On optimal and near-optimal turbo decoding using generalized max operator

On Optimal and Near-Optimal Turbo Decoding Using Generalized $\max^*$ Operator

Stylianos Papaharalabos, *Member, IEEE*, P. Takis Mathiopoulos, *Senior Member, IEEE*, Guido Masera, *Senior Member, IEEE*, and Maurizio Martina, *Member, IEEE*.

***Abstract*—Motivated by a recently published robust geometric programming approximation, a generalized approach for approximating efficiently the $\max^*$ operator is presented. Using this approach, the $\max^*$ operator is approximated by means of a generic and yet very simple $\max$ operator, instead of using additional correction term as previous approximation methods require. Following that, several turbo decoding algorithms are obtained with optimal and near-optimal bit error rate (BER) performance depending on a single parameter, namely the number of piecewise linear (PWL) approximation terms. It turns out that the known Max-Log-MAP algorithm can be viewed as special case of this new generalized approach. Furthermore, the decoding complexity of the most popular previously published methods is estimated, for the first time, in a unified way by hardware synthesis results, showing the practical implementation advantages of the proposed approximations against these methods.**

***Index Terms*—Turbo codes, iterative decoding.**

I. Introduction

OVER the past decade or so several algorithmic approaches aiming to simplify the well-known $\max^*$ operator [1] for decoding turbo codes [2] have been proposed and analyzed. These algorithms include: Improved Max-Log-MAP [3], Constant Log-MAP [4], Linear Log-MAP [5], Average Log-MAP [6], and recently the algorithms in [7], [8]. The penalty paid for such approximations is a small bit error rate (BER) performance degradation as compared with the performance achieved by the optimal Log-MAP algorithm [9]. However, although these sub-optimal algorithms are computationally simpler as compared with the optimal solution, an additional correction term is required to be added to the $\max$ operation in order to minimize performance degradation.

Geometric programing is an optimization problem used in various fields, such as information theory, analog/RF circuit design, power control of wireless communication networks and statistics [10]. However, its application to turbo decoding has not been investigated so far. Recently, the authors in [11] have dealt with the convex log-sum-exp (lse) function and considered a robust geometric programming problem as robust linear programming problem. In particular, a constructive algorithm was presented in order to find the best piecewise linear (PWL) approximation terms of the bivariate lse function. Their research has shown that the exact number of PWL terms depends on the approximation error resulting from the original bivariate lse function.

In this letter, we modify and apply for the first time the optimal approximations presented in [11] to decode turbo codes noting that the bivariate lse function is equivalent to the $\max^*$ operator. Consequently, a generalized approximation for the $\max^*$ operator is obtained using: (i) a small number of efficient PWL terms that ease a hardware implementation; and (ii) the very simple $\max$ operator. The main feature of this approximation is that there is no need to use any correction term, in contrast to all previously published methods [4]–[8]. Furthermore, it turns out that the known Max-Log-MAP algorithm can be viewed as special case of this new approach, when two PWL terms are considered. Our research has shown that by considering four PWL approximation terms with the additional use of scaling as in [3], the resulting turbo code BER performance is essentially identical to the performance of the optimal Log-MAP algorithm. Hardware synthesis results have shown that by considering three PWL approximation terms, the resulting algorithm outperforms most of the previously published methods, i.e. [5]–[8], in terms of occupied area savings, and achieves near Log-MAP performance.

II. Reduced Complexity Turbo Decoding

In this section the most important reduced complexity turbo decoding algorithms are reviewed. Consider an information sequence of N bits denoted with $\bar{u} = [u_1, u_2, \ldots, u_N]$. This sequence is turbo encoded, then each coded bit is binary phase-shift keying (BPSK) modulated, taking values from the alphabet $\{\pm 1\}$ with equal probabilities, and transmitted with bit energy E_b over an additive white Gaussian noise (AWGN) channel with one-sided power spectral density N_o. At the receiver the turbo decoder estimates the transmitted sequence of bits.

The $\max^*$ operation, i.e. Jacobian logarithm, used in turbo decoding is defined as [1]

$$\max^*(x_1, x_2) \stackrel{\triangle}{=} \log\{\exp(x_1) + \exp(x_2)\} = \max(x_1, x_2) + \ln\{1 + \exp(-|x_1 - x_2|)\} = \max(x_1, x_2) + f_c(|x_1 - x_2|) \tag{1}$$

where $f_c(|x_1 - x_2|)$ is a non-linear function referred to as 'correction term' [9] and $|.|$ denotes absolute value. For more

S. Papaharalabos and P. T. Mathiopoulos are with the Institute for Space Applications and Remote Sensing (ISARS), National Observatory of Athens, Vas. Pavlou and Metaxa Str., P. Penteli, 15236, Athens, Greece (e-mail: spapaha@space.noa.gr).

G. Masera and M. Martina are with the VLSI Lab, Dipartimento di Elettronica, Politecnico di Torino, corso Duca degli Abruzzi 24, 10129 Torino, Italy.

This research work was supported by EU IST NEWCOM++ (IST-216715) project.

Manuscript received Month xx, 2009.

than two arguments, the Jacobian logarithm is applied recursively. For example, considering three arguments, it yields

$$\max^*(x_1, x_2, x_3) = \max^* \{\max^*(x_1, x_2), x_3\}. \qquad (2)$$

For the Log-MAP algorithm, a look-up table (LUT) substitutes $f_c(|x_1 - x_2|)$, which is usually implemented with eight values [9]. If LUT is omitted, then the Log-MAP simplifies to the Max-Log-MAP algorithm. In the past, several reduced complexity decoding algorithms have been obtained by approximating $f_c(|x_1 - x_2|)$ with different methods, e.g. see [4]–[8], a summary of which is presented in Table I.

Having evaluated the performance of reduced complexity decoding algorithms given in Table I and as shown in Fig. 1 (for the turbo code simulation parameters see the next section), it is concluded that at BER of 10^{-4}: (i) The Max-Log-MAP is the worst performing algorithm with approximately 0.4 dB degradation as compared with Log-MAP; (ii) The maximum performance degradation of reduced complexity algorithms as compared with Log-MAP is approximately 0.1 dB; and (iii) the Linear Log-MAP and also the algorithm of [7] achieve the best, i.e. near Log-MAP performance. A thorough complexity estimation of these algorithms is reported in the next section.

III. Optimal $\max^*$ Approximations and Their Application to Turbo Decoding

From pure mathematical curiosity, the authors in [11] have, instead of approximating $f_c(|x_1 - x_2|)$, approximated (1) as a whole, i.e. the $\max^*$ operator directly. Hence, (1) becomes

$$\max^*(x_1, x_2) \approx \max(\kappa_1 * x_1 + \lambda_1 * x_2 + \mu_1, \ldots, \kappa_i * x_1 + \lambda_i * x_2 + \mu_i) \qquad (3)$$

where κ_i, λ_i, and μ_i are real positive values and $i \geq 2$. The best PWL approximations of the $\max^*$ operator with different number of terms are shown in Table II. The approximation error reduces in the order of $\sqrt{2}/r^2$ and for practical applications $5 \leq r \leq 10$ has been considered [11]. It is underlined that in case of turbo decoding, the $r = 2$ approximation is identical to the Max-Log-MAP algorithm.

Performance evaluation results for various values of r, with the additional use of scaling as in [3], have shown that at BER of 10^{-5}: (i) Both $r = 5$ and $r = 4$ approximations achieve essentially identical to the Log-MAP performance; and (ii) The $r = 3$ approximation has performance degradation of less than 0.03 dB against the Log-MAP algorithm. In order to ease a hardware implementation, the $r = 3$ and $r = 4$ approximations have been modified, respectively as

$$\max^*(x_1, x_2) \approx \max[x_1, 0.5 * (x_1 + x_2 + 1), x_2] \qquad (4)$$

$$\begin{aligned} \max^*(x_1, x_2) \approx \max(&x_1, 0.25 * x_1 + 0.75 * x_2 + 0.5, \\ &0.75 * x_1 + 0.25 * x_2 + 0.5, x_2). \end{aligned} \qquad (5)$$

In terms of implementation, synthesizable VHDL descriptions have been produced for the $\max^*$ approximations of (4) and (5) as well as for Log-MAP with LUT, Max-Log-MAP and the algorithms shown in Table I. In order to derive fair comparisons, the same area optimization effort of the synthesis tool must be guaranteed for all cases. To this purpose, although all considered implementations of the $\max^*$ operation are pure combinational architectures, registers have been placed at the architecture inputs and output. This allows setting a unique clock frequency constraint for all considered cases, $f_{CK} = 200$ MHz. Synthesis results obtained in terms of area occupied by the combinational part on a 130 nm standard cell CMOS technology are given in Table III for precision metrics represented with 8, 10 and 12 bits, respectively. These results show, on the one hand, that the proposed $r = 3$ approximation outperforms most of the previously published methods, i.e. [5]–[8]. In particular, its occupied area is 35% smaller than that required by the Log-MAP algorithm and it is only inferior to the Constant Log-MAP algorithm by 8%. On the other hand, the proposed $r = 4$ approximation has comparable complexity with the method in [7] and outperforms [5], [6], [8].

Performance evaluation results have been obtained for the most efficient method, i.e. Constant Log-MAP, the $r = 3$ and $r = 4$ proposed approximations, and these are illustrated in Fig. 2. A 16-states turbo code is considered with coding rate equal to 1/2 and generator polynomials $(1, 33/23)_o$ in octal form representing the feed-forward and backward polynomials, respectively. Furthermore, an information sequence of $N = 10^3$ bits is assumed, whereas the total number of transmitted frames is 10^5. A pseudo-random turbo interleaver is considered and at the receiver a maximum of 10 decoding iterations are performed. In order to reduce computer simulation time, a genie stopping rule is assumed at the turbo decoder. In computer-based simulations the scaling factor, denoted with s, was constant when varying the number of decoding iterations, having the following values: (i) 0.65 for Max-Log-MAP; (ii) 0.9 for Log-MAP; (iii) 0.85 for Constant Log-MAP; and (iv) 0.75 for the rest of the algorithms.

As shown in Fig. 2, the best performance is achieved by the Constant Log-MAP algorithm and the proposed $r = 4$ approximation followed by the $r = 3$ approximation. In more detail, at BER of 10^{-4} and with respect to the Log-MAP algorithm: (i) The Constant Log-MAP algorithm has performance degradation of approximately 0.03 dB; (ii) The $r = 4$ approximation has comparable performance with the Constant Log-MAP algorithm; and (iii) The $r = 3$ approximation has performance degradation of 0.05 dB. However, both $r = 3$ and $r = 4$ approximations offer practical implementation advantages with respect to other methods.

IV. Conclusion

It has been shown that the $\max^*$ operator used in turbo decoding can be simplified into different number of PWL approximation terms in an efficient way using $\max$ only operation. The proposed $r = 3$ approximation is 0.05 dB inferior to the Log-MAP algorithm but 35% much simpler. Further, it outperforms, in terms of occupied area savings, most of the previously published methods, such as Linear Log-MAP [5], Average Log-MAP [6], and the algorithms from [7], [8]. The proposed $r = 4$ approximation is only 0.03 dB inferior to the Log-MAP algorithm and it it less complex than the methods presented in [5], [6], [8].

TABLE I
THE MOST IMPORTANT REDUCED COMPLEXITY TURBO DECODING ALGORITHMS USING DIFFERENT APPROXIMATION IN $f_c(|x_1 - x_2|)$ OF (1).

	Constant Log-MAP [4]	Linear Log-MAP [5]	Average Log-MAP [6]
$f_c(\|x_1 - x_2\|)$	$\begin{cases} 3/8, & \text{if } \|x_1 - x_2\| < 2 \\ 0, & \text{otherwise} \end{cases}$	$\max(\ln 2 - 0.25 * \|x_1 - x_2\|, 0)$	$\begin{cases} \ln 2 + 0.5 * (x_1 + x_2), & \text{if } \|x_1 - x_2\| < 2 * \ln 2 \\ 0, & \text{otherwise} \end{cases}$
	Ref. [7]	Ref. [8]	Max-Log-MAP
$f_c(\|x_1 - x_2\|)$	$\ln 2 * 2^{-\|x_1 - x_2\|}$	$\max(0, \ln 2 - 0.5 * \|x_1 - x_2\|)$	0

TABLE II
BEST APPROXIMATIONS OF THE $\max^*$ OPERATION USING DIFFERENT NUMBER OF PWL TERMS AND DEPLOYING $\max$ ONLY OPERATION [11].

No. of Terms (r)	Resulting $\max^*$ Approximation	No. of $\max$ Ops.	Approx. Error
2	$\max(x_1, x_2)$	1	0.693
3	$\max(x_1,$ $0.5 * (x_1 + x_2) + 0.693,$ $x_2)$	2	0.223
4	$\max(x_1,$ $0.271 * x_1 + 0.729 * x_2 + 0.584,$ $0.729 * x_1 + 0.271 * x_2 + 0.584,$ $x_2)$	3	0.109
5	$\max(x_1,$ $0.167 * x_1 + 0.833 * x_2 + 0.45,$ $0.5 * x_1 + 0.5 * x_2 + 0.693,$ $0.833 * x_1 + 0.167 * x_2 + 0.45,$ $x_2)$	4	0.065

TABLE III
OCCUPIED AREA COMPARISON (SQUARE μm) OF DIFFERENT $\max^*$ APPROXIMATIONS ON A 130 NM STANDARD CELL TECHNOLOGY AND VARIOUS PRECISION METRICS REPRESENTATIONS.

Algorithm	8 bits	10 bits	12 bits
Max-Log-MAP	250.133	312.666	373.182
Log-MAP	1022.72	1276.888	1613.76
$r = 3$ of (4)	653.573	831.086	1006.583
$r = 4$ of (5)	883.534	1149.804	1426.160
Constant Log-MAP [4]	599.108	764.519	931.946
Linear Log-MAP [5]	978.342	1264.784	1559.296
Average Log-MAP [6]	1069.116	1363.627	1555.261
Ref. [7]	891.602	1135.684	1377.748
Ref. [8]	1137.701	1456.418	1758.988

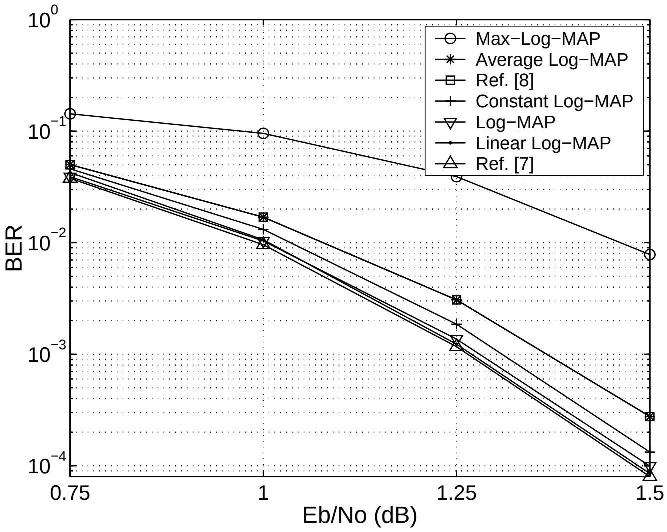

Fig. 1. Turbo code BER performance comparison with the most important reduced complexity decoding algorithms.

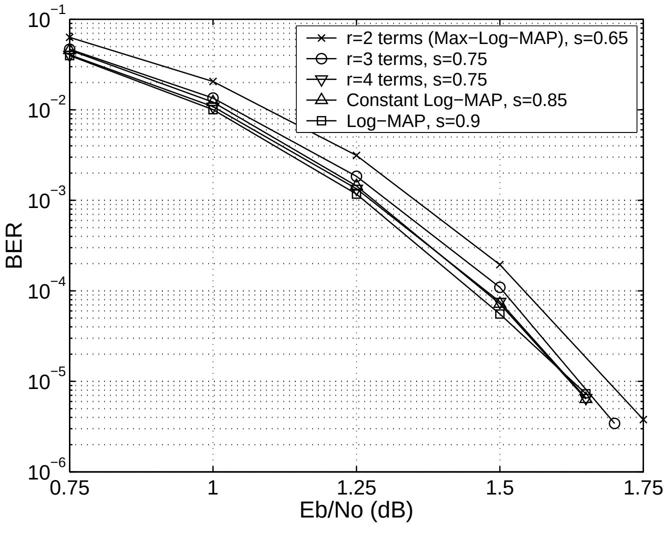

Fig. 2. Turbo code BER performance with efficient approximations of the $\max^*$ operator and different number of PWL terms. The extrinsic information is scaled by a factor of s.

REFERENCES

[1] A. J. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 260–264, Feb. 1998.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo codes," in *IEEE Int. Conf. Commun. (ICC)*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[3] J. Vogt and A. Finger, "Improving the Max-Log-MAP turbo decoder," *IEE Electron. Lett.*, vol. 36, no. 23, pp. 1937–1939, Nov. 2000.

[4] W. J. Gross and P. G. Gulak, "Simplified MAP algorithm suitable for implementation of turbo decoders," *IEE Electron. Lett.*, vol. 34, no. 16, pp. 1577–1578, Aug. 1998.

[5] J. F. Cheng and T. Ottosson, "Linearly approximated Log-MAP algorithms for turbo decoding," in *IEEE Vehic. Techn. Conf. (VTC), Spring*, Tokyo, Japan, May 2000, pp. 2252–2256.

[6] B. Classon, K. Blankenship, and V. Desai, "Channel coding for 4G systems with adaptive modulation and coding," *IEEE Wireless Commun. Mag.*, vol. 9, no. 2, pp. 8–13, Apr. 2002.

[7] H. Wang, H. Yang, and D. Yang, "Improved Log-MAP decoding algorithm for turbo-like codes," *IEEE Commun. Lett.*, vol. 10, no. 3, pp. 186–188, Mar. 2006.

[8] S. Talakoub, L. Sabeti, B. Shahrrava, and M. Ahmadi, "An improved Max-Log-MAP algorithm for turbo decoding and turbo equalization," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 3, pp. 1058–1063, Jun. 2007.

[9] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE Int. Conf. Commun. (ICC)*, Seattle, USA, Jun. 1995, pp. 1009–1013.

[10] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[11] K. L. Hsiung, S. J. Kim, and S. Boyd, "Tractable approximate robust geometric programming," *Springer Optim. Eng. Journal*, vol. 9, no. 2, pp. 95–118, Jun. 2008, available on-line.