A Novel Suboptimal Piecewise-Linear-log-MAP Algorithm for Turbo Decoding

Yu.Yu. Ivanov, A.N. Romanyuk, A.Ia. Kulyk
Faculty of Automatics, Electronics and Computer Systems
Vinnytsia National Technical University
Vinnytsia, Ukraine
E-mail: ran12345@mail.ru, YuraII@yandex.ru

O.V. Stukach
Department of Computer Measurement Systems and Metrology
National Research Tomsk Polytechnic University
Tomsk, Russia
E-mail: tomsk@ieee.org

***Abstract*—In this paper, a new suboptimal modification for the log-MAP turbo decoding algorithm is proposed. This method (PL-log-MAP) based on the piecewise linear approximation of the the compensation term (correction function) in the Jacobian logarithm used by the MAP (BCJR) decoder. Using the proposed approximation, the complex functions ln(.) and exp(.) in the log-MAP algorithm can be estimated with high accuracy and lower computational complexity. The efficacy of the proposed approximation is investigated and demonstrated by applying it to digital communication system. The performance of the PL-log-MAP algorithm is shown to have the closest performance to the original log-MAP solution.**

Keywords: turbo code, iterative decoding, MAP (BCJR) algorithm, approximation, log-MAP, max-log-MAP, Jacobian logarithm, correction function, compensation term, BER .

I. Introduction

At the time of the latest technology an information became the object of an automated processing. The data transmission process in information systems is susceptible to errors, because any error in the calculations can breach them. To combat with interferences in data transmission systems error-correcting coding is used to provide reliability and credibility of transmitted information. The analysis allowed us to determine, that the most prominent achievement in the theory of error-correcting coding in recent years is turbo code – the powerful algorithm of the modern communication systems. This code is the iterative probabilistic method of error-correcting coding with the reliable performance, which is very close to the C.E. Shannon theoretical limit [1]. Turbo codes are used to encode large volume information messages at the high speed with high error-correction.

In 1993, at a moment when there were not many people to believe in the practicability of capacity approaching codes, the presentation of turbo codes by C. Berrou *et al* [2] was a revival for the channel coding research community. Historical turbo codes, also sometimes called parallel concatenated convolutional codes, are based on a parallel concatenation of two recursive systematic convolutional codes (RSCC) separated by an inter leaver (INT). They are called turbo in reference to the analogy of their decoding principle with the turbo principle of a turbo-compressed engine, which reuses the exhaust gas in order to improve efficiency. The turbo decoding principle calls for an iterative algorithm involving two component decoders exchanging information in order to improve the error correction performance with the decoding iterations.

The near-capacity performance of turbo codes and their suitability for practical implementation explain their adoption in various standards − TV, mobile and space communications. For example, NASA JPL research (Mars Reconnaissance Orbiter + Mars Science Laboratory − rover Curiosity) on Mars allowed to receive 24 GB high-quality photos and videos from Red Planet's surface. But firstly, they were chosen in the telemetry coding standard by the Consultative Committee for Space Data Systems (CCSDS) and for the medium to high data rate transmissions in the third generation mobile communication 3GPP/UMTS standard. They have further been adopted as part of the digital video broadcast–return channel via satellite (DVB–RCS) and terrestrial (DVB–RCT) links. More recently, they were also selected for the next generation of 3GPP2/cdma2000 wireless communication systems as well as for the IEEE 802.16 standard (WiMAX) intended for broadband connections over long distances. Turbo codes are used in several Inmarsat's communication systems, such as in the new Broadband Global Area Network (BGAN) that entered service in 2006 [3]-[5]. In genetics and bioinformatics for working with DNA turbo coding constructions are also used along with the Reed-Solomon code [6].

Creating error-correcting code is closely associated with decoding algorithms. Almost all of the codes can be decoded only exhaustive search methods, while decision variants are greater than the number of atoms in the Universe. Therefore it is necessary ***to find and explore*** nonexhaustive decoding methods, ***to provide decoding quality***, considering the conditions of real communication systems.

There are two main turbo decoding algorithms, such as Soft-Output-Viterbi-Algorithm (SOVA, 1989) [7] and BCJR (MAP, 1974) algorithm [8]. The SOVA differs from the classical Viterbi Algorithm (VA, 1969) [7, 9] in that it uses a modified path metric, which takes into account the a priori

978-1-4799-7103-9/15/$31.00 ©2015 IEEE

probabilities of the input symbols, and produces a soft output indicating the reliability of the decision. We will not consider SOVA in our article (watch [7] and [9]), because our objective is MAP algorithm, which is more efficient, but with high numerical difficulty. The MAP algorithm is too complex for practical implementation in real system. So, we work with BCJR in log domain or log-MAP algorithm [10]. In log-MAP turbo decoding, the complicated log exponential sum is often simplified with the Jacobian logarithm which consists of the max operation along with an exponential correction function. Although the max-log-MAP algorithm [10] reduces the complexity of the Jacobian logarithm implementation by omitting the correction function, its performance is inferior to the original log-MAP algorithm [10–12]. Hence, a simple approximation to the correction function is needed to complement the max-log-MAP algorithm.

This paper presents a simplified suboptimal log-MAP. The algorithm employs piecewise linear fitting methods to compute the correction function that will be used in the calculations of state metrics and ultimately the log-likelihood ratios (*LLR*s). In this algorithm, only linear multiplication, addition, comparator are required to obtain near log-MAP performance.

The rest of the paper is organized as follows: in Section II, a brief review of the original MAP, log-MAP are presented. Section III reviews existing methods for the approximation of the correction function. In Section IV we present the novel piecewise approximation for the compensation term in the Jacobian logarithm for the log-MAP decoding algorithm. Next, we compared numerical results of our method with other approximations. Then the simulation with Monte-Carlo method for our PL-log-MAP turbo decoding algorithm is presented. The paper is finally concluded in Section V.

II. TURBO DECODING WITH OPTIMAL ALGORITHMS

In this section, the derivation of the MAP algorithm will not be detailed but the results of the algorithm will be stated. Following this, a review of the log-MAP algorithm will be presented. For details on the derivation, see [2, 8–13].

A. Turbo Principle

The main principle of the classical turbo encoding is the usage of two parallel working constituent encoders, although you can use the arbitrary dimension encoders (Fig. 1). The information block is encoded twice, the second time − after interleaving process, which helps to create the stream of independent errors (eliminate error packets). The perforation patterns for turbo codes are used to increase the data rate. The concept of log-likelihood ratio is used in the decoding process. The decoded information (*N* bits) from the first (second) decoder output is used as a priori information for the second (first) decoder input in order to clarify the decoding result using iterative C. Berrou algorithm BCJR [2] on maximum a posteriori probability (modification of the initial BCJR [8]). This is very efficient soft decoding algorithm for turbo codes, which minimizes the probability of bit error (bit-by-bit decoding). After exiting from the second decoder output the soft values of the extrinsic log-likelihood ratio is deinterleaved and hard decision on the transmitted data bit is accepted (Fig. 2).

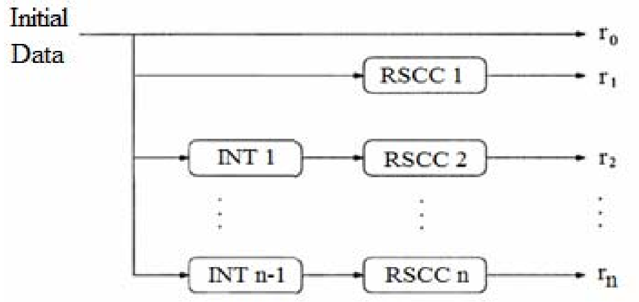

FIG. 1. N-DIMENSIONAL TURBO ENCODER

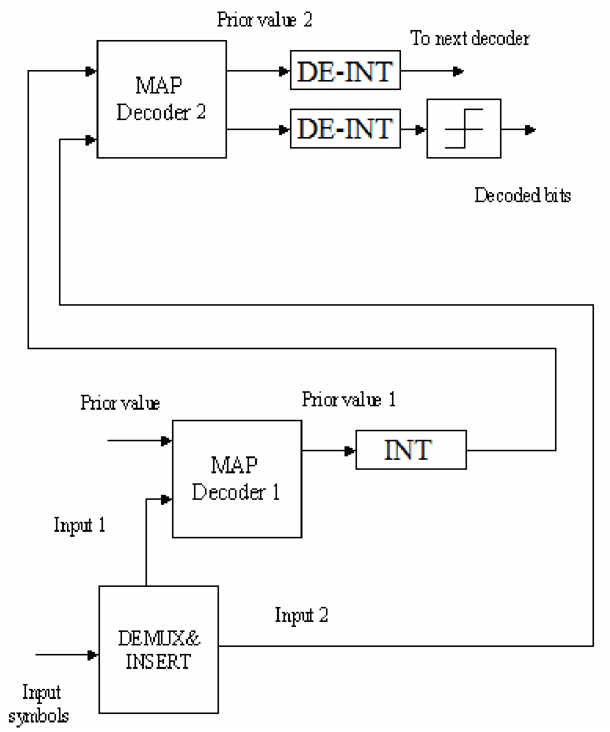

FIG. 2. TURBO DECODER (TWO ITERATIONS)

B. Maximum A Posteriori Probability Algorithm or MAP

The idea of the BCJR turbo decoder is to modify the algorithm, which was first introduced in 1974 by L. Bahl, J. Cocke, F. Jelinek and J. Raviv [8]. Such algorithm in modern coding theory has several names: BCJR (the first letters of the authors names), MAP (maximum a posteriori probability), APP (for a posteriori probability), FBA (forward-backward algorithm), BPA (belief propagation algorithm), SPA (sum-product algorithm).

During MAP-decoding it is necessary to calculate the a posteriori probabilities of information symbols, using received sequence $\bar{r}$ and the *LLR*s, that can be written in the formula

$$LLR(\overline{U_i}) = \log\left(\frac{p(\overline{U_i}=1|\overline{r})}{p(\overline{U_i}=0|\overline{r})}\right), \quad (1)$$

where $\overline{U_i}$ is an information symbol, if $LLR(\overline{U_i}) > 0$ the hard binary decision is 1, in otherwise − 0.

Generalized expression for the *LLR* computation procedure in MAP decoding algorithm has the form

$$LLR(\overline{U_i}) = \log\left(\frac{\sum_m\sum_{m'}\alpha_{i-1}(m')\cdot\gamma_i^{(1)}(m',m)\cdot\beta_i(m)}{\sum_m\sum_{m'}\alpha_{i-1}(m')\cdot\gamma_i^{(0)}(m',m)\cdot\beta_i(m)}\right), \quad (2)$$

where $\alpha_i(m) = \sum_{m'}\alpha_{i-1}(m')\cdot\sum_{j=0}^{1}\gamma_i^{(j)}(m',m)$ – forward path metric on the trellis for RSCC, $\gamma_i^{(j)}(m',m)$ – transit or rib metric, $\beta_i(m) = \sum_{m'}\beta_{i+1}(m')\cdot\sum_{j=0}^{1}\gamma_i^{(j)}(m',m)$ – backward path metric on the trellis for RSCC, ln(.) – natural logarithm.

C. Maximum A Posteriori Probability Algorithm in log domain or log-MAP

In log-MAP algorithm the logarithmic metrics are used for reducing the computational complexity of the MAP algorithm. The next expression is obtained from an equation for the forward path metric on the trellis

$$\log a_i(m) = \log\left(\sum_{m'}\sum_{j=0}^{1}\exp(\log a_{i-1}(m') + \log\gamma_i^j(m',m))\right). \quad (3)$$

Similarly, using the formula for obtaining the backward path metric, we can find the expression

$$\log\beta_i(m) = \log\left(\sum_{m'}\sum_{j=0}^{1}\exp(\log\beta_{i+1}(m') + \log\gamma_i^{(j)}(m',m))\right). \quad (4)$$

The rib metric $\gamma_i^{(j)}(m',m)$ in Gaussian channel will have the form

$$\gamma_i^{(j)}(m',m) = p\{\overline{U_i}=j\}\cdot\delta_{ij}(m,m')\times \times\exp\left(-\frac{1}{N_0}\cdot\sum_{q=0}^{N-1}(\overline{r_{i,q}}-\overline{X_{i,q}})^2\right), \quad (5)$$

where $\overline{U_i}=j$ is information bit from the set of ribs $\Theta_i^{(j)}$ between the states $S_i^{(m)}$ and $S_{i-1}^{(m')}$, $j\in\{0, 1\}$, $\overline{X_{i,q}}$ is transmitted symbol, $\overline{r_{i,q}}$ is received value, $\delta_{ij}(m,m')=1$, if $\{m,m'\}\in\Theta_i^{(j)}$ and $\delta_{ij}(m,m')=0$, if $\{m,m'\}\notin\Theta_i^{(j)}$, $\frac{E_b}{N_0}$ – the energy per bit to noise power spectral density ratio.

If we take the logarithm in formula (5) and make reductions in the expression, we get

$$\log\gamma_i^{(j)}(m',m) = \delta_{ij}(m,m')\times \times\left\{\log p\{\overline{U_i}=j\} - \frac{1}{N_0}\cdot\sum_{q=0}^{N-1}(\overline{r_{i,q}}-\overline{X_{i,q}})^2\right\}. \quad (6)$$

Then, if we denote $Y_i^{(j)}(m',m) = \log\gamma_i^{(j)}(m',m)$, $A_i(m) = \log a_i(m)$, $B_i(m) = \log\beta_i(m)$, the expression (2) can be written in the form

$$LLR(\overline{U_i}) = \log\left(\frac{\sum_m\sum_{m'}\exp(A_{i-1}(m') + Y_i^{(1)}(m',m) + B_i(m))}{\sum_m\sum_{m'}\exp(A_{i-1}(m') + Y_i^{(0)}(m',m) + B_i(m))}\right). \quad (7)$$

So in formula (7) we obtained algorithm MAP in logarithmic form as a result of our transformations and simplifications [12].

D. Jacobian Logarithm

The MAP decoder is often operated in the log domain in order to reduce computational complexity. However, the computation for state metrics and *LLR*s is still burdened by the log-exponential-sum calculation. The Jacobian logarithm is used to simplify the log-exponential-sum by employing a correction function f_{cor} along with the maximum operator in the log domain [10]-[13]

$$\log(\exp(x)+\exp(y)) = \max(x,y) + \log(1+\exp(-|x-y|)) = = \max(x,y) + f(|x-y|) = \max(x,y) + f_{cor}(z). \quad (8)$$

Although simplified, the correction function remains a nonlinear exponential function. The manner in which the correction function is calculated is critical to the performance and complexity of the decoder. Several methods have been proposed to simplify its computation, which gives a tradeoff between complexity and performance [14–20], but such decoding algorithms are suboptimal. We will consider these algorithms further in Section III.

III. TURBO DECODING WITH SUBOPTIMAL ALGORITHMS: APPROXIMATIONS TO THE CORRECTION FUNCTION

This section gives a brief review of max-log-MAP algorithm and existing algorithms, which approximate the correction function in order to achieve a simple implementation yet improved performance as compared log-MAP.

A. max-log-MAP Algorithm

To reduce the computational complexity of the log-MAP decoding algorithm Robertson *et al* in [10] proposed to neglect the compensation term f_{cor}

$$\log(\exp(x)+\exp(y)) \approx \max(x, y). \quad (9)$$

As a result the generalized expression (7) takes the form

$$LLR(\overline{U_i}) \approx \max_{m',m}\left\{A_{i-1}(m') + Y_i^{(1)}(m'm) + B_i(m)\right\} - \\ - \max_{m',m}\left\{A_{i-1}(m') + Y_i^{(0)}(m',m) + B_i(m)\right\}. \quad (10)$$

Calculations on the forward and backward paths on the RSCC trellis may be represented as follows

$$A_i(m) = \max_{m'} \max_{j\in\{0,1\}}\left\{A_{i-1}(m') + Y_i^{(j)}(m',m)\right\}, \quad (11)$$

$$B_i(m) = \max_{m'} \max_{j\in\{0,1\}}\left\{B_{i+1}(m') + Y_i^{(j)}(m',m)\right\}. \quad (12)$$

The max-log-MAP simplifies the log-MAP algorithm by simply omitting the f_{cor} altogether. The performance for this algorithm gives up to a 10% performance drop [14], when compared to the log-MAP. The max-log-MAP algorithm is the least complex of all the existing methods, but offers the worst bit error rate (BER) performance. This creates a need to complement the max-log-MAP algorithm with a simple implementation of the correction function in order to improve performance.

B. log-MAP Algorithm with Robertson Lookup Table

Robertson *et al* [10] proposed to approximate f_{cor} by a pre-computed lookup table and they showed that only few values need to be stored. For example, we choose good approximation parameters for the correction term with rule

$$f_{cor}(z) \approx \begin{cases} 0.65, & if \quad 0 \le z \le 0.20, \\ 0.55, & if \quad 0.20 < z \le 0.43, \\ 0.45, & if \quad 0.43 < z \le 0.70, \\ 0.35, & if \quad 0.70 < z \le 1.05, \\ 0.25, & if \quad 1.05 < z \le 1.50, \\ 0.15, & if \quad 1.50 < z \le 2.25, \\ 0.05, & if \quad 2.25 < z \le 3.70, \\ 0, & if \quad z > 3.70. \end{cases} \quad (13)$$

In this approximation, an 8-values correction table results in a negligible performance loss.

C. Constant-log-MAP Algorithm

A slightly more complex variant than the max-log-MAP called constant-log-MAP was proposed in [15]. The constant-log-MAP algorithm is equivalent to a Robertson's approximation with a two-values correction lookup table. In this algorithm the f_{cor} is approximated with the following rule

$$f_{cor}(z) \approx \begin{cases} K, & if \quad z \le T, \\ 0, & if \quad z > T. \end{cases} \quad (14)$$

where K, T is two parameters are optimized by computer search, for example $K = 0.5$, $T = 1.5$.

The BER performance of the constant-log-MAP is between that of the max-log-MAP and the log-MAP algorithms. But the disadvantage of constant-log-MAP is that it is susceptible to noise variance estimation errors than is log-MAP [16].

D. Linear-log-MAP Algorithm

A linear approximation was proposed in [17] for which

$$f_{cor}(z) \approx \begin{cases} F\cdot(z-V), & if \quad z < V, \\ 0, & if \quad z \ge V. \end{cases} \quad (15)$$

where F, V is the two parameters are also optimized via computer search, for example $F = -0{,}24904$, $V = 2{,}5068$.

The linear-log-MAP algorithm is between the log-MAP and the constant-log-MAP algorithms in term of performance and complexity, and it converges faster than the constant-log-MAP algorithm.

E. MacLaurin-Linear-log-MAP Algorithm

In [18], the authors use the MacLaurin series expansion to approximate the f_{cor} about zero. By neglecting Maclaurin's series order two and above, the approximation for the correction term is given as

$$f_{cor}(z) \approx \max\left(0, \ln 2 - \frac{z}{2}\right). \quad (16)$$

This approximation offers better performance than the constant-log-MAP algorithm and requires only a simple linear implementation.

F. Fractional-log-MAP Algorithm

In [19] the correction function is approximated with the following rule

$$f_{cor}(z) \approx \begin{cases} \dfrac{R}{L+z} + C, & if \quad z < D, \\ 0, & if \quad z \ge D \end{cases} \quad (17)$$

The values of *R, L, C* and *D* are calculated in such a way that the error is minimized via computer search. Asoodeh in [19] found that *R* = 1.343 and *L* = 1.405, *C* = –0.25 and *D* = 4 are optimized values.

G. Quadratic-log-MAP Algorithm

A quadratic approximation was proposed by Zhang and Yu in [14] for which, if $z \le 4$ we use an expression

$$f_{cor}(z) \approx 0.0576 \cdot z^2 - 0.3920 \cdot z + 0.6782. \qquad (18)$$

In otherwise $f_{cor} = 0$ is accepted. This quadratic-log-MAP algorithm gets a better compromise between the complexity and the computation precision compared with the log-MAP and max-log-MAP algorithms. Because there are only additions and multiplications in this algorithm.

H. Multistep-log-MAP Algorithm

A more accurate solution to approximate the correction function is given by [20]:

$$f_{cor}(z) \approx \frac{\ln 2}{2^{\lfloor z+0,5 \rfloor}}. \qquad (19)$$

where $\lfloor \ \rfloor$ is floor function.

Division by two can be easily done in digital systems by implementing $\lfloor z+0,5 \rfloor$ number of binary shifts. The algorithm employs shift registers storing the constant ln(2) to perform the division. In order to facilitate fast computation, a high speed shift register is needed for this algorithm.

I. Hybrid-log-MAP Algorithm

Hybrid log-MAP algorithm [20] is based on an approximation of the correction term in the form

$$f_{cor}(z) \approx \begin{cases} 0.6512 - 0.3251 \cdot z, & if\ z < 1.5, \\ \dfrac{0.1635}{2^{\lfloor 0.5 \cdot z \rfloor}}, & if\ z \ge 1.5. \end{cases} \qquad (20)$$

This function also requires $\lfloor 0.5 \cdot z \rfloor$ number of binary shift registers, but faster, and its result is more accurate.

IV. A New Piecewise-linear-log-MAP Algorithm: Simulation and Numerical Results

In this paper, we develop a new modification of the log-MAP algorithm, which is called piecewise-linear-log-MAP (PL-log-MAP) based on the piecewise linear approximation to the correction term. The ***main goal*** of the proposed approximation is to estimate the compensation term in the Jacobian logarithm by another function that is very close to it and that has less computation operations.

A. Numerical Results

The correction function f_{cor} is a descending non-linear function and has a zero asymptote in infinity (Fig. 3). So, when $z \in [0; 4]$, if we divide this range into sub-ranges we can model the correction function by linear functions, for each sub-range, of the form $f_{cor} = a \cdot |x - y| + b = a \cdot z + b$, where *a* and *b* are the slope and the intersection of each linear function. In our case, we take 5 sub-ranges ($z_0 \in [0; 1)$, $z_1 \in [1; 1.5)$, $z_2 \in [1.5; 2)$, $z_3 \in [2; 3)$ and $z_4 \in [3; 4]$). From the Fig. 3, we can remark that when the *z* is greater than 4, the compensation term takes almost a small constant value less than 0.02 (we set this measure on 0.01). Figure 4 shows the comparison for the proposed original function and correction term, which was found via computer search by Ordinary Least

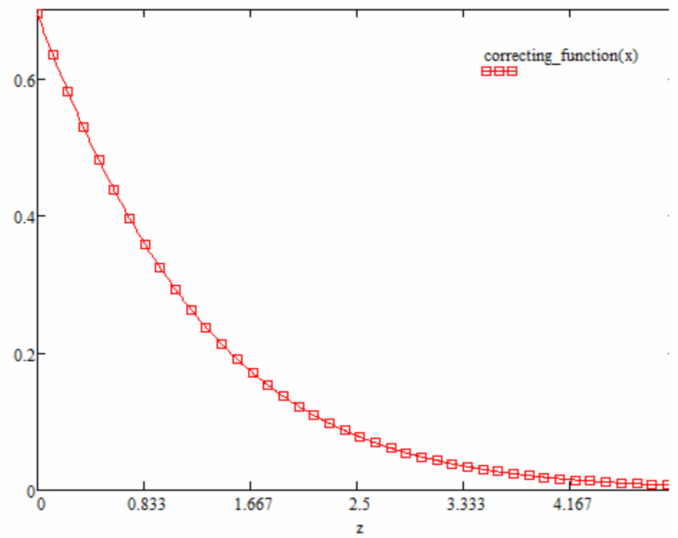

FIG. 3. EXACT VALUES OF THE CORRECTION FUNCTION VERSUS THE ABSOLUTE VALUES OF THE DIFFERENCE $Z = |X - Y|$

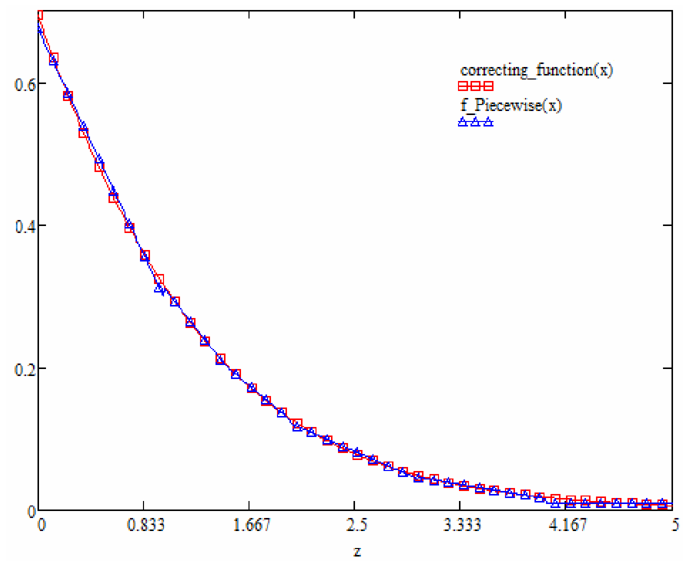

FIG. 4. COMPARISON BETWEEN THE EXACT VALUES OF THE CORRECTION FUNCTION VERSUS THE PIECEWISE LINEAR APPROXIMATION VALUES

Squares method.

From table 1 we see, that the proposed approximation is very precise on sub-ranges (Root-Mean-Square Errors (RMSE) from 0.0013 to 0.0108, correlation R between original f_{cor} and proposed from 0,991 to 0.999).

TABLE 1. NUMERICAL ESTIMATIONS FOR PROPOSED PIECEWISE LINEAR APPROXIMATION TO THE CORRECTION FUNCTION

Piecewise Linear Approximation	Sub-ranges	Slope	Inter-section	RMSE	R
$f_0(z) = a_0 \cdot z_0 + b_0$	[0; 1]	-0.3792	0.6754	0.0108	0.996
$f_1(z) = a_1 \cdot z_1 + b_1$	[1; 1.5]	-0.2229	0.5327	0.0019	0.999
$f_2(z) = a_2 \cdot z_2 + b_2$	[1.5; 2]	-0.1483	0.4213	0.0014	0.998
$f_3(z) = a_3 \cdot z_3 + b_3$	[2; 3]	-0.0773	0.2758	0.0032	0.991
$f_4(z) = a_4 \cdot z_4 + b_4$	[3; 4]	-0.0300	0.1362	0.0013	0.993

The graphical comparisons for different approximations for correction terms are presented on the plots (Fig. 5a and 5b).

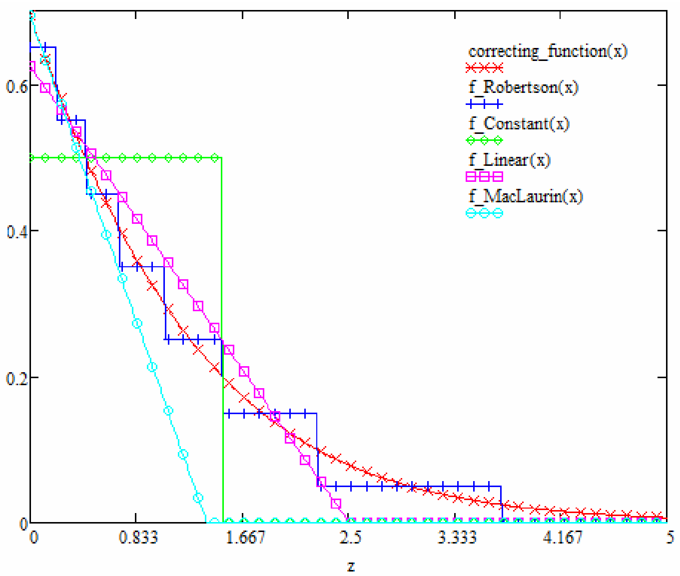

FIG. 5A. COMPARISONS BETWEEN THE CORRECTION FUNCTION APPROXIMATIONS (PART 1)

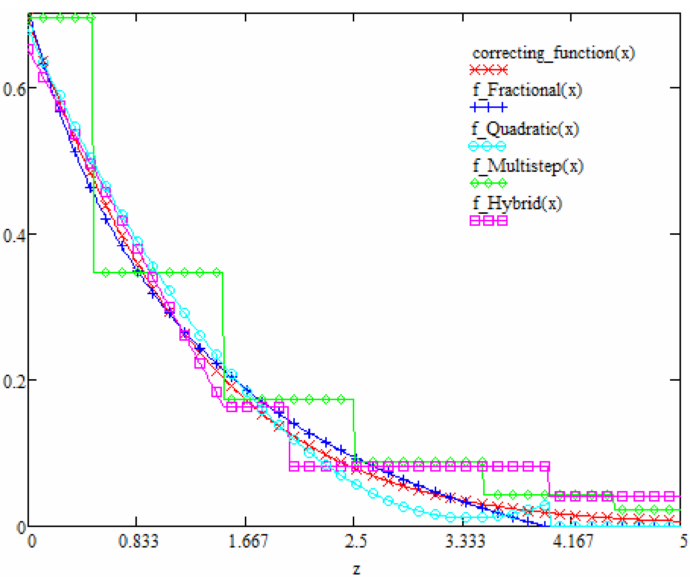

FIG. 5B. COMPARISONS BETWEEN THE CORRECTION FUNCTION APPROXIMATIONS (PART 2)

The accuracy of our correction function is verified through the comparison against different approximations in table 2 for test range [0:0.01:10], where MAE − Mean Absolute Error, SSE − Sum of Square Errors, $w = |R_i - R_{prop}| / R_i$.

TABLE 2. NUMERICAL ESTIMATIONS FOR PROPOSED PIECEWISE LINEAR APPROXIMATION AGAINST OTHER CORRECTION TERMS

Approximation	MAE	SSE	RMSE	R	*w*, %
Robertson *et al* , [10]	0.0108	0.0003	0.0170	0.9940	**0.5433**
Zhang and Yu , [14]	0.0091	0.0002	0.0135	0.9974	**0.2005**
Gross and Gulak, [15]	0.0432	0.0065	0.0807	0.8941	**11.777**
Cheng and Ottosson,[17]	0.0179	0.0008	0.0285	0.9860	**1.3590**
Talakoub *et al* , [18]	0.0342	0.0041	0.0641	0.9351	**6.8763**
Asoodeh , [19]	0.0063	0.0001	0.0089	0.9983	**0.1102**
Lim, [20]	0.0265	0.0020	0.0453	0.9756	**2.4395**
Lim , [20]	0.0227	0.0007	0.0265	0.9925	**0.6952**
Proposed Approximation	**0.0059**	**10^{-5}**	**0.0070**	**0.9994**	-

Our approximation for correction term is more accurate than other good methods, such as fractional (by 0.11 %), quadratic (by 0.2 %), Robertson lookup table (by 0.54 %) and hybrid (0.69 %) compensation functions.

So, our proposed approximation is very precise on test range (best MAE, SSE, RMSE and R) versus all other approximations. There isn't much difference between the log-MAP and the proposed PL-log-MAP algorithms.

B. Monte-Carlo BER Simulation

Turbo code with two identical RSCC with generator polynomial $(7, 5)_8$, implemented in 6 iterations, with overall rate of 1/3, and a random interleaver of size 400 bits was used in the simulation over the AWGN channel for data modulated using BPSK, stopping rule is 1 error in frame [21]. The BER performance for the log-MAP and PL-log-MAP algorithms are presented in Fig. 6. Numerical results are shown in table 3.

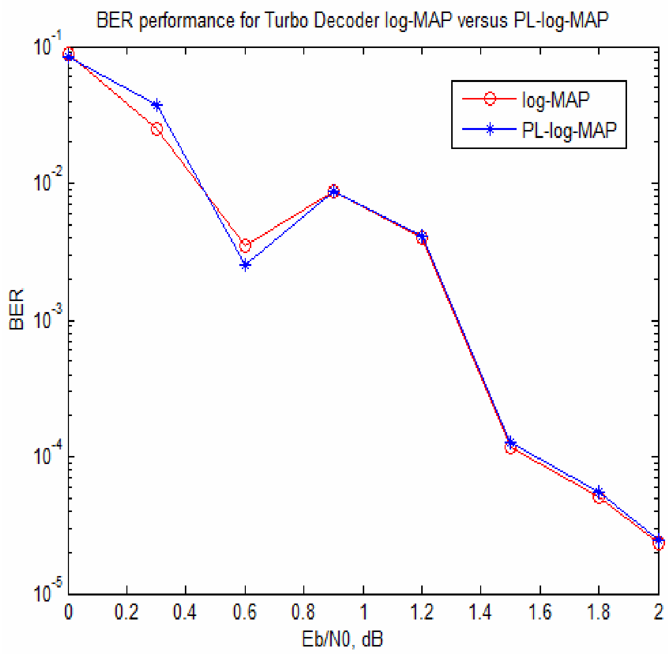

FIG. 6. BER PERFORMANCE FOR THE LOG-MAP AND PL-LOG-MAP ALGORITHMS (6 ITERATION)

TABLE 3. NUMERICAL RESULTS FOR MONTE-CARLO BER SIMULATION

Eb/N0, dB	BER, log-MAP	BER, PL-log-MAP
0.00	$8.750 \cdot 10^{-2}$	$8.,500 \cdot 10^{-2}$
0.30	$2.500 \cdot 10^{-2}$	$3.750 \cdot 10^{-2}$
0.60	$3.500 \cdot 10^{-3}$	$2.500 \cdot 10^{-3}$
0.90	$8.714 \cdot 10^{-3}$	$8.750 \cdot 10^{-3}$
1.20	$4.000 \cdot 10^{-3}$	$4.120 \cdot 10^{-3}$
1.50	$1.180 \cdot 10^{-4}$	$1.280 \cdot 10^{-4}$
1.80	$5.061 \cdot 10^{-5}$	$5.494 \cdot 10^{-5}$
2.00	$2.312 \cdot 10^{-5}$	$2.463 \cdot 10^{-5}$

The PL-log-MAP algorithm is shown to have the closest performance to the exact log-MAP solution (MAE = 0.0025, SSE = $2 \cdot 10^{-5}$, RMSE = 0.0047 and R = 0.9865).

V. CONCLUSIONS

The original log-MAP algorithm involves computationally intensive operations in order to attain the ideal performance. Neglecting the correction function, the log-MAP algorithm reduces to the simple max-log-MAP algorithm. However, this rough approximation gives a capacity loss, and hence the correction function will have to be approximated. In this paper, a new approximation to the correction function in the Jacobian logarithm was proposed and its accuracy has been compared with the exact version. The proposed piecewise linear approximation offers a simple implementation on hardware involving linear multiplications, comparators and addition operations without complex function ln(.) and exp(.).

The novel PL-log-MAP algorithm is a suboptimal log-MAP solution that achieves nearly identical performance to the log-MAP algorithm. In addition, the simulation results also show that the PL-log-MAP algorithm outperforms existing suboptimal log-MAP algorithms with different correction term approximations.

So, we conclude that the proposed PL-log-MAP algorithm is suitable for iterative turbo decoding in different digital communication systems.

REFERENCES

[1] C.E. Shannon, "*A Mathematical Theory of Communication*," Reprinted from The Bell System Technical Journal, vol 27, 1948, pp. 379–423, 623–656.

[2] C. Berrou, A. Glavieux, and P. Thitimajshima, "*Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes*," in ICC'93, Geneva, Switzerland, May 93, pp. 1064-1070.

[3] E. Boutillon, C. Douillard, and G. Montorsi, "*Iterative Decoding of Concatenated Convolutional Codes: Implementation Issues*," Proc. in IEEE, vol. 95, 2007, pp. 1201-1227.

[4] K. Sripimanwat, "*Turbo Code Applications: A Journey from a Paper to Realization*," New York, Springer, 2005, part 4.

[5] Yu.Yu. Ivanov, A.Ya. Kulyk, and I.Yu. Ivanov, "*The Turbo-codes Application in Telemedicine Systems with the Wireless Body Area Sensor Networks*," Proc. in XII Conference "Control and Management in Complex Systems", Ukraine, Vinnytsia, VNTU, 2014, pp. 206.

[6] G. Battail, "*Genetics as a Communication Process Involving Error-Correcting Codes*," Biosemiotics, vol. 1, № 1, 2005, pp. 103-144.

[7] J. Hagenauer, P. Hoeher, "*A Viterbi algorithm with soft-decision outputs and its applications*," Proc. in IEEE Global Telecommunications Conference, USA, Texas, 1989, pp. 1680–1686.

[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, 1974, pp. 284-287.

[9] J. Hagenauer, E. Offer, and L. Papke, "*Iterative Decoding of Binary Block and Convolutional Codes*," IEEE Transactions on Information Theory, vol. 42, № 2, 1996, pp. 429-445.

[10] P. Robertson, P. Villebrun, and P. Hoeher, "*Optimal and Sub-Optimal Maximum A Posteriori Algorithms Suitable for Turbo Decoding*," European Trans. on Telecommunications, vol. 8, 1997, pp. 119-125.

[11] M.R. Soleymani, Y. Gao, and U. Vilaipornsawai, "*Turbo Coding for Satellite and Wireless Communications*," New York, Kluwer Academic, 2002, 231 p.

[12] R. Morelos-Zaragoza, "*The Art of Error Correction Coding*," 2nd Edition, Chippenham, John Wiley & Sons, Ltd, 2006, 278 p.

[13] T.K. Moon, "*Error Correcting Coding: Mathematical Methods and Algorithms*," New Jersey, John Wiley & Sons, Ltd, 2005, 800 p.

[14] L. Zhang, and S.-Z. Yu, "*A Simplified log-MAP Turbo Decoder by Fitting Method*," Proc. in IEEE International Conference on Advanced Communication Technology, vol. 2, 2005, pp. 854–857.

[15] W.J. Gross, and P.G. Gulak, "*Simplified MAP Algorithm Suitable for Implementation of Turbo Decoders*," Electronic Letters, IET Journal, vol. 34, 1998, pp. 1577-1578.

[16] M.C. Valenti, "*An Efficient Software Radio Implementation of the UMTS Turbo Code*", Proc. in IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRS), USA, San Diego, 2001, pp. 108-113.

[17] J.F. Cheng, and T. Ottosson, "*Linearly Approximated log-MAP Algorithms for Turbo Decoding*," Proc. in IEEE Vehicular Technol. Conf. (VTC), USA, Boston, 2000, pp. 2252-2256.

[18] S. Talakoub, L. Sabeti, B. Shahrrava, and M. Ahmadi, "*A Linear log-MAP Algorithm for Turbo Decoding and Turbo Equalization*," IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, vol. 1, 2005, pp. 182-186.

[19] S. Asoodeh, "*A New Constructive Approximation in Log-Map Turbo Decoder*," 7th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications, UK, Cambridge, 2008, pp. 220-224.

[20] L.L. Lim, and D.W.G. Lim, "*Hybrid Log-MAP Algorithm for Turbo Decoding Over AWGN Channel*," The Seventh International Conference on Wireless and Mobile Communications (ICWMC), 2011, pp. 211-214.

[21] Yu.Yu. Ivanov, A.Ya. Kulyk, S.G. Krivogubchenko, and D.S. Krivogubchenko, "*The Software Implementation of Turbo Decoder With Metric Processing by MAP Algorithm*," Research Bulletin SWorld "Modern scientific research and their practical application", Ukraine, Odessa, vol. J11307, 2013, pp. 53-58.

Yuriy Y. Ivanov has received the master of engineering degree (2013) at the Vinnytsia National Technical University.

He work with his engineering degree thesis in postgraduate school. Also he has won the President Fellowship for the postgraduate students in 2014. At the moment he is an author of 35 scientific publications and 25 patents. His research interests include information theory, error-correcting coding, bioinformatics, special mathematics sections, optimization tasks and numerical analysis software.

Alexander N. Romanyuk has receieved an engineering degree in 1986 and Ph.D. degree in 2008. At the moment he is an author of 400 scientific publications, including 6 monographs and 60 patents.

He was appointed on the position of Corresponding member of Telecommu-nications and Computer Science Academy in Russia. Within recent years he was supervisor of 14 scientific projects. He created 2 devices (graphics accelerators) for the industry. He has tight scientific relations with scientists in Russia, USA, Canada, Poland, Roumania and Sweden. His research interests include methods and tools for interpolation and parallelization of the computational process for computer graphics, rendering.

Anatoliy Ia. Kulyk has received an engineering degree in 1992 and Ph.D. degree in 2009 at the Vinnytsia National Technical University.

He was joined the Vinnytsia Technical University in 1984, where he is currently professor at the Faculty of Automatics, Electronics and Computer Systems. He was an engineer at the Institute of Control Problems (Academy of Sciences, Moscow, USSR) in 1984. He is an author of more than 110 scientific publications and 40 patents. His research interests include information transmission, error-correcting and adaptive coding, signal processing.

Oleg V. Stukach is currently Ph.D, Professor, Member of the European Microwave Association, IEEE Senior Member, Chair of the Tomsk IEEE Chapter.

He is an author of more than 210 scientific publications and 5 patents. Also he has won the Tomsk Region Prize in the Field of Science and Education. His research interests include mathematical modeling in radioelectronics, theory of control, differential-Taylor transformations theory, software for new methods of the analog devices synthesis.