms a cascade of two DMCs. For the distribution of the error introduced by the buffer, we assume spatially uniformly and independently distributed bit errors on the bit representation of the quantized values. Then it can be shown that the so called memory channel, which represents the unreliable buffer memory, is symmetric and an analytical expression for the capacity can be directly given.

The memory channel results in a severe degradation of decoding performance of a conventional Turbo decoder, which is mainly caused by the irregularly distributed (i.e. non-Gaussian) values at the decoders' input. The proposed fault tolerant (FT) Turbo decoder therefore employs a modified transition metric based on the PMF of the cascaded DMCs. It is further shown that besides a proper formulation of the transition metric, quantizer properties and index assignment, i.e. the mapping of quantization values to bit vectors, also influence the decoding performance. In order to find suitable settings for received values and LLRs, the mutual information of the decoder's input and the extrinsic information transfer (EXIT) characteristics of the decoder are employed as optimization criteria, respectively. Note that, while [5] also analyzes an error-resilient Turbo decoder, the impact of quantizer, index assignment and LLR buffer is not considered in their work.

II. Problem Description

A. Turbo Coded Transmission

In the following, we assume a transmission system as shown in Fig. 1. A binary information sequence $\mathbf{u} = \{u\}$ is encoded using two parallel concatenated recursive systematic convolutional codes. The code rate of the resulting Turbo coded data is denoted by R. BPSK modulation of the coded bits v results in the equally probable modulation symbols $x = 2v - 1$, which are weighted by a gain μ before transmission. The communication channel is modeled by an additive white Gaussian noise (AWGN) channel, such that the received values $\tilde{r}$ can be written as

$$\tilde{r} = \mu x + n, \qquad \text{where } n \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

where σ^2 denotes the noise power of the channel. The channel is characterized by its signal-to-noise ratio $E_b/N_0 = 10\log_{10} \mu^2/2R\sigma^2$.

At the receiver side, a demapping is accomplished in this case (BPSK) by scaling the received values with $L_C = 2\mu/\sigma^2$. A subsequent scaling using a parameter $\gamma > 0$ is required to match the resulting LLRs to the quantizer. In this work quantization is done using an N_r bit uniform quantizer, where the reconstruction values $r \in \mathcal{Q}_r$ are defined as the set $\mathcal{Q}_r$,

$$\mathcal{Q}_r = \{-2^{N_r-1} + k : 0 \leq k < 2^{N_r}\}. \tag{2}$$

The probability mass function (PMF) $P(r|x)$ of the quantized values r conditioned on the original symbols x can be

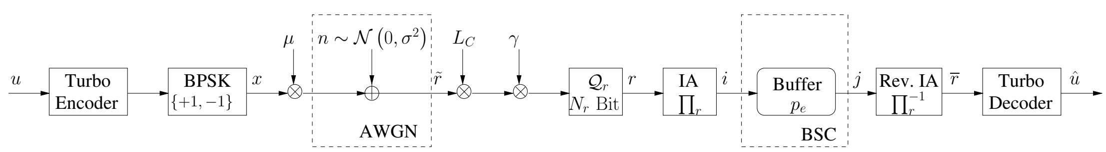


Fig. 1. Transmission of Turbo coded data over an AWGN channel with cascaded unreliable receive buffer.

calculated as

$$P(r|x) \sim \begin{cases} \int_{-\infty}^{r+0.5} e^{-\left((\tilde{r}-\mu_r x)^2/(2\sigma_r^2)\right)} d\tilde{r}, & r = -2^{N_r-1} \\ \int_{r-0.5}^{\infty} e^{-\left((\tilde{r}-\mu_r x)^2/(2\sigma_r^2)\right)} d\tilde{r}, & r = 2^{N_r-1}-1 \\ \int_{r-0.5}^{r+0.5} e^{-\left((\tilde{r}-\mu_r x)^2/(2\sigma_r^2)\right)} d\tilde{r}, & \text{else}, \end{cases} \tag{3}$$

where $\mu_r = L_C\gamma\mu$ and $\sigma_r^2 = L_C^2\gamma^2\sigma^2$ are gain and variance of the bimodal Gaussian distribution at quantizer input.

After quantization, indices $i, 0 \leq i < 2^{N_r}$ are assigned to the quantized values r. The index assignment (IA) is represented by $\Pi_r(\cdot)$, which defines the one-to-one mapping between the elements of $\mathcal{Q}_r$ and the indices i. The binary N_r-bit representation of $i = \Pi_r(r)$ is written to the buffer.

Given the assumption that the buffer is unreliable and that bit errors are spatially independently and uniformly distributed, it can be modeled as a binary symmetric channel with bit error probability p_e. Then the conditional PMF of reading an index j if i has been written is given as

$$P(j|i) = p_e^{d_H(i,j)}(1-p_e)^{N_r-d_H(i,j)}, \tag{4}$$

where $d_H(i,j)$ denotes the Hamming distance between i and j. Finally, the reverse index assignment $\overline{r} = \Pi_r^{-1}(j)$ re-assigns quantization values $\overline{r}$. The consecutive Turbo decoder then delivers an estimate of u based on $\overline{r}$, taking into account the scaling parameters L_C and γ.

B. Capacity of Memory Channel

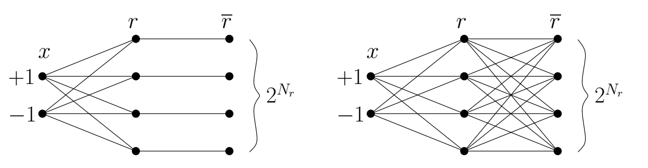
The combination of communication channel with successive quantizer and memory channel can be seen as a cascade of two DMCs, as shown in Fig. 2. For perfectly operating buffer memory ($p_e = 0$), the restored values $\overline{r}$ match the written values r (Fig. 2(a)). However, in case of unreliable buffer memory, ($p_e > 0$), r and $\overline{r}$ do not necessarily match, as shown in Fig. 2(b). In this case the buffer causes an additional distortion.

This distortion depends on the bit error probability p_e and may be characterized by analyzing the capacity C_m of the memory channel. The capacity is given as

$$C_m = \max_{\mathbf{P}(r)} I(R;\overline{R}), \tag{5}$$

where $\mathbf{P}(r)$ is a distribution of input values, and $I(R;\overline{R})$ is the average mutual information (MI) between the discrete random variables R and $\overline{R}$ representing r and $\overline{r}$, respectively. Taking into account that the $2^{N_r} \times 2^{N_r}$ transmission matrix $\mathbf{P}(\overline{r}|r) = [P(\overline{r}|r)]$ can be written as

$$\mathbf{P}(\overline{r}|r) = \mathbf{\Pi}\mathbf{P}(j|i)\mathbf{\Pi}^T \text{ with } \mathbf{P}(j|i) = [P(j|i)], \tag{6}$$



(a) Perfect buffer ($p_e = 0$). (b) Unreliable buffer ($p_e \neq 0$).

Fig. 2. Transmission channel and buffer memory as a cascade of two DMCs.

where $\mathbf{\Pi}$ is a permutation matrix reflecting the index assignment, it can be seen that the memory channel is symmetric. In this case the MI is maximized for uniform $\mathbf{P}(r)$ [7] and the capacity can be directly expressed as

$$\begin{aligned} C_m &= H(R) - H(R|\overline{R}) \\ &= N_r + \sum_{n=0}^{N_r} \binom{N_r}{n} p_e^n (1-p_e)^{N_r-n} \text{ ld } p_e^n (1-p_e)^{N_r-n}. \end{aligned}$$

Fig. 3 illustrates C_m as a function of the error probability p_e for different N_r. For example it can be seen that for $p_e = 0.1$ the relative capacity loss is around 50% for all N_r.

Note that uniform $\mathbf{P}(r)$ not only maximizes $I(R;\overline{R})$, due to symmetry of the channel, but also $I(\tilde{R};R)$ [8]. In terms of an optimal EXIT chart entry point we are however interested in optimizing $I(X;\overline{R})$ rather than $I(\tilde{R};R)$, as will be discussed in Sec. III-B.

III. Fault Tolerant Decoding

The objective is to make the Turbo decoder resilient against the additional errors introduced by the memory channel. This is achieved by

- modifying the Turbo decoder itself, such that the MAP algorithm is adapted to the PMF $P(\overline{r}|x)$ of the cascade of Gaussian transmission channel and memory channel,
- analyzing the impact of scaling factor γ and IA of received values and IA of extrinsic LLRs on the EXIT characteristics of the decoding process.

A. MAP Algorithm

A key observation is that the probability distribution of the memory output $\overline{r}$ conditioned on x is no longer Gaussian shaped if $p_e > 0$: Depending on the employed IA, $P(\overline{r}|x)$ results from superposition of 2^{N_r} PMFs corresponding to the 2^{N_r} different possible error patterns. As an example, possible resulting PMFs at memory input and output are shown in Fig. 4. The PMFs $P(\overline{r}|x)$ are shown for two different IAs, namely natural binary coding (NBC, $\mathbf{\Pi} = \mathbf{I}$) and a randomly selected IA (RND, $\mathbf{\Pi}$ is a random permutation of $\mathbf{I}$). Clearly $P(\overline{r}|x)$ is no longer Gaussian shaped, and in case of the RND IA even non-symmetric.

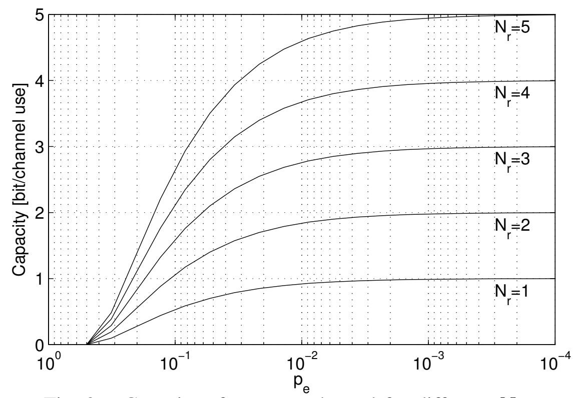


Fig. 3. Capacity of memory channel for different N_r.

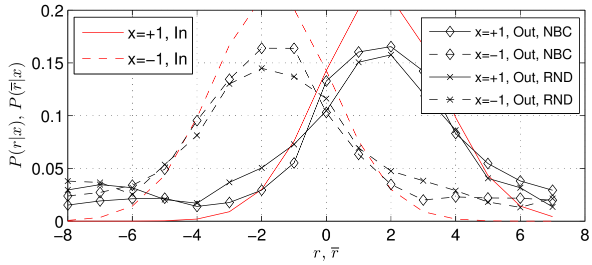


Fig. 4. PMFs at memory input ($P(r|x)$, red) and output ($P(\overline{r}|x)$, black) for $N_r = 4$, $\mu = 1$, $E_b/N_0 = 1dB$, $\gamma = 1$, $p_e = 0.1$.

If Turbo decoding is implemented based on the MAP algorithm, then this observation suggests to implement the MAP algorithm based on the transition probability of the cascade of communication channel and memory channel, and not on the conventional metric, which is based on the Gaussian assumption. Given the model from Sec. II, the required conditional PMFs for $x = \pm 1$ can be computed following

$$\mathbf{P}(\overline{r}|x) = \mathbf{\Pi}\mathbf{P}(i|j)\mathbf{\Pi}^T\mathbf{P}(r|x), \quad (7)$$

where $\mathbf{P}(\overline{r}|x) = [P(\overline{r}|x)]$ is the resulting $2^{N_r} \times 2$ transmission matrix of the two cascaded DMCs.

Besides the receive buffer, considered until now, the Turbo decoder also requires a buffer to store the extrinsic LLRs $\tilde{\lambda}$ exchanged between the constituent decoders during iterations. An unreliable LLR buffer is modeled similarly to the receive buffer, as shown in Fig. 5: The input of the quantizer, i.e. a constituent decoder's output, conditioned on the systematic part x_s of x is modeled as consistently Gaussian distributed with variance σ_λ^2.

The quantizer uses $N_\lambda = d_\lambda + f_\lambda$ bits and assigns reconstruction values λ from the set

$$\mathcal{Q}_\lambda = \{-2^{d_\lambda - 1} + k2^{-f_\lambda} : 0 \le k < 2^{N_\lambda}\}. \quad (8)$$

Based on the IA Π_λ, the binary representation of the index is written to the buffer, and the restored value $\overline{\lambda}$ may differ from the original value λ if $p_e > 0$. The conditional PMFs $P(\overline{\lambda}|x_s)$ can be derived in the same way as for the received values.

Given this model, a suitable transition metric at time instant t can be computed as

$$\log P(\overline{r}_s^{(t)}|x_s) + \log P(\overline{r}_p^{(t)}|x_p) + \log P(\overline{\lambda}^{(t)}|x_s), \quad (9)$$

where half rate encoders are assumed, and $\overline{r}_s^{(t)}$ and $\overline{r}_p^{(t)}$ are systematic and parity part of the received values, respectively, and x_s and x_p denote systematic and parity parts of the current trellis transition. The metric computation can be favorably implemented using look-up tables (LUTs) for $P(\overline{r}|x)$ and $P(\overline{\lambda}|x)$, where the LUTs can be directly addressed using the indices of the restored values. Precomputing $P(\overline{r}|x)$ requires knowledge of the noise power σ^2 and error probability p_e, which we assume is known in the decoder. Similarly, precomputing $P(\overline{\lambda}|x)$ requires knowledge of σ_λ^2 (and p_e), which has to be estimated while writing λ into the buffer.

The MAP algorithms based on (9) is referred to as fault tolerant MAP (FT MAP) in the following.

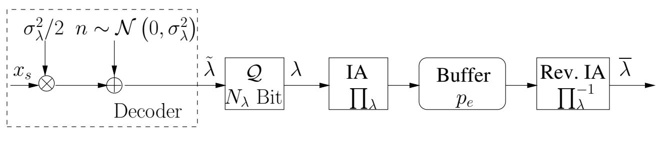


Fig. 5. Extrinsic LLRs modeled as consistent Gaussian distributed values with quantizer and consecutive unreliable buffer memory.

Note that an alternative way of implementing the MAP decoder for two's complement based quantization using multiple shuffled versions of the original PMF has been described in [6]. In this case precomputing of a PMF LUT is not required.

B. EXIT Chart based Optimization

The FT decoder is supposed to exploit "available information" in its input values $\overline{r}$ as well as possible. In terms of Turbo decoding, the EXIT chart [9] is a convenient tool to visualize the decoding characteristics: Firstly, convergence behaviour is improved if the entry point into the chart is higher, i.e. in our case if the "available information" measured as the MI $I(X;\overline{R})$ is as large as possible. This can be influenced by adjusting γ and Π_r. Secondly, the EXIT tunnel, i.e. the distance between both decoder curves, should be as wide as possible, and the point, where the generated extrinsic MI of a constituent decoder matches its *a priori* MI, should be shifted as high as possible. This can be accomplished by adjusting Π_λ.

1) Optimization of $I(X;\overline{R})$: The MI $I(X;\overline{R})$ is a function of the PMF $P(\overline{r}|x)$,

$$I(X;\overline{R}) = \frac{1}{2}\sum_{x=\pm 1}\sum_{\overline{r}} P(\overline{r}|x) \log \frac{P(\overline{r}|x)}{P(\overline{r})}, \quad (10)$$

and $P(\overline{r}|x)$ in turn is influenced by the quantizer properties, which is the scaling parameter γ in this work, and by the IA Π_r. Thus, the MI for a given pair (γ, Π_r) is denoted by $I_{(\gamma,\Pi_r)}(X;\overline{R})$. Consequently it is reasonable to assume that for a given E_b/N_0, there is a combination (Π_r^*, γ^*) that maximizes $I_{(\gamma,\Pi_r)}(X;\overline{R})$:

$$(\Pi_r^*, \gamma^*) = \arg\max_{\Pi_r,\gamma} I_{(\gamma,\Pi_r)}(X;\overline{R}) \quad (11)$$

In this work, (11) is optimized by first solving

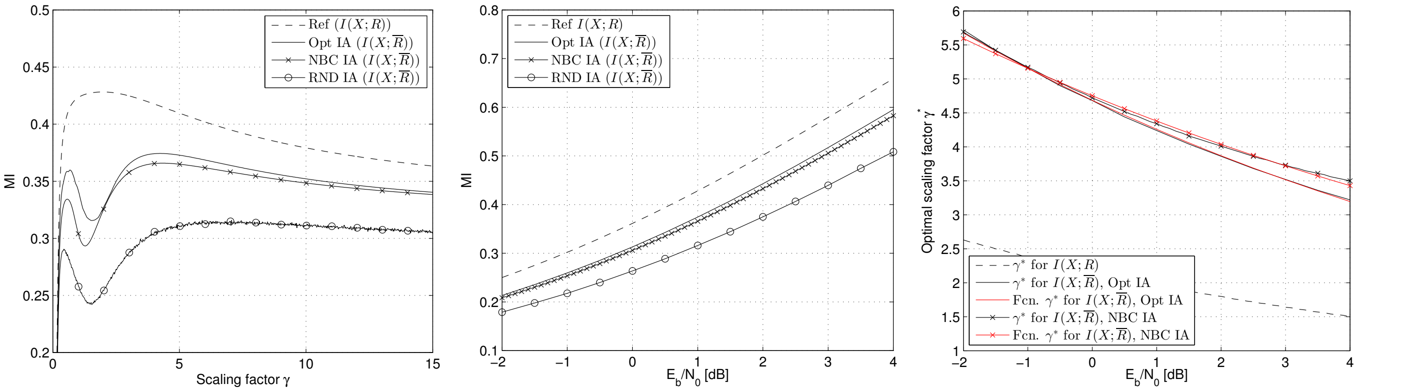
$$\Pi_r^*(\gamma) = \arg\max_{\Pi_r} I_{(\gamma,\Pi_r)}(X;\overline{R}), \quad (12)$$

i.e. finding optimized $\Pi_r^*(\gamma)$ for reasonable values of γ, and then by selecting γ^* as

$$\gamma^* = \arg\max_{\gamma} I_{(\gamma,\Pi_r^*(\gamma))}(X;\overline{R}) \text{ and } \Pi_r^* = \Pi_r^*(\gamma^*). \quad (13)$$

The problem (12) of finding an IA subject to some optimization criterion is a combinatorial optimization problem, which for example also appears in source-channel-coding using MSE [10] or MinMax criteria [11]. The number of possible solutions is determined by the number of bits N_r and given by $(2^{N_r}!)$. Although the search space can be reduced a little, because there are 2^{N_r} equivalent solutions, an exhaustive search is infeasible for $N_r > 3$. Therefore, following [12], simulated annealing (SA) is adopted in this work to optimize (12). While SA as a metaheuristic cannot guarantee finding a globally optimal solution, our experiments showed satisfactory results for several configurations.

For a fixed $E_b/N_0 = 1dB$, Fig. 6(a) shows the MI before memory channel $I(X;R)$ as reference, and compares the MI $I(X;\overline{R})$ for optimized IA ("Opt IA"), fixed NBC IA, and randomly selected IA ("RND IA") as functions of γ. It can be observed that in all cases there is a γ^* that maximizes the MI, and that NBC IA is only slightly worse than optimized IA, while randomly selected IA shows significantly lower MI. Similarly, Fig. 6(b) shows the best MI for the three different IAs as function of E_b/N_0. Again, optimized IA performs



(a) MI as function of γ at $E_b/N_0 = 1dB$.

(b) Best MI as function of E_b/N_0.

(c) Required γ^* to optimize MI.

Fig. 6. Effect of scaling factor γ and IA on MI ($N_r = 4$, $p_e = 0.05$, $R = 1/3$).

little better than NBC, whereas random IA is significantly worse. The required optimal scaling factor γ^* for optimized IA and NBC IA is shown in Fig. 6(c), and compared to the scaling factor required to optimize $I(X;R)$: A clear difference between the first and the latter is visible, where optimization subject to $I(X;\overline{R})$ requires a larger γ for the given system.

The results indicate that the IA has considerable impact on the MI at memory channel output: Selecting an arbitrary, random IA may result in a performance decrease, while NBC IA is close to optimized IA. Practical reasons suggest the use of NBC or an equivalent IA, like two's complement representation, over using optimized IAs, because otherwise optimized IAs would have to be pre-computed and stored for each possible channel condition. The scaling factor on the other hand should be adjusted dynamically depending on current E_b/N_0. For the considered system, γ^* can readily be computed for the current E_b/N_0 using a function of the form

$$\log \gamma^* = m \cdot E_b/N_0 + c, \tag{14}$$

where the parameters m, c may be found by least squares fitting. Fig. 6(c) illustrates that the fitted functions ("Fcn λ^*") provide a good approximation for the cases of NBC and optimized IA.

2) Optimization of EXIT characteristic: As described in the previous paragraph, the impact of Π_r and γ can be qualified in terms of the MI $I(X;\overline{R})$ for a given E_b/N_0. In the decoder, an optimal scaling factor (and possibly an optimal IA, if further improvement over NBC IA is desired) can then be selected given an estimate of E_b/N_0. The influence of the IA Π_λ, which is the IA used for the LLRs exchanged between the constituent decoders, should however not be analyzed for a fixed noise power σ_λ^2: Keeping the decoder implementation in mind, it is not reasonable to find optimized IAs for each possible σ_λ^2, because there is a wide range of possible σ_λ^2 depending on the convergence of the current decoding process. This would require precomputing and selecting optimized IAs for a significant amount of possible values σ_λ^2.

As an alternative we propose to select an IA, which is fixed for all σ_λ^2, based on its impact on the decoding process. This impact is reflected by the shape of the function that relates *a priori* information supplied to the decoder to the extrinsic information it generates. We define the *a priori* MI of a decoder as $I_A := I(X;\Lambda_A)$, and the resulting MI of the extrinsic values as $I_E(I_A,\Pi_\lambda) := I(X;\Lambda_E)$, where the remaining system parameters are kept fixed:

- I_A is the MI of the quantized *a priori* values λ_A before writing them into the buffer. If $p_e > 0$, these values are distorted by the memory channel, such that the input $\overline{\lambda}_A$ of the constituent decoder actually has a decreased MI $0 \leq I(X;\overline{\Lambda}_A) < I(X;\Lambda_A) \leq 1$.
- $I_E(I_A,\Pi_\lambda)$ is the MI of the output of a constituent decoder, given the *a priori* MI I_A.

Based on these definitions the optimization problem

$$\Pi_\lambda^* = \arg\max_{\Pi_\lambda}\min_{I_A}\left\{I_E(I_A,\Pi_\lambda) - I_A | I_A \leq \Delta\right\}, \tag{15}$$

where $0 < \Delta \leq 1$, formalizes the following performance criterion: Find an IA Π_λ^* that maximizes the minimum increase in MI $I_E(I_A,\Pi_\lambda)$ in the extrinsic values, compared to the MI I_A of the supplied *a priori* values, for each $I_A \leq \Delta$. In other words, optimizing (15) yields an EXIT chart, where the minimum distance of the two decoder curves is maximized in the range $0 \ldots \Delta$. We use SA to optimize (15).

The parameter Δ controls the resulting decoding behaviour: A large value, for example $\Delta = 1$, may shift the point $I_E(I_A,\Pi_\lambda) = I_A$ as far as possible towards $(1,1)$. This may lead to a steeper BER curve, but higher decoding threshold. On the other hand, a smaller Δ, for example $\Delta = 0.9$, leads to a wider tunnel, meaning earlier convergence, but disregards the upper part in the EXIT chart, resulting in a less steep BER curve.

IV. Simulation Results

In this section simulation results for EXIT charts and BER are shown. The BER simulation follows the system as shown in Fig. 1. All results are based on a binary rate $R = 1/3$ Turbo code using parallel concatenation of two UMTS/LTE compatible [1] recursive systematic encoders $\mathbf{G}(D) = \left[1, \frac{1+D+D^3}{1+D^2+D^3}\right]$. BPSK mapped symbols with gain factor $\mu = 1$ are then transmitted over an AWGN channel. The scaling factor γ^* is selected depending on the current E_b/N_0 using (14), with parameters (m,c) as $(-0.09, 0.77)$, $(-0.09, 1.16)$ and $(-0.08, 1.56)$ for $p_e = 0$, $p_e = 0.01$ and $p_e = 0.05$, respectively. In all cases quantization of the received values is done using $N_r = 4$ bit with two's complement representation. Two's complement is equivalent to NBC IA, which according to the results from Sec. III-B is only slightly worse than optimized IA in terms of the resulting MI $I(X;\overline{R})$. For the LLRs λ a $N_\lambda = 7$ bit quantizer is used, where $d_\lambda = 5$ and $f_\lambda = 2$. The Turbo decoder uses the FT MAP as described in Sec. III-A based on the

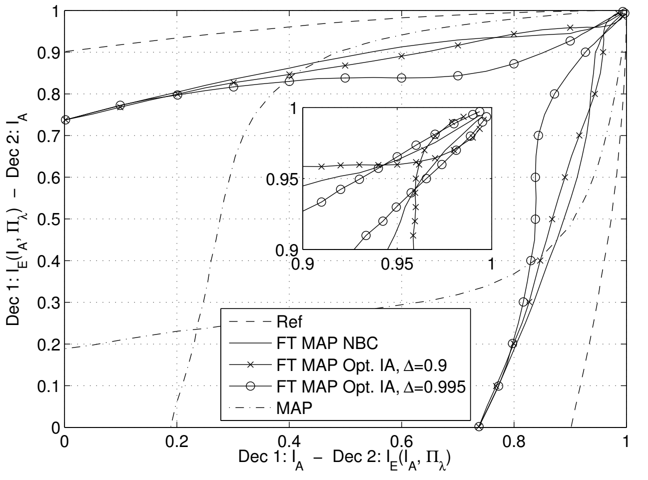


Fig. 7. EXIT chart for conventional MAP and FT MAP with different IA ($E_b/N_0 = 4dB$, $p_e = 0.05$).

LogMAP algorithm with 8 iterations. The required LUTs are precomputed assuming perfect knowledge of the parameters p_e, σ^2 and σ_λ^2. Figs. 7 and 8 compare the EXIT charts and BER of a decoder based on the FT MAP to a conventional MAP. For the FT MAP, three cases are considered: NBC IA, and optimized IA with $\Delta = 0.9$ and $\Delta = 0.995$. Additionally, results for a decoder with error-free buffer memory are shown as a reference ("Ref").

Looking at Fig. 7 it is obvious that the EXIT chart of the conventional MAP decoder in case of erroneous buffer memory ($p_e = 0.05$) is strongly degraded: The early crossing point of both curves indicates that the decoding process will not converge at the given SNR. A significantly higher SNR would be required for convergence. On the other hand, the curves of the FT MAP based decoder using NBC IA do not intersect at all, such that an improved BER can be expected.

The effect of using an IA based on optimizing (15) can also be observed clearly: For $\Delta = 0.9$, the region $I_A > 0.9$ is disregarded during optimization, which leads to an intersection at about 0.96, but also to a wider "bottleneck", which is supposed to increase robustness of the decoder. On the other hand, for $\Delta = 0.995$ the "bottleneck" is smaller in the region $I_A < 0.94$, but improved in the very upper part. This leads to better performance for higher SNR and possibly an improved error floor.

The BER curves in Fig. 8 confirm these conclusions. Considering for example a BER working point of 10^{-4} and $p_e = 0.05$ (blue curves), the Turbo decoder based on the conventional MAP algorithm shows a performance loss of about 7dB compared to the reference. The FT MAP based decoder significantly improves the BER performance: A performance gain of 3dB is reached by using NBC IA. The optimized IA with $\Delta = 0.995$ is improved by another 0.4dB, while the optimized IA with $\Delta = 0.9$ is degraded by about 1dB. However, if the latter IA is used, a BER of 10^{-2} is already reached at 2.4dB, while NBC IA and the IA optimized for $\Delta = 0.995$ require 2.8dB and 3.6db, respectively. Finally, it can be observed that the FT MAP using NBC IA exhibits an error floor, while there is a significantly lower floor in the considered BER range for the optimized IAs. The same holds for a smaller bit error probability of $p_e = 0.01$ (red curves).

It can be concluded that selecting an optimized, fixed IA Π_λ with a suitable Δ may outperform NBC IA, depending on the BER requirements, at little implementation cost.

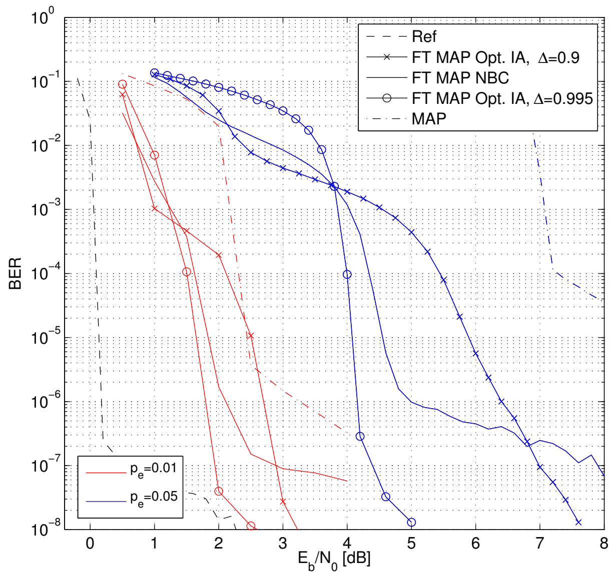


Fig. 8. Comparison of BER performance after 8 iterations for FT MAP and conventional MAP for $p_e = 0.01$ and $p_e = 0.05$, and reference ($p_e = 0$).

V. Conclusions

The effect of unreliable buffer memory on Turbo decoding has been discussed: While a decoder based on the conventional MAP algorithm exhibits a strong performance degradation, the proposed FT MAP can significantly reduce these additional distortions, given a proper selection of quantizer and IA. For the latter it has been observed that NBC or an equivalent IA delivers good results for the considered system. But especially in case of the LLR buffer, the IA can be further optimized using the proposed EXIT chart based criterion to yield a steeper BER curve or earlier decoding threshold.

Further analysis is required for the robustness of the FT MAP regarding to the parameters σ^2, p_e, and σ_λ^2. More generally, different quantizer designs strongly influence the performance and the degrees of freedom of the involved optimization problems and useful upper bounds for these problems are still unknown. Both problems might be considered for future work.

References

[1] *Evolved Universal Terrestrial Radio Access (E-UTRA) - 3GPP TS 36.213 V8.8.0*, 3rd Generation Partnership Project Std., 2009.

[2] A. K. Djahromi, A. M. Eltawil, F. J. Kurdahi, and R. Kanj, "Cross layer error exploitation for aggressive voltage scaling," in *8th Int. Symp. on Quality Electronic Design (ISQED '07)*, Mar 2007, pp. 192–197.

[3] S. Ghosh and K. Roy, "Parameter variation tolerance and error resiliency: New design paradigm for the nanoscale era," *Proceedings of the IEEE*, vol. 98, no. 10, pp. 1718–1751, 2010.

[4] A. Hussien, M. Khairy, A. Khajeh, A. Eltawil, and F. Kurdahi, "Combined channel and hardware noise resilient viterbi decoder," in *Asilomar Conf. on SS&C*, Pacific Grove, CA, Nov 2010.

[5] ——, "A class of low power error compensation iterative decoders," in *IEEE GLOBECOM*, Dec 2011.

[6] J. Geldmacher, K. Hueske, and J. Götze, "Turbo equalization for receivers with unreliable buffer memory," in *IEEE Vehicular Technology Conference (VTC Fall)*, Sept 2011.

[7] T. M. Cover and J. A. Thomas, *Elements of Inf. Theory*. Wiley, 2006.

[8] C. Novak, P. Fertl, and G. Matz, "Quantization for soft-output demodulators in bit-interleaved coded modulation systems," in *IEEE ISIT*, July 2009, pp. 1070–1074.

[9] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. on Communications*, vol. 49, no. 10, pp. 1727–1737, Oct 2001.

[10] K. Zeger and A. Gersho, "Pseudo-gray coding," *IEEE Transactions on Communications*, vol. 38, no. 12, pp. 2147 –2158, Dec. 1990.

[11] L. Potter and D.-M. Chiang, "Minimax nonredundant channel coding," *IEEE Trans. on Communications*, vol. 43, no. 234, pp. 804–811, 1995.

[12] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. on Information Theory*, vol. 36, no. 4, pp. 799–809, Jul. 1990.