Advances in Radio Science (2003) 1: 259–263

Advances in Radio Science

Efficient MAP-algorithm implementation on programmable architectures

F. Kienle, H. Michel, F. Gilbert, and N. Wehn

University of Kaiserslautern, Germany

Abstract. Maximum-A-Posteriori (MAP) decoding algorithms are important HW/SW building blocks in advanced communication systems due to their ability to provide soft-output informations which can be efficiently exploited in iterative channel decoding schemes like Turbo-Codes. Multi-standards demand flexible implementations on programmable platforms.

In this paper we analyze a quantized turbo-decoder based on a Max-Log-MAP algorithm with Extrinsic Scaling Factor (ESF). Its communication performance approximate to a Turbo-Decoder with a Log-MAP algorithm and is less sensitive to quantization effects. We present Turbo-Decoder implementations on state-of-the-art DSPs and show that only a Max-Log-MAP implementation fulfills a throughput requirement of $\sim$2 Mbit/s. The negligible overhead for the ESF implementation strengthen the use of Max-Log-MAP with ESF implementation on programmable platforms.

1 Introduction

Third generation's wireless communication systems comprise advanced signal processing algorithms that increase the computational requirements more than ten-fold over 2G's systems (Third Generation Partnership Project). Numerous existing and emerging standards require flexible implementations ("software radio") (Greifendorf et al., 2002). It is argued in Bickerstaff et al. (2000) that "the trends in decoding algorithms are moving from standard Viterbi towards more computationally-expansive algorithms like soft-output Viterbi algorithm (SOVA) and maximum a posteriori (MAP) algorithm. The implementation efficiency of these algorithms will become a differentiating factor for next generation wireless communications – particularly for those employing programmable DSP-devices."

Turbo-Codes, introduced by Berrou et al. (1993), have near Shannon-limit error correction capacity and are among the most advanced channel-coding algorithms and thus used in many communication standards like the 3GPP standard. The important innovation was the reintroduction of iterative decoding schemes of convolutional codes by means of soft-output information exchange. The basic building block of a turbo-decoder is the component decoder which provides soft-output information. This information is a measure on the decoder confidence in its decoding decision. These component decoders are typically based on the MAP algorithm.

To reduce the MAP implementation complexity the algorithm is transformed into the logarithm-domain (Log-MAP) (Robertson et al., 1997) which reduces the operation strength but implicates the so-called max^* operation. This operation is composed of a maximum search and a correction term. Discarding the correction term results in the so called Max-Log-MAP algorithm, its implementation is less complex and thus faster but implies a loss in the communication performance of up to 0.3 dB.

The implementation of the max^* operation is especially very time consuming on standard DSP-processors. The additional assembler commands to add the correction term decreases the overall throughput of a Log-MAP implementation by a factor 2–3 compared to a Max-Log-MAP implementation. (the throughput degradation in a dedicated VLSI implementation is less worse). This trade-off between communication performance versus implementation complexity is a typical problem in the design of communication systems. Recently some very advanced DSPs like the TigerSharc from Anlog Devices have implemented a special max^* instruction to support an efficient Log-Map implementation.

In this paper we investigate different MAP-algorithms with respect to their communication performance and implementation complexity on state-of-the-art DSPs. In Vogt et al. (1999) an Extrinsic Scaling Factor (ESF) was proposed to improve the turbo-decoder performance with Max-Log-MAP component decoders. We observe the usage of ESF in a quantized turbo-decoder model and show that the Max-Log-MAP in combination with ESF is less sensitive to quantization effects than the Log-MAP algorithm. We present different turbo-decoder implementations on modern DSPs and state the enormous throughput difference between a Log-MAP and Max-Log-MAP implementation. The low implementation complexity of ESF and Max-Log-MAP and the

Correspondence to: F. Kienle
(kienle@eit.uni-kl.de)

good communication performance strengthen the use of this combination for turbo-decoder implementations on state-of-the-art DSPs.

The remainder of this paper is structured in two parts: Sect. 2 explains the system model and the MAP algorithm. In Sect. 2.2 we present turbo-decoder performance under quantization and SNR mismatch. Section 3 is devoted to Turbo-Decoder implementation on state-of-the-art DSP architectures: Starcore SC140 from Moterola/Lucent, ST120 from ST-Microelectronics and TigerSharc from AnalogDevices. Section 4 concludes this paper.

2 Turbo-System

Forward error correction is enabled by introducing parity bits. In Turbo-Codes, the original information ($\boldsymbol{x}^s$), denoted as *systematic information*, is transmitted together with the parity information ($\boldsymbol{x}^{1p}, \boldsymbol{x}^{2p}$). In the Third Generation Partnership Project (3GPP), the encoder consists of two recursive systematic convolutional (RSC) encoders with constraint length $K = 4$. One RSC encoder works on the block of information in its original, one on an interleaved sequence, see Fig. 1. On the receiver side a corresponding component decoder for each of them exists. The MAP-Decoder has been recognized as the component decoder of choice as it is superior to the Soft-Output Viterbi Algorithm (SOVA) in terms of communications performance and implementation scalability, see Vogt et al. (1999).

The soft-output of each component decoder ($\boldsymbol{\Lambda}$) is modified to reflect only its own confidence ($\boldsymbol{z}$) in the received information bit of being sent either as "0" or "1". These confidences are exchanged between the decoders to bias their next estimations iteratively. During this exchange, the produced information is interleaved following the same scheme as in the encoder. The exchange continues until a stop criterion, see Worm et al. (2000b), is fulfilled. The last soft-output is not modified and becomes the soft-output of the Turbo-Decoder ($\boldsymbol{\Lambda}^2$). Its sign represents the $0/1$ decision and its magnitude the confidence of the Turbo-Decoder in it.

2.1 The MAP algorithm

Given the received samples of systematic and parity bits (*channel values*) for the whole block (y_0^N, where N is the block length), the MAP algorithm computes the probability for each bit to have been sent as $d_k = 0$ or $d_k = 1$. The logarithmic likelihood ratio (LLR) of these probabilities is the soft-output, denoted as:

$$\Lambda_k = \log \frac{\Pr\{d_k = 1 | y_0^N\}}{\Pr\{d_k = 0 | y_0^N\}}. \tag{1}$$

Equation 1 can be expressed using three probabilities, which refer to the encoder states S_k^m, where $k \in \{0 \ldots N\}$ and $m, m' \in \{1 \ldots 8\}$:

The *branch metrics* $\gamma_{m,m'}^{k,k+1}(d_k)$ is the probability that a transition between S_k^m and $S_{k+1}^{m'}$ has taken place. It is derived from the received signals, the a-priori information given by the previous decoder, the code structure and the assumption of $d_k = 0$ or $d_k = 1$, for details see Robertson et al. (1997). From these branch metrics the probability α_m^k that the encoder reached state S_m^k given the initial state and the received sequence y_0^k, is computed through a forward recursion:

$$\alpha_{m'}^k = \sum_m \alpha_m^{k-1} \cdot \gamma_{m,m'}^{k-1,k}.$$

Performing a backward recursion yields the probability $\beta_{m'}^{k+1}$ that the encoder has reached the (known) final state given the state $S_{m'}^{k+1}$ and the remainder of the received sequence y_{k+1}^N:

$$\beta_m^k = \sum_{m'} \beta_{m'}^{k+1} \cdot \gamma_{m,m'}^{k,k+1}$$

αs and βs are both called *state metrics*. Equation 1 can be rewritten as:

$$\Lambda_k = \log \frac{\sum_m \sum_m' \alpha_m^k \cdot \beta_{m'}^{k+1} \cdot \gamma_{m,m'}^{k,k+1}(d_k = 1)}{\sum_m \sum_m' \alpha_m^k \cdot \beta_{m'}^{k+1} \cdot \gamma_{m,m'}^{k,k+1}(d_k = 0)}. \tag{2}$$

The original probability based formulation implies many multiplications and has thus been ported to the logarithmic domain resulting in the *Log-MAP Algorithm* (Robertson et al., 1997). Multiplications turn into additions and additions into the already mentioned max^* operation which is defined as:

$$max^*(\delta_1, \delta_2) = max(\delta_1, \delta_2) + ln(1 + e^{-|\delta_2 - \delta_1|}). \tag{3}$$

This transformation does not decrease the communication performance. Arithmetic complexity can further be reduced by discarding this correction term which results in a 0.3 dB communication performance loss.

If we multiply the extrinsic information which is passed between the different constituent decoders with an appropriate scaling factor ESF the communication performance can be approximated to that of a Log-MAP decoder (J. Vogt, 2000). Simulations show that the optimal scaling factor is 0.7. For fixed-point implementation (which is a must on DSPs) an ESF=0.75 is used, which can be easily implemented by a shift operation.

2.2 Turbo-Decoder performance under quantization and SNR mismatch effects

SNR estimation is a difficult task in wireless communication systems. An imprecise SNR estimation can have strong influence on the communication performance of the decoding process. Worm et al. (2000a) have proven that Turbo-decoding based on a Max-Log-MAP algorithm is SNR independent, whereas the decoding performance with the Log-MAP algorithm depends on the accuracy of the SNR estimation. In this case the authors propose to work with SNR operating points L_{op}.

The estimated SNR values are used to scale the received channel input values. These values are interpreted as Log-Likelihood values and are calculated as follows:

$$\Lambda_k = \frac{4E_s}{N_0} a_k y_k \tag{4}$$

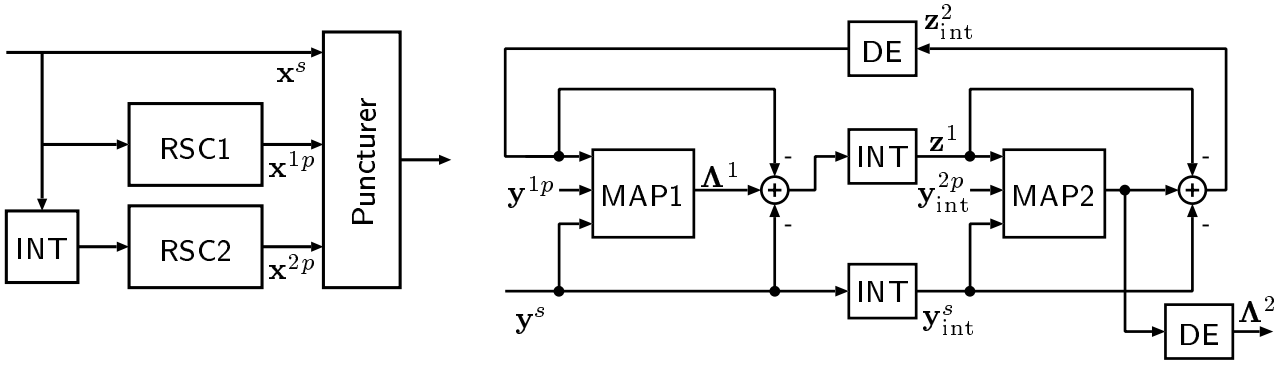


Fig. 1. Turbo-Encoder and Decoder

The factor $\frac{4E_s}{N_0}a_k$, denoted as L_c in the following, is commonly referred to as channel reliability factor (Hagenauer et al., 1996). If the channel characteristics does not change over time, it is sufficient to use a single SNR operating point which is constant, i.e. $L_{op} = L_c = \text{const}$.

The investigations in Worm et al. (2000a) were based on floating-point models - however VLSI- or DSP implementations require fixed-point representations with limited accuracy. The move from floating to fixed-point models (*quantization*) with minimized wordwidth requires a thorough exploration: the bitwidth has to be wide enough to cover the dynamic range and the number of fractional bits has to be large enough to ensure an appropriate processing accuracy (Michel and Wehn, 2001). To avoid an implementation of a multiplier for the channel reliability factor to evaluate Eq. (4), we approximate the SNR operating points with 2^x, $x \in \mathcal{Z}$. Thus the multiplication with L_{op} in Eq. (4) can be substituted by simple shift operations with a resulting SNR operating point granularity of 3 dB.

Figure 2 shows the communication performance of different MAP decoding algorithms. Bit Error Rates (BER) are simulated with an Additive White Gaussian Noise (AWGN) channel, the blocksize is 5114 bits (3GPP standard). Eight iterations are carried out.

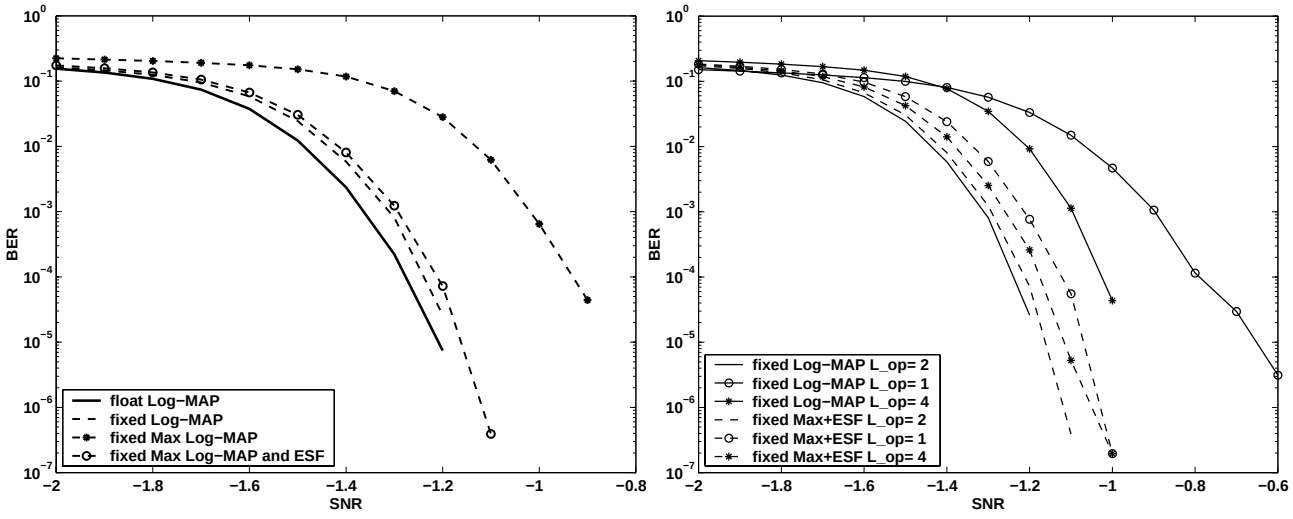
In Fig. 2a we have compared the performance of a floating point Log-MAP algorithm with a fixed-point Log-MAP, a Max-Log-MAP and a Max-Log-MAP combined with ESF=0.75 scaling algorithm respectively. A time invariant stable operating point $L_{op} = 2$ is used, which is an optimum SNR operating point for an AWGN channel. The quantization (bitwidth 6, fractional part 2 bit) is accurate enough to prevent an error floor or any other larger quantization effects. Thus the fixed-point Log-MAP algorithm has a very small performance loss compared to the reference Log-MAP algorithm. The fixed-point Max-Log-MAP algorithm has a degradation of $\sim$0.3 dB due to its algorithmic simplification. But remarkable is that the Max-Log-MAP algorithm with ESF=0.75 scaling degrades only minimal compared to the fixed-point Log-MAP algorithm. Thus in the case of stable operating points we can use the fixed-point Max-Log-MAP algorithm with ESF scaling which has a similar communication performance as the Log-MAP algorithm.

In Fig. 2b we have investigated the influence of SNR mismatches for different fixed-point algorithms. The performance with an optimal SNR operating point ($L_{op} = 2$) is compared with the operating points $L_{op} = 1$ and $L_{op} = 4$ which correspond to a -3 dB and +3 dB SNR mismatch. Obviously the fixed-point Log-MAP algorithm is very sensitive with respect to the SNR operating point e.g. the performance degradation is 0.2 dB for $L_{op} = 4$ and even larger than $\sim$0.4 dB for $L_{op} = 1$. The performance of the Max-Log MAP algorithm with ESF scaling is by far not so sensitive to L_{op} variations. The degradation ranges between 0.05 and 0.1 dB for $L_{op} = 4$ and $L_{op} = 1$ respectively. In a floating-point model all the Max-Log-MAP graphs for the different operating-points would coincide, but due quantization different graphs result.

Under the consideration of quantization effects and SNR mismatches we recommend the use the Max-Log-MAP algorithm with ESF scaling in turbo-decoder implementations.

3 Turbo-Decoder Implementation on modern DSP architectures

Modern DSP architectures attempt to increase the signal processing performance by exploiting the inherent parallelism of many signal processing algorithms. This class of DSP architectures provides several independent ALU units along with wide and fast busses to the internal memories. To allow this increased degree of instruction level parallelism, parallel executed instructions for each active unit are grouped together to so-called very large instruction words (VLIW). Further, the processing units usually support the single-instruction/multiple-data approach (SIMD). This exploits sub-word parallelism (SWP), where several sub-words of a data word can be processed with the same operation. In the following we present state-of-the art DSP architectures and implementation results.



(a) (b)

Fig. 2. BER of UMTS Turbo-Decoder: **(a)** Different MAP implementations 8 iterations, **(b)** Performance of different algorithms dependent on the SNR operating points.

3.1 Motorola/Lucent SC140

The **Starcore SC140** is jointly designed by Motorola and Lucent. A significant design issue is the variable length instruction set (VLES). Most instructions are 16-bit wide, but can be grouped into VLES packtets of 128 bit. We use the SC140 as an example for a multiple ALU VLIW DSP, supporting sub-word parallelism. Its architecture employs 4 ALU and 2 AGU units. In one clock cycle the SC140 is thus able to perform 4 ALU operations, each using 32 bit wide operands, and a 128 bit data transfer. A (Max-)Log-MAP implementation for decoding an 8-state Turbo-Code on this DSP should exploit the benefits of the sub-word parallelism by using 16-bit packed data types.

In Chass and Gubeskys (2000) the update of the path metrics of a Max-Log-MAP, comprising four butterflies, needs 3 clock cycles. This fully utilizes the architectural capabilities of the DSP. One Max-Log-MAP decoder can be realized in 16 cycles per bit. Assuming 5 iterations the resulting Turbo-decoding performance is 1875 kbit/s at 300 MHz. The component decoder with a Log-MAP implementation has a cycle count of approximately 50 cycles per bit, leading to a Turbo-decoding throughput of 600 kbit/s at 300 MHz (see Table 1).

This decrease in decoding performance results from the complexity of the max*-operation (see Eq. 3), which includes the steps: difference of parameters, absolute value, access to lookup-table (LUT), and adding the LUT result to maximum of parameters. The sequence of these steps needs 10 clock cycles in contrast to just one cycle needed for the plain maximum operation (Chass and Gubeskys, 2000).

3.2 STM ST120

The **ST120** is provided by ST-Microelectronics. It features two ALU units and supports three different instruction sets: a 16-bit instruction set (GP16) for compact microcontroller code, a 32-bit instruction set (GP32) for higher performance and more complex instructions, and a third one for an increased level of instruction parallelism. In this 4×32-bit Score-boarded Long Instruction Word (SLIW) mode the processor is able to execute four GP32 instructions in one clock cycle. Following the SIMD approach, the processor supports 2×16-bit data packed into one 32-bit data word.

An optimized hand-coded assembly language implementation of an 8-state Turbo-Decoder has a performance of 37 cycles per bit and MAP, using packed data types and Max-Log-MAP algorithm. Assuming 5 iterations the Turbo-decoding performance results to 540 kbit/s at 200 MHz. For a Log-MAP implementation the throughput degrades to 200 kbit/s with 100 cycles per bit and MAP decoder.

3.3 TigerSharc

The TigerSharc DSP processor from Analog Devices is a high-performance architecture, which is targeted, e.g. for wireless infrastructure applications, such as cellular base stations. With its VLIW architecture, TigerSharc is capable to execute up to four intructions in a single cycle and combines hierarchically both types of data-level parallelism: SIMD and SWP.

The TigerSharc is the only processor with dedicated instruction support to implement the Log-MAP algorithm. A max* instruction is provided, which operates only on a set of enhanced communication registers. Transferring values between the ALU register file and these special registers causes significant data transfer overhead. The max* operation processes a (sub-word parallel) maximum selection of the input parameters and adds a respective correction term value from an integrated lookup-table to the maximum. Thus, full Log-MAP support is achieved without performance penalty.

Table 1. Turbo-Decoder throughput with (Max-)Log-MAP decoder on DSPs

Processor	Architecture	Clock freq.	cycles/(bit · MAP)	Throughput @ 5 iter.
Max-Log-MAP				
ST120	VLIW, 2 ALU	200 MHz	37	540 kbit/s
SC140	VLIW, 4 ALU	300 MHz	16	1875 kbit/s
Log-MAP				
ST120	VLIW, 2 ALU	200 MHz	≈100	≈200 kbit/s
SC140	VLIW, 4 ALU	300 MHz	50	600 kbit/s
ADI TS	VLIW, 2 ALU	180 MHz	27	666 kbit/s

Unfortunately, the integrated LUT is unsymmetrical to zero. Thus, the two parameters of the max* operation are not commutative any more, which complicates the validation of implementation's bit-true behavior versus a bit-true model written in a high-level language.

A single Log-MAP decoder requires 27 cycles per bit. Assuming 5 iterations the overall throughput of this Turbo-Decoder implementation on TigerSharc results to 666 kbit/s at 180 MHz.

3.4 Summary

Table 1 summarizes performance results of 3GPP compliant Turbo-Decoder implementations. The number of cycles per bit and MAP is three times higher for a Log-MAP than for a Max-Log-MAP implementation. The required Turbo-Decoder throughput for UMTS (up to 2 Mbit/s) can only be reached with the SC140 processor, using the Max-Log-MAP algorithm. The Turbo-Decoder with Log-MAP implementation achieves only a throughput of 666 kbit/s. For higher throughput requirements a multiprocessor architecture is mandatory. By using the Max-Log-MAP with ESF=0.75 the communication degradation can be almost avoided with an negligible implementation overhead (1 cycle/(bit· MAP)).

4 Conclusions

Turbo-Codes are part of the 3G cellular wireless standard. The complexity of the decoding algorithm and the throughput requirements pose great demands on the computational power of the signal processing devices. Current DSPs support the kernel operations of the Viterbi algorithm, however for the MAP algorithm, this support is lacking. Using VLIW DSPs one 3G data channel can be processed using the sub-optimal Max-Log-MAP algorithm. This validates the throughput of 1875 kbit/s for the Starcore SC140. Therefore we propose a Max-Log-MAP with Extrinsic Scaling Factor. Its communication performance is close to the performance of a Log-MAP implementation and is less sensible to SNR mismatch. The implementation complexity of the ESF Max-Log-MAP is nearly equal to the Max-Log-MAP and can be implemented on standard DSPs without specific instruction extensions.

Acknowledgements. This work has been supported by the Deutsche Forschungsgesellschaft (DFG) under grant We 2442/1-1 within the Schwerpunktprogramm "Grundlagen und Verfahren verlustarmer Informationsverarbeitung (VIVA)".

References

Berrou, C., Glavieux, A., and Thitimajshima, P.: Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes, in: Proc. 1993 International Conference on Communications (ICC '93), pp. 1064–1070, Geneva, Switzerland, 1993.

Bickerstaff, M., Hughes, G., Nicol, C., Xu, B., and Yan, R.-H.: DSP Systems for Next-Generation Mobile Wireless Infrastructure, in: Proc. ICASSP 2000, pp. 3710–3713, 2000.

Chass, A. and Gubeskys, A.: On Performance/Complexity Analysis and SW Implementation of Turbo Decoding, in: Proc. 2nd International Symposium on Turbo Codes & Related Topics, pp. 531–534, Brest, France, 2000.

Greifendorf, D., Stammen, J., and Jung, P.: The Evolution of Hardware Platforms for Mobile 'Software Defined Radio' Terminals, in: Proc. 2002 International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC '02), 2002.

Hagenauer, J., Offer, E., and Papke, L.: Iterative Decoding of Binary Block and Convolutional Codes, IEEE Transactions on Information Theory, 42, 429–445, 1996.

J. Vogt, A. F.: Improving the Max-Log-MAP Turbo Decoder, IEEE Electronic Letters, 36, 2000.

Michel, H. and Wehn, N.: Turbo-Decoder Quantization for UMTS, IEEE Communications Letters, 5, 55–57, 2001.

Robertson, P., Hoeher, P., and Villebrun, E.: Optimal and Sub-Optimal Maximum a Posteriori Algorithms Suitable for Turbo Decoding, European Transactions on Telecommunications (ETT), 8, 119–125, 1997.

Third Generation Partnership Project, 3GPP home page, www.3gpp.org.

Vogt, J., Koora, K., Finger, A., and Fettweis, G.: Comparison of Different Turbo Decoder Realizations for IMT-2000, in: Proc. 1999 Global Telecommunications Conference (Globecom '99), vol. 5, pp. 2704–2708, Rio de Janeiro, Brazil, 1999.

Worm, A., Hoeher, P., and Wehn, N.: Turbo-Decoding without SNR Estimation, IEEE Communications Letters, 4, 193–195, 2000a.

Worm, A., Michel, H., Gilbert, F., Kreiselmaier, G., Thul, M. J., and Wehn, N.: Advanced Implementation Issues of Turbo-Decoders, in: Proc. 2nd International Symposium on Turbo Codes & Related Topics, pp. 351–354, Brest, France, 2000b.