Turbo-Decoding Without SNR Estimation

Alexander Worm, *Student Member, IEEE*, Peter Hoeher, *Senior Member, IEEE*, and Norbert Wehn, *Member, IEEE*

***Abstract*—Theoretically, it is necessary to estimate the SNR when using a MAP or Log-MAP constituent decoder. The effect of an SNR mismatch on the bit error rate performance of Turbo-codes and the design of good variance estimators have been addressed by several authors.**

In this letter, we study the SNR sensitivity of Turbo-decoding with Log-MAP and Max-Log-MAP constituent decoders, respectively, for AWGN and Rayleigh fading channels. Our theoretical and simulation results indicate that an estimation of the SNR is not necessary from a practical point of view. Our setup is aligned with decoder implementation aspects of future mobile communication systems.

***Index Terms*—MAP decoding, mobile communication, Turbo-codes.**

I. Introduction

TURBO-CODES [1] have been paid considerable attention since their proposal in 1993. One aspect that gained attention quite recently is the dependency of the decoding algorithm on accurate estimates of the channel characteristics, such as the fading amplitude and phase and the signal-to-noise ratio (SNR) of the channel.

The effects of an SNR mismatch on the bit error rate (BER) performance of Turbo-codes have been investigated by Jordan and Nichols for additive white Gaussian noise (AWGN) and for binary symmetric channels [2]. It was found that errors of up to several dB were tolerated before the onset of significant degradation in decoding performance. Reed and Asenstorfer also included fully interleaved Rayleigh fading channels [3]. However, as with Summers and Wilson [4], they were mainly interested in novel variance estimators. Valenti and Woerner extended the discussion toward correlated Rayleigh and Rician fading, also proposing a novel channel estimator [5]. Frenger included the error variance of the channel estimate in the decoder metric derivation, but assumed the channel SNR perfectly known [6].

Our research on the effects of an SNR mismatch in Turbo-decoding was driven by an exploration of implementation aspects for mobile communications, as outlined in [7]. We therefore investigated conventional Turbo-decoding of parallel concatenated convolutional codes on the basis of the following simulation conditions: We used two memory $m = 2$ recursive systematic convolutional constituent codes with generators $5/7$ [8]. By periodically puncturing the parity bit, a rate $R = 1/2$ code is obtained [1]. For the purpose of reproducibility, we applied a uniform random interleaver of block length $L_{\text{block}} = 600$; new permutations were generated for each block. Zero tailing was emulated by transmitting the all-zeros sequence. We used a conventional symmetric decoder structure as described in [9]. Both AWGN and fully interleaved frequency-flat Rayleigh fading channels were assumed.

The remainder of this letter is structured as follows: Section II highlights the differences of Turbo-decoding with Log-MAP and Max-Log-MAP constituent decoders [10] with respect to the SNR sensitivity. These theoretical results are supported in Section III by means of simulation. We finally draw conclusions in Section IV.

Manuscript received October 12, 1999. The associate editor coordinating the review of this letter and approving it for publication was Prof. E. Biglieri.

A. Worm and N. Wehn are with the Institute of Microelectronic Systems, University of Kaiserslautern, Kaiserslautern, Germany (e-mail: {worm@eit.uni-kl.de; wehn@eit.uni-kl.de).

P. Hoeher is with the Information and Coding Theory Laboratory, University of Kiel, Kiel, Germany (e-mail: ph@techfak.uni-kiel.de).

Publisher Item Identifier S 1089-7798(00)05697-0.

II. Theoretical Background

For implementation purposes, the MAP constituent decoders of the conventional Turbo-decoder proposed in [1] should be replaced by the Log-MAP decoder or sub-optimal variants, such as the Max-Log-MAP decoder [9], [10]. The Log-MAP decoder is an optimal constituent decoder, whereas the sub-optimal Max-Log-MAP decoder is less complex. For the further discussion, we will use the notation of [10].

A. Log-MAP Decoder

The Log-MAP decoder computes the log-likelihood ratio

$$\begin{aligned}\Lambda(d_k) = &\max_{(S_k,\, S_{k-1})}{}^{*} \{\overline{\gamma}_1[(y_k^s,\, y_k^p),\, S_{k-1},\, S_k] + \overline{\alpha}_{k-1}(S_{k-1}) \\ &+ \overline{\beta}_k(S_k)\} - \max_{(S_k,\, S_{k-1})}{}^{*} \{\overline{\gamma}_0[(y_k^s,\, y_k^p),\, S_{k-1},\, S_k] \\ &+ \overline{\alpha}_{k-1}(S_{k-1}) + \overline{\beta}_k(S_k)\}\end{aligned} \tag{1}$$

where

$$\begin{aligned}\overline{\gamma}_i[(y_k^s,\, y_k^p), S_{k-1}, S_k] = &\frac{2E_s}{N_0} y_k^s x_k^s(i) + \frac{2E_s}{N_0} y_k^p x_k^p(i, S_k, S_{k-1}) \\ &+ \ln\, \Pr\{S_k | S_{k-1}\}\end{aligned} \tag{2}$$

are the branch metrics $(i = 0,\, 1)$,

$$\begin{aligned}&\ln\, \Pr\{S_k | S_{k-1}\} \\ &\quad = \begin{cases} L(d_k) - \ln\left(1 + e^{L(d_k)}\right), & q(d_k = 1 | S_k,\, S_{k-1}) = 1 \\ -\ln\left(1 + e^{L(d_k)}\right), & q(d_k = 0 | S_k,\, S_{k-1}) = 1 \end{cases}\end{aligned} \tag{3}$$

is the *a priori* information, and

$$\overline{\alpha}_k(S_k) = \max_{(S_{k-1}, i)}{}^{*} \{\overline{\gamma}_i[(y_k^s, y_k^p),\, S_{k-1}, S_k] + \overline{\alpha}_{k-1}(S_{k-1})\} \tag{4}$$

$$\overline{\beta}_k(S_k) = \max_{(S_{k+1}, i)}{}^{*} \{\overline{\gamma}_i[(y_{k+1}^s, y_{k+1}^p), S_k,\, S_{k+1}] + \overline{\beta}_{k+1}(S_{k+1})\} \tag{5}$$

are the accumulated path metrics. We define

$$\max{}^{*}(\delta_1,\, \delta_2) = \max(\delta_1,\, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}) \tag{6}$$

hence

$$\ln(1+e^{L(d_k)}) = \max{}^*(0,\, L(d_k)). \tag{7}$$

Iterative Turbo-decoding relies on a superposition of independent log-likelihood ratios:

$$\begin{aligned}\Lambda(d_k) = {} & \max_{(S_k,\, S_{k-1})}{}^* \{\overline{\gamma}_1{}'[y_k^p,\, S_{k-1},\, S_k] + \overline{\alpha}_{k-1}(S_{k-1}) \\ & + \overline{\beta}_k(S_k)\} - \max_{(S_k,\, S_{k-1})}{}^* \{\overline{\gamma}_0{}'[y_k^p,\, S_{k-1},\, S_k] \\ & + \overline{\alpha}_{k-1}(S_{k-1}) + \overline{\beta}_k(S_k)\} - \frac{4E_s y_k^s}{N_0} + L(d_k).\end{aligned} \tag{8}$$

The first two terms comprise the so-called extrinsic component, the third term is the systematic information, and the last term is the *a priori* component, which is the extrinsic component of the previous stage.

The Log-MAP decoder requires knowledge of the SNR (E_s/N_0) for two reasons:

1. The SNR is required for the metric calculation, see (2).
2. The SNR is required to properly scale the information passed between the constituent decoders, see (8).

Due to the nonlinear $\max^*$-operation (6), it is impossible to factor E_s/N_0 out.

B. Max-Log-MAP Decoder

The Max-Log-MAP decoder, as described in [10], is deduced from the Log-MAP decoder by substituting each $\max^*$-operation by a $\max$-operation ("high-SNR-rule"). This algorithm, when employed in Turbo-decoding, does not require knowledge of the SNR:

1. Now, the *a priori* information in (3) becomes either $-L(d_k)$, 0 or $L(d_k)$. Thus, if $L(d_k)$ is assumed to be proportional to E_s/N_0, then the metric will also be proportional to E_s/N_0, and the SNR can be factored out.
2. With the metric assumed to be proportional to E_s/N_0 and with $\max^*$ replaced by $\max$, the SNR can be factored out in (1), (4) and (5). Now, the soft outputs are scaled with the SNR, but the hard decisions become SNR independent. However, this requires proper initialization of the metric accumulations.
3. As $L(d_k)$ is the extrinsic information of the previous stage in the context of Turbo-decoding, see (8), the assumption that $L(d_k) \propto E_s/N_0$ can be easily proved for reasonable initialization $(L(d_k) = 0\, \forall_k)$.

Sometimes, the Max-Log-MAP decoder is derived without substituting the $\max^*$-operation in (7) to calculate the extrinsic information. Then the metric (2) remains SNR dependent. However, simulations indicate that this Max-Log-MAP decoder version is nevertheless virtually independent of the assumed SNR. This may be due to the path decisions [10] taken during the recursions and the Λ-calculation.

C. Conclusion

Theoretically, decoding with the Log-MAP decoder requires SNR estimates. We therefore expect the Log-MAP decoder to be sensitive wrt. an SNR mismatch. The consequences have to be investigated by simulation. In contrast, Turbo-decoding with the Max-Log-MAP decoder (when properly defined) is SNR independent.

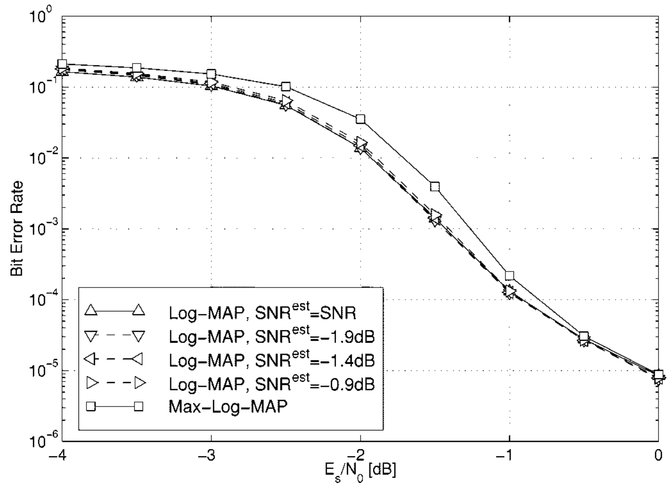

Fig. 1. Fixed estimated SNR, AWGN channel. $R = 1/2$, $m = 2$, $\mathrm{L}_{\mathrm{block}} = 600$, random interleaver, 10 iterations.

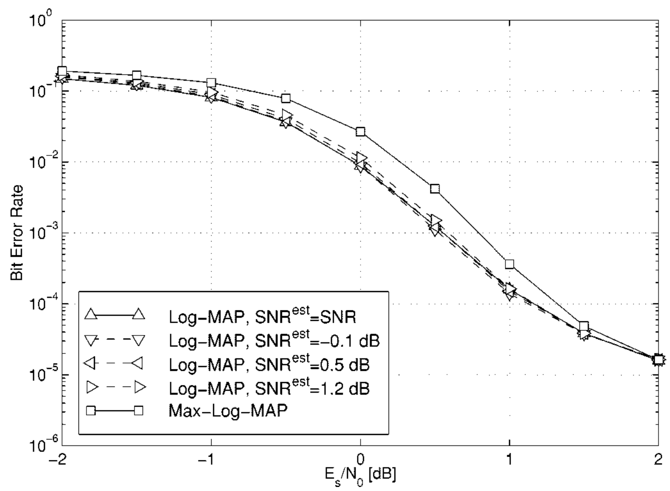

Fig. 2. Fixed estimated SNR, Rayleigh fading channel. $R = 1/2$, $m = 2$, $\mathrm{L}_{\mathrm{block}} = 600$, random interleaver, 10 iterations.

III. Simulation Results

Two types of simulation have been conducted: The first type assumes fixed "estimated" SNR, the second type investigates offsets. Perfect channel state information is assumed.

A. Fixed Estimated SNR

Figs. 1 and 2 show the BER versus SNR for AWGN and for Rayleigh fading channels, respectively. For 10 iterations, each figure features one curve for the Log-MAP decoder with perfectly known SNR and three curves for the Log-MAP decoder with fixed SNR's according to three *design points* with BER's $\in \{10^{-2},\, 10^{-3},\, 10^{-4}\}$. Note that we have to plot only one curve for the Max-Log-MAP decoder.

As already reported for the AWGN channel in [9], the differences are surprisingly small. Only the Max-Log-MAP decoder deteriorates significantly for small SNR. For Rayleigh fading, the differences are still small.

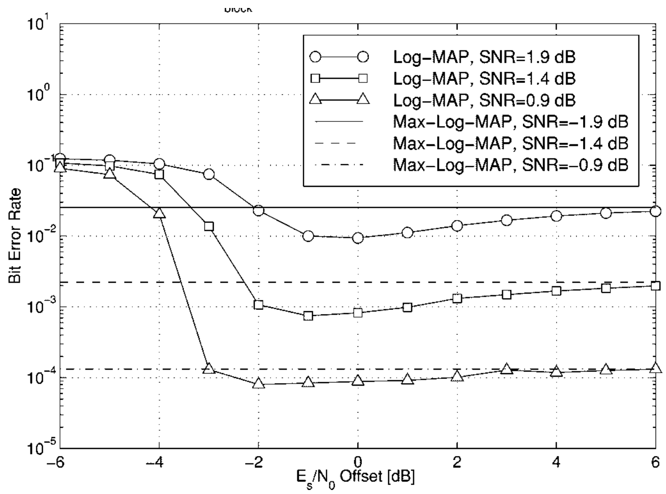

Fig. 3. SNR estimation offset, AWGN channel. $R = 1/2,\ m = 2,\ \mathrm{L_{block}} = 600$, random interleaver, 10 iterations.

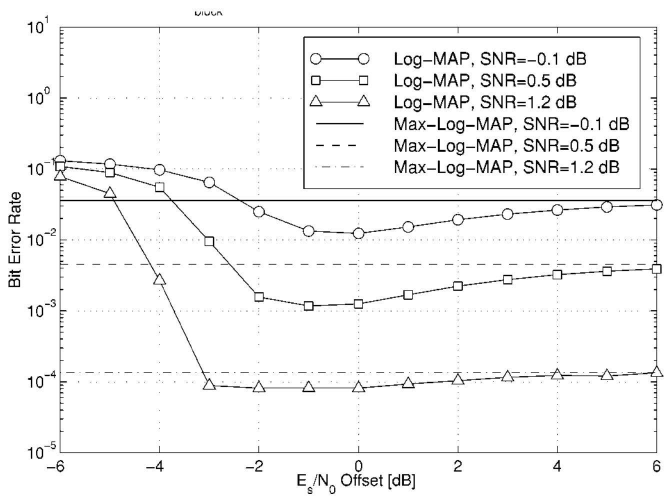

Fig. 4. SNR estimation offset, Rayleigh fading channel. $R = 1/2,\ m = 2,\ \mathrm{L_{block}} = 600$, random interleaver, 10 iterations.

B. SNR Estimation Offset

Figs. 3 and 4 show the BER versus SNR estimation offset for AWGN and for Rayleigh fading channels, respectively. For 10 iterations, each graph features three curves for the Log-MAP decoder and for the Max-Log-MAP decoder. These correspond approximately to BER's $\in \{10^{-2}, 10^{-3}, 10^{-4}\}$. As opposed to simulating the effects of using the SNR at the design points, here we simulate errors in estimating that SNR. Therefore, a mismatch of up to 6 dB is taken into account.

Note that in Figs. 3 and 4 the minimum BER is not achieved for an offset-SNR of 0 dB, as we may expect. This effect, however, occurs only for the short block sizes under consideration. For long block sizes we could prove in further simulations that the minimum will indeed be obtained for an offset-SNR of 0 dB. The simulation results confirm those of [2]–[4], using the (Log-)MAP decoder. Overestimating the SNR is less detrimental than underestimation. However, our simulations also include the Max-Log-MAP decoder, where the picture is different. The Max-Log-MAP decoder yields a better BER for underestimation above 2–4 dB and performs virtually as well as the Log-MAP decoder in case of overestimation above ≈4 dB, independently of the channel model.

IV. Conclusions

We have proven theoretically that Turbo-decoding with the Max-Log-MAP decoder (when properly defined) is SNR independent. Turbo-decoding with the Log-MAP decoder theoretically requires SNR estimates, but simulation results indicate that an estimation of the SNR is not necessary from a practical point of view.

- If channel characteristics remain constant, then the design point is sharply defined, and it is sufficient to use the SNR of that design point. Because of avoiding the risk of estimating a wrong SNR, one can even gain in performance.

  It is safe to assume a design point for another reason: If the actual SNR is smaller than the SNR of the design point, then the decoder will not perform in a satisfying manner, anyway. If the actual SNR is larger, then the achievable BER will be even better than anticipated.
- If channel characteristics change over time, the offset simulations recommend that one should use a set of design points. If this is not convenient, one may use the Max-Log-MAP decoder. Our results for the AWGN and Rayleigh channels can be interpolated for mobile communication channels affected by Rician fading.

Acknowledgment

The authors would like to thank Dr. F. Berens, T. Bing, H. Michel, and Dr. W. Sauer-Greff for initial discussions.

References

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

[2] M. Jordan and R. Nichols, "The effects of channel characteristics on turbo code performance," in *Proc. Milcom'96*, McLean, VA, Oct. 1996, pp. 17–21.

[3] M. C. Reed and J. Asenstorfer, "A novel variance estimator for turbo-code decoding," in *Proc. Int. Conf. on Telecommun.*, Melbourne, Australia, Apr. 1997, pp. 173–178.

[4] T. A. Summers and S. G. Wilson, "SNR mismatch and online estimation in turbo decoding," *IEEE Trans. Commun.*, vol. 46, pp. 421–423, Apr. 1998.

[5] M. C. Valenti and B. D. Woerner, "Performance of turbo-codes in interleaved flat fading channels with estimated channel state information," in *Proc. VTC'98*, Ottawa, Canada, May 1998, pp. 66–70.

[6] P. Frenger, "Turbo decoding on Rayleigh fading channels with noisy channel estimates," in *Proc. VTC'99*, May 1999.

[7] F. Berens, A. Worm, H. Michel, and N. Wehn, "Implementation aspects of turbo-decoders for future radio applications," in *Proc. VTC'99 Fall*, Amsterdam, The Netherlands, Sept. 1999, pp. 2601–2605.

[8] S. Benedetto and G. Montorsi, "Design of parallel concatenated convolutional codes," *IEEE Trans. Commun.*, vol. 44, pp. 591–600, May 1996.

[9] P. Hoeher, "New iterative ("Turbo") decoding algorithms," in *Proc. Int. Symp. on Turbo Codes & Related Topics*, Brest, France, 1997, pp. 63–70.

[10] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *ETT*, vol. 8, no. 2, pp. 119–125, Mar./Apr. 1997.