ELectronics EXpress

LETTER

IEICE Electronics Express, Vol.15, No.1, 1–8

An improved implementation of MAX* operation for Turbo decoder

Zhen Liu[1,2a)], **Tianchun Ye**[1], **Bin Wu**[1], **and Xiao Ge Zhu**[1,2]

[1] *Institute of Microelectronics of Chinese Academy of Science, Beijing, China*

[2] *University of Chinese Academy of Sciences*

a) *liuzhen@ime.ac.cn*

Abstract: An improved method to approximate max* operation with Taylor Series for Turbo decoder is presented. Multiple expansion points are adapted to improve the BER performance. The parameter δ is introduced to determine suitable expansion points. The simulation results show that the proposed method with three expansion points when δ is set to 0.025 has almost identical performance compared with ideal Log-MAP algorithm and outperforms both PWL method and the original Maclaurin series method. The architecture of proposed method is also presented for implementation. Compared with PWL methods, the proposed scheme has reduced computational complexity and is feasible for hardware implementation. Besides, the architecture of proposed method keeps identical for different number of expansion points.

Keywords: Turbo decoder, max* operation, Taylor Series

Classification: Integrated circuits

References

[1] C. Berrou, *et al.*: "Near Shannon limit error-correcting coding and decoding: Turbo-codes," IEEE International Conference on Communications (1993) 1064 (DOI: 10.1109/icc.1993.397441).

[2] A. J. Viterbi: "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," IEEE J. Sel. Areas Commun. **16** (1998) 260 (DOI: 10.1109/49.661114).

[3] P. Robertson, *et al.*: "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," IEEE International Conference on Communications (1995) 1009 (DOI: 10.1109/icc.1995.524253).

[4] W. J. Gross and P. G. Gulak: "Simplified MAP algorithm suitable for implementation of turbo decoders," Electron. Lett. **34** (1998) 1577 (DOI: 10.1049/el:19981120).

[5] J. F. Cheng and T. Ottosson: "Linearly approximated log-MAP algorithms for turbo decoding," IEEE 51st Vehicular Technology Conference Proc. (2000) 2252 (DOI: 10.1109/vetecs.2000.851673).

[6] H. Wang, *et al.*: "Improved Log-MAP decoding algorithm for Turbo-like codes," IEEE Commun. Lett. **10** (2006) 186 (DOI: 10.1109/LCOMM.2006.1603379).

[7] S. Talakoub, *et al.*: "An improved Max-Log-MAP algorithm for Turbo decoding and Turbo equalization," IEEE Trans. Instrum. Meas. **56** (2007) 1058

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

IEICE Electronics Express, Vol.15, No.1, 1–8

(DOI: 10.1109/TIM.2007.894228).
[8] S. Papaharalabos, *et al.*: "On optimal and near-optimal turbo decoding using generalized max operator," IEEE Commun. Lett. **13** (2009) 522 (DOI: 10.1109/LCOMM.2009.090537).
[9] H. Samadian and A. M. Nia: "Linear-constant log-map, a fast accurate algorithm for map decoding," J. Franklin Inst. **349** (2012) 1721 (DOI: 10.1016/j.jfranklin.2012.02.003).
[10] V. H. Rohit, *et al.*: "A composite sub-optimal approach for hardware implementation of Turbo decoder," IEEE International Conference on Communications (2013) 4515 (DOI: 10.1109/icc.2013.6655279).
[11] Z. Liu, *et al.*: "Improved Turbo decoding with multivariable Taylor Series expansion," IEEE Commun. Lett. To be published (DOI: 10.1109/LCOMM.2017.2705643).

1 Introduction

Turbo code is firstly proposed by Berrou and Glavieux in 1993 [1]. It has been widely adapted in modern communication systems for its excellent performance. The maximum a *posteriori* (MAP) algorithm is adapted as the decoding algorithm for component decoders and is commonly performed in logarithmic domain in which multiplications and exponential operations can be transformed to log exponential sum. The kernel of computation involved in decoding can resort to $\max^*$ operation. It's well known that the implementation of Turbo decoder is quite area and resource consuming for the entire baseband receiver. So it's necessary to exploit the method to simplify $\max^*$ operation.

With the application of Jacobian logarithm, the $\max^*$ operation can be expressed by the sum of maximum variable and a correction term. Several methods are proposed to approximate the $\max^*$ operation with reduced complexity, including Max-Log-MAP [2], LUT approach [3], Constant-Log-MAP [4], linear-Log-MAP [5]. An exponential approximation is proposed in [6]. The correction term approximated by the first order Maclaurin Series expansion around zero was proposed in [7] for the first time. In [8], $\max^*$ operation was dealt with piecewise linear (PWL) approximation terms with different r values. The performance will be improved as the number of PWL terms increases. However, the cost is greatly increased complexity. The combination of linear approximation and constant is proposed in [9] while the method combining LUT and linear approximation is proposed in [10]. In [11], a method based on multivariable Taylor Series for n-input ($n \geq 4$) $\max^*$ operation is proposed.

Inspired by the method proposed in [7] and [11], we extend the method with multiple expansion points to further improve the approximation precision and BER performance. Besides, the principle to select expansion points is also proposed. We found that the PWL method proposed in [8] can also be expressed by proposed method with corresponding expansion points. Compared with the PWL methods, the proposed method presents lower computational complexity and simpler structure for implementation. Besides, the computational complexity of proposed method is almost unchanged as the number of the expansion points increases.

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

IEICE Electronics Express, Vol.15, No.1, 1–8

This remainder of this paper is organized as follows. Section 2 briefly reviews Turbo codes and the method proposed in [7]. The proposed method with multiple expansion points is also presented in this section. The simulation results of proposed method are presented in Section 3. The architecture for implementation of proposed method is presented and compared with PWL method in section 4. Finally, conclusive remarks are presented in Section 5.

2 Proposed method for 2-input max* operation

2.1 Existing method with Taylor Series

In Log-MAP algorithm, for radix-2 case, the forward recursion of state metric $\alpha_k(s)$ can be computed as

$$\alpha_k(s) = \max{}^*(\alpha_{k-1}(s'_0) + \gamma_k(s'_0, s), \alpha_{k-1}(s'_1) + \gamma_k(s'_1, s)) \tag{1}$$

where s'_0 and s'_1 represent the two possible predecessor states of s. The backward recursion can be expressed in similar way.

Obviously, the computation of state metric can resort to max* operation with two variables. Applied with Jacobian logarithm, max* operation yields

$$\begin{aligned} \max{}^*(x_1, x_2) &= \log(e^{x_1} + e^{x_2}) \\ &= \max(x_1, x_2) + \log(1 + e^{-|x_1 - x_2|}) \\ &= \max(x_1, x_2) + f_c(\delta) \end{aligned} \tag{2}$$

where $f_c(\delta)$ is thought as a correction term, and δ is equal to $|x_1 - x_2|$ which represents the absolute value of difference between x_1 and x_2.

The Taylor Series can be used to approximate a nonlinear function with a linear function. It was firstly introduced to Turbo decoding in [7]. The approximation with Taylor Series at point $x = x_0$ can be expressed as

$$f_c(x) \approx \widehat{f_c}(x) = f_c(x_0) + f'_c(x_0)(x - x_0) \tag{3}$$

where $\widehat{f_c}(x)$ is the linear approximation and $f'_c(x_0)$ is the first order derivative value of $f_c(x)$ at $x = x_0$.

The method proposed in [7] adapts only one expansion point and expands around zero. The approximation neglects these orders greater than one and yields

$$\max{}^*(x_1, x_2) \approx \max(x_1, x_2) + \max\left(0, \log 2 - \frac{1}{2}|x_1 - x_2|\right) \tag{4}$$

For simplicity, the method proposed in [7] is named with 'MS-Log-MAP' in the rest paper.

2.2 Proposed method with multiple expansion points

The approximation with only one point is suitable for the region close to expansion point. For the region that far away from the expansion point, the mismatch between the approximation and the exact value will be significant. In order to reduce this mismatch, we come up with the idea that approximate the correction term with multiple expansion points which will generate multiple linear approximations. Since the function of $f_c(x)$ is concave, so the linear approximation $\widehat{f_c}(x)$ is always not greater than the exact value and the approximation can be expressed as

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

$$\widehat{f}_c(x_2 - x_1) = \max(0, k_1 * |x_1 - x_2| + C_1, \ldots, k_i * |x_1 - x_2| + C_i) \tag{5}$$

where k, l and C represent the corresponding coefficients at different expansion point. And i is the number of expansion points. Obviously, the approximations are always not greater than the exact values.

When it comes to the selection of expansion points, an auxiliary parameter δ is introduced to determine the expansion points. The parameter δ can be expressed as follows

$$\delta = \max_x |f_c(x) - \widehat{f}_c(x)| \tag{6}$$

where δ indicates the maximum error between the exact value $f_c(x)$ and the linear approximation $\widehat{f}_c(x)$.

As is known, the expansion point is also the tangent point of $f_c(x)$. The steps to determine the expansion points are summarized as follows:
Step 1: Find the point $(x, \widehat{f}_c(x))$ that satisfies $\widehat{f}_c(x) = f_c(x) - \delta$. These points are points of intersection which are between approximation and axis, inter approximations. We start the searching of these points from the point of intersection between $f_c(x)$ and axis y. Obviously, the initial point is $(0, \log(x) - \delta)$ on axis y.
Step 2: Compute the tangent point $(x_0, f_c(x_0))$. According to the point $(x, \widehat{f}_c(x))$ from Step 1, we can compute the tangent point of $(x_0, f_c(x_0))$ as follows

$$\begin{aligned} f_c(x) - \delta &= f_c'(x_0) * (x - x_0) + f_c(x_0) \\ \Rightarrow \log(1 + e^{-x}) - \delta &= \log(1 + e^{-x_0}) - \frac{1}{1 + e^{x_0}} * (x - x_0) \end{aligned} \tag{7}$$

where x is the point of intersection we got in Step 1, and x_0 is the tangent point to be determined. This is a transcendental equation. So the numerical result rather than analytical result is desired. We scan the x_0 from the point of intersection x to 4 with step size 0.01, and choose the point that $\widehat{f}_c(x)$ is closest to $f_c(x)$ as the tangent point x_0.
Step 3: Compute the linear approximation $\widehat{f}_c(x)$ at expansion point x_0. With the tangent point computed in last step, the approximation can be computed as Eq. (3).
Step 4: Repeat Step 1 to Step 3 until the error $f_c(x) - \widehat{f}_c(x)$ is not greater than δ at the point of intersection between $\widehat{f}_c(x)$ and axis x. Otherwise repeat Step 1 to Step 3.

The parameter δ is set to be 0.025 in our design and the determined expansion points are 0.45, 1.46 and 2.97, the correction term can be approximated as

$$\begin{aligned} f_c(x_2 - x_1) \approx \max(0, &\, 0.6685 - 0.3894 * |x_2 - x_1|, \\ &\, 0.4840 - 0.1885 * |x_2 - x_1|, \\ &\, 0.1950 - 0.0488 * |x_2 - x_1|). \end{aligned} \tag{8}$$

However, if Eq. (8) is implemented directly, the complexity will increase greatly with the increased number of expansion points which involves more max operations, multiplications and additions. If the expansion point is determined, then only the linear approximation at the selected expansion point is necessary to compute and the max operation can be removed. The expansion point can be selected according to $|x_2 - x_1|$, so the Eq. (8) can be reformulated as

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

IEICE Electronics Express, Vol.15, No.1, 1–8

$$f_c(x_2 - x_1) \approx \begin{cases} 0.6685 - 0.3894 * |x_2 - x_1|, & for\ 0 < |x_2 - x_1| < 0.92 \\ 0.4840 - 0.1885 * |x_2 - x_1|, & for\ 0.92 < |x_2 - x_1| < 2.07 \\ 0.1950 - 0.0488 * |x_2 - x_1|, & for\ 2.07 < |x_2 - x_1| < 3.9959 \\ 0, & for\ 3.9959 < |x_2 - x_1| \end{cases} \tag{9}$$

where the expansion points are around 0.45, 1.46 and 2.97, respectively. For the region that $|x_1 - x_2|$ is above 3.9959, the correction term is taken as zero. The summary of the approximations and corresponding expansion points are shown in Table I.

Table I. The expansion points and corresponding approximation term

Range of $\lvert x_2 - x_1\rvert$	Approximation term	Expansion point
0~0.92	$0.6685 - 0.3894 * \lvert x_2 - x_1\rvert$	0.45
0.92~2.07	$0.4840 - 0.1885 * \lvert x_2 - x_1\rvert$	1.46
2.07~3.9959	$0.195 - 0.0488 * \lvert x_2 - x_1\rvert$	2.97

It's found that the PWL method in [8] can also be interpreted by the proposed method with multiple expansion points according to Eq. (5). The PWL method combines the max operation and the correction term as a whole. The PWL method removes the absolute operation, so it computes the approximations with assumption $x_2 > x_1$ and $x_1 > x_2$ separately. For PWL method with $r = 3$, it can be verified that the corresponding expansion point is around zero. And the approximation is identical with the 'MS-Log-MAP' proposed in [7] if the max value is combined with the correction term. For PWL method with $r = 4$, the corresponding expansion point is around 0.9895. When it comes to $r = 5$, two expansion points are adapted, which are around 0 and 1.6.

Fig. 1 plots the curves of different approximations and the curve of ideal correction term. Among these plotted methods, it's obvious that the proposed method has the highest accuracy while the 'MS-Log-MAP' has the largest derivation.

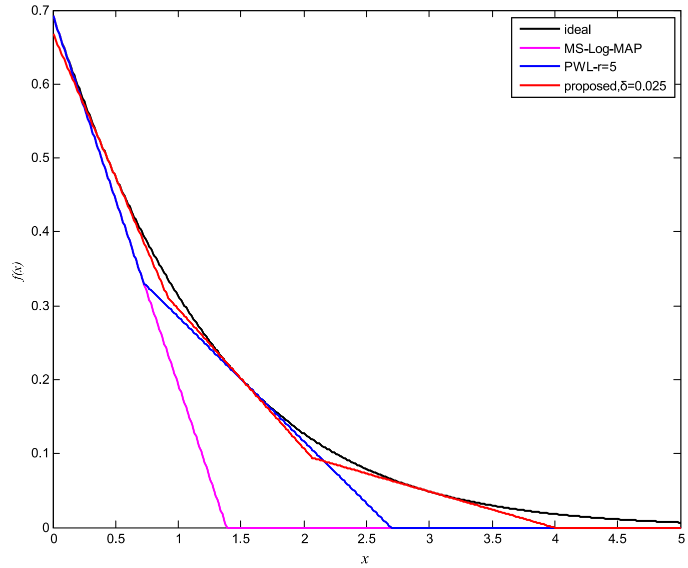

Fig. 1. The curves of different approximations of correction term.

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

On the perspective of parameter δ, we can compare the existing methods 'MS-Log-MAP' and 'PWL-$r = 5$'. The maximum error of 'MS-Log-MAP' is 0.2244. And the maximum error of 'PWL-$r = 5$' at the points of intersection are 0.0001, 0.0642 and 0.0654, respectively.

3 Numerical results

The simulation is performed for 8-state turbo codes defined in 3GPP LTE standard. The code rate is set to $1/2$, and the block size is 2048. The channel is modeled to be AWGN and BPSK modulation is considered. For decoder, the number of iterations is 8. Simulations are carried out with ideal floating point algorithms and the total number of transmitted blocks is 10^4.

Fig. 2 shows the bit error performance (BER) versus E_b/N_0, where E_b is the bit energy and N_0 is the one-sided power spectral density of AWGN channel. The curves labeled with 'Log-MAP', 'max-Log-MAP', 'LUT-Log-MAP', 'constant-Log-MAP', represent the BER performance of such a decoder that are mentioned in Section 1. Eight values of approximation term for 'LUT-Log-MAP' are stored in LUT. The curves labeled with 'MS-Log-MAP' and 'PWL-$r = 5$' are BER performance of the method proposed in [7] and [8], respectively. The 'proposed-TS-3EPs, $\delta = 0.025$' represents the method proposed with three expansion points in this paper.

As shown in Fig. 2, the 'max-Log-MAP' has the worst performance among the existing methods. The 'MS-Log-MAP' has inferior performance compared with 'constant-Log-MAP'. The proposed method has about 0.11 dB performance superior than 'MS-Log-MAP' when BER is 10^{-5}. Besides, with the magnified detailed curves, it's shown that the proposed method is slightly superior than the 'PWL-

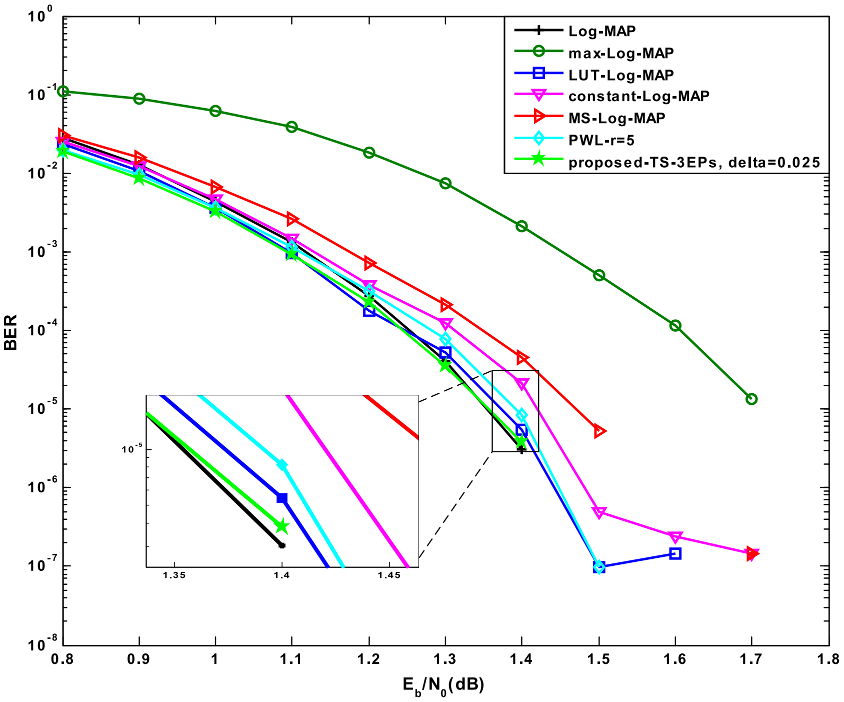

Fig. 2. BER performance comparison, N = 2048, 8 iterations, AWGN channel

© IEICE 2018
DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

$r = 5$' method among all SNR region, which has about 0.04 dB advantage when BER is 10^{-4}. The proposed method has almost the same performance with 'LUT-Log-MAP' when SNR is low and outperforms 'LUT-Log-MAP' when SNR is high. Compared with 'Log-MAP', the proposed method has identical performance.

4 Design architecture and complexity analysis

As mentioned above, as the parameter δ decreases, then the number of expansion points increases, the computational complexity of Eq. (8) will increase significantly. For PWL method, the same problem occurs as the value of r increases. As comparison, the detailed architecture of PWL method with $r = 5$ is firstly presented in Fig. 3. For $r = 5$, 6 multiplications and 6 additions are needed. Besides, 4 max operations are required. Much more computations are required for larger r value. Definitely, it's too complex for implementation.

Fig. 4 shows the detailed architecture of the proposed method according to Eq. (9). Considering simplicity and feasibility for extension, the coefficients k, l and C involved in Eq. (9) are stored in LUT. The max operation will be executed firstly for the two inputs, and the lager value will be chosen via the sign of the difference value. Then the difference value is used to addressing the LUT to read the corresponding coefficients at specified expansion point. The correction term is computed and added with the largest input finally to get the complete result.

Compared with Fig. 3, it's obvious that the proposed architecture has much lower complexity which only one max operation, one multiplication and two additions are needed. What's more, compared with the PWL method, the architecture of proposed method is consistent for different number of expansion points. So another advantage of proposed method is that the complexity keeps almost unchanged as the number of expansion points increases. The only difference is the coefficients stored in LUT. The addressing signals for LUT can be generated by bit logic according to the value of difference. The cost of the proposed method is that

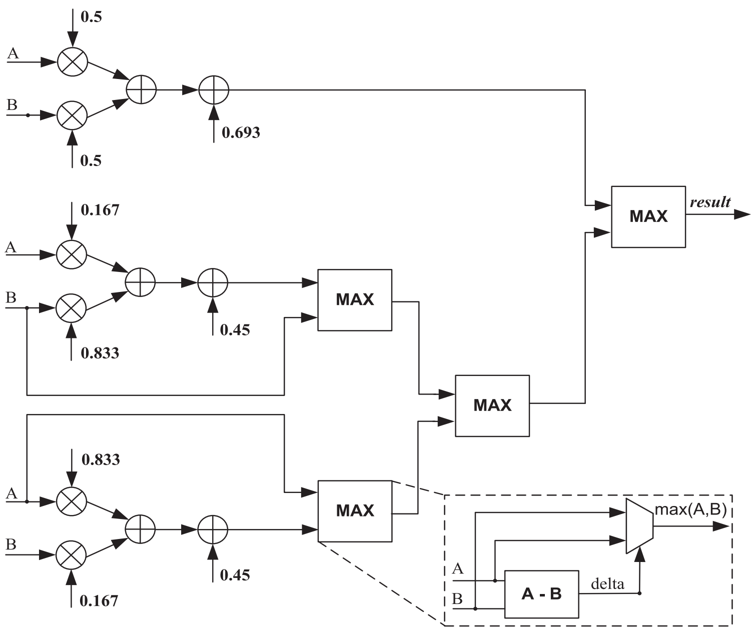

Fig. 3. Detailed Architecture for PWL-$r = 5$ method in [8]

© IEICE 2018
DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018

IEICE Electronics Express, Vol.15, No.1, 1–8

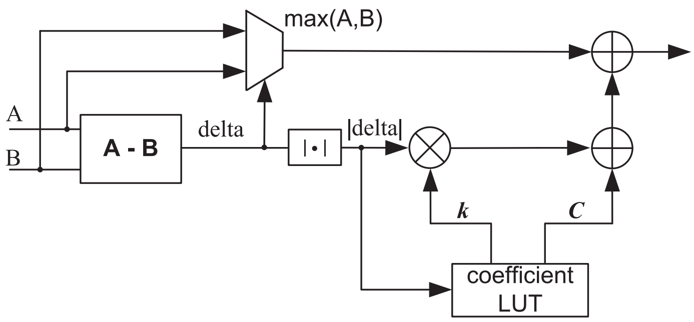

Fig. 4. Architecture of proposed method for Eq. (9)

one small LUT is needed. And three groups of coefficients are needed to be stored in the LUT in which k, l and C can be stored as a entity.

The summary of complexity comparison between existing methods is presented in Table II for two input max* operation case. Compared with the 'PWL-$r = 5$' and 'PWL-$r = 4$', the computational complexity of proposed method is greatly reduced but with slightly superior BER performance. Besides, the proposed method has comparable complexity with original 'MS-Log-MAP', which has one more multiplication while has one less addition and none shift operation.

Table II. Comparison of computational complexity for different approximations of max* operation

Algorithms	Max.	Add.	Mult.	Shift	Abs.	LUT
MS-Log-MAP	1	3	0	1	1	0
PWL-$r = 4$	3	4	4	0	0	0
PWL-$r = 5$	4	6	6	0	0	0
Proposed	1	2	1	0	1	1

5 Conclusion

The improved method based on Taylor Series with multiple expansion points is presented in this paper. The proposed method aims to provide an improved approximation of max* operation along with excellent BER performance. Aided with parameter δ, the suitable expansion points can be determined and the corresponding approximations can be derived based on Taylor series expansion. When δ is set to be 0.025, three expansion points can be determined. With application of three expansion points, the simulation results shows that the proposed method achieves excellent BER performance which is almost identical with Log-MAP algorithm. Compared with existing methods, especially the PWL method, the proposed method has greatly reduced computational complexity and slightly superior BER performance. Besides, the computational complexity keeps almost consistent for different number of expansion points. For ASIC or FPGA implementation, the proposed method will cost reduced resources for max* operation.

DOI: 10.1587/elex.14.20171145
Received November 20, 2017
Accepted November 28, 2017
Publicized December 13, 2017
Copyedited January 10, 2018