LETTER

Reduced-Complexity Belief Propagation Decoding for Polar Codes*

Jung-Hyun KIM[†a)], **Inseon KIM**[†b)], **Gangsan KIM**[†c)], ***Nonmembers*, *and*** **Hong-Yeop SONG**[†d)], ***Member***

SUMMARY We propose three effective approximate belief propagation decoders for polar codes using Maclaurin's series, piecewise linear function, and stepwise linear function. The proposed decoders have the better performance than that of existing approximate belief propagation polar decoders, min-sum decoder and normalized min-sum decoder, and almost the same performance with that of original belief propagation decoder. Moreover, the proposed decoders achieve such performance without any optimization process according to the code parameters and channel condition unlike normalized min-sum decoder, offset min-sum decoder, and their variants.
key words: *polar codes, belief propagation decoding, approximation*

1. Introduction

Polar codes [1] are the first theoretically provable capacity-achieving error correcting codes with near-linear time encoding and decoding complexities. However, the result holds for very long code length in conjunction with successive cancelation (SC) decoder [1]. To improve the performance of polar codes in the finite length regime, various decoders have been proposed in the literature. Based on successive cancellation (SC) decoder, authors of [2] introduced a successive cancelation list (SCL) decoder. The SCL decoder offers the performance close to that of maximum-likelihood (ML) decoder. Furthermore, it was shown in [3] that, using cyclic redundancy check (CRC), polar codes with SCL decoder even outperform more than some turbo codes. However, due to the serial processing nature of SC decoder, all its variants would suffer a low decoding throughput and high latency.

In contrast, belief propagation (BP) decoder proposed in [1] has the intrinsic advantage of parallel processing which is attractive for low-latency applications. Moreover, BP decoder, unlike SC decoder and its variants, provides soft-outputs that are necessary for iterative detectors for inter-symbol interference or multiple-antenna channels. Recenly, to reduce the complexity of BP decoder, authors of [4], [5] proposed approximate polar BP decoder using min-sum (MS) algorithm and normalized MS (also known as scaled MS) algorithm, respectively. In this paper, we propose three effective approximate polar BP decoders using Maclaurin's series [6], piecewise linear function [7], stepwise linear function. To the best of our knowledge, this is the first work that applies these approximation techniques to polar decoder.

2. Preliminaries

The BP decoder for polar codes can be represented by a factor graph [1]. Given the code length $n = 2^m$ and information length k, the binary source bits consists of k information bits and $n - k$ frozen bits. The codeword $\mathbf{x}$ with code rate k/n can be obtained as follows:

$$\mathbf{x} = \mathbf{u} \cdot G_n,$$

where $G_n = F^{\otimes m}$ is the generator matrix, $F^{\otimes m}$ is the m-th Kronecker power of $F = \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$, and $m = \log_2 n$.

A factor graph of polar code of length $n = 2^m$ consists of m stages and $(m + 1)n$ nodes. Figure 1 shows a factor graph representation of polar code of length $n = 2^3 = 8$. The message passing on a unit graph is shown in Figure 2. The messages are iteratively updated as follows:

$$\begin{aligned} L^t_{i,j} &= f(L^{t-1}_{i,j+1}, L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}), \\ L^t_{i+n/2^j,j} &= L^{t-1}_{i+n/2^j,j+1} + f(L^{t-1}_{i,j+1}, R^t_{i,j}), \\ R^t_{i,j+1} &= f(R^t_{i,j}, L^{t-1}_{i+n/2^j,j+1} + R^t_{i+n/2^j,j}), \end{aligned}$$

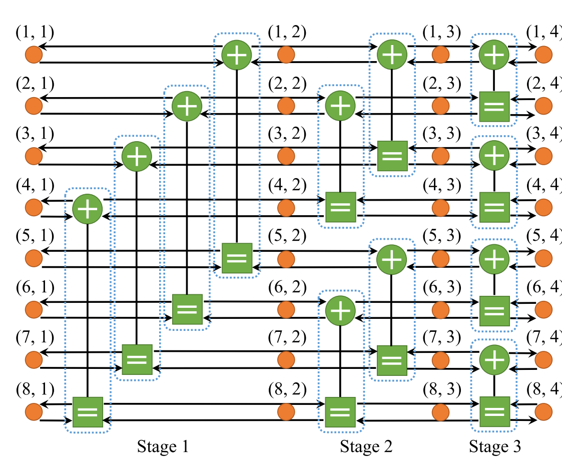

Fig. 1 Factor graph representation of polar code with code length 8.

Manuscript received March 25, 2017.
†The authors are with the Yonsei University, Korea.
*This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.2016-0-00181, Development on the core technologies of transmission, modulation and coding with low-power and low-complexity for massive connectivity in the IoT environment).
a) E-mail: jh.kim06@yonsei.ac.kr
b) E-mail: is.kim@yonsei.ac.kr
c) E-mail: gs.kim@yonsei.ac.kr
d) E-mail: hysong@yonsei.ac.kr (Corresponding author)
DOI: 10.1587/transfun.E100.A.2052

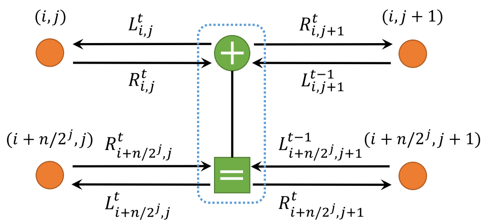

Fig. 2 Message passing in a unit graph.

$$R^t_{i+n/2^j,j+1} = R^t_{i+n/2^j,j} + f(L^{t-1}_{i,j+1}, R^t_{i,j}),$$

where $L^t_{i,j}$ and $R^t_{i,j}$ represent logarithmic likelihood ratio (LLR)-based right-to-left and left-to-right propagation messages for the i-th bit in the j-th stage during the t-th iteration, respectively, and $f(x, y) = \log\left(\frac{1+e^x e^y}{e^x+e^y}\right)$.

By Jacobian approach, the $f(x, y)$ can be rewritten as follows [8]:

$$\begin{aligned} f(x, y) &= \mathrm{sgn}(x) \cdot \mathrm{sgn}(y) \cdot \min(|x|, |y|) \\ &\quad + g(|x + y|) - g(|x - y|), \end{aligned} \tag{1}$$

where $g(z) = \log(1 + e^{-z})$.

The MS decoder, normalized MS decoder, and offset MS decoder respectively approximate (1) as follows [9]:

$$\begin{aligned} f_{MS}(x, y) &= \mathrm{sgn}(x) \cdot \mathrm{sgn}(y) \cdot \min(|x|, |y|), \\ f_{nMS}(x, y) &= \mathrm{sgn}(x) \cdot \mathrm{sgn}(y) \cdot \min(|x|, |y|)/\alpha, \\ f_{oMS}(x, y) &= \mathrm{sgn}(x) \cdot \mathrm{sgn}(y) \cdot \max(0, \min(|x|, |y|) - \beta), \end{aligned}$$

where $\alpha > 1$ is the normalized factor and $\beta > 0$ is the offset factor. The values of the factors should be optimized according to the code parameters and channel condition.

3. Reduced-Complexity Belief Propagation Decoding for Polar Codes

3.1 Approximate Belief Propagation Using Maclaurin's Series

In general, the Maclaurin's series expansion consists of infinite number of terms. For $g(z)$ in (1), the expansion is $g(z) = \sum_{n=0}^{\infty} \frac{g^{(n)}(0)}{n!} z^n$. Therefore, we can approximate the function $g(z)$ as a summation of a finite number of terms. For example, the 4th order approximation is $g(z) \approx \log(2) - \frac{1}{2}z + \frac{1}{8}z^2 - \frac{1}{192}z^4$. Figure 3 shows the values of non-approximate function and 1st, 2nd, 4th, 6th order approximate functions of $g(z)$. In the figure, the differences between the original function and the approximate functions using Maclaurin's series increase with increasing z. Moreover, since both z and $g(z)$ always have non-negative values in (1), we propose an approximate function for polar BP decoder as follows:

$$g(z) = \begin{cases} \log(2) - z/2 + z^2/8 - z^4/192 & \text{if } 0 \leqslant z \leqslant 2.6, \\ 0 & \text{otherwise.} \end{cases}$$

We note that the 0th order approximation is the same with MS decoding and a similar approach for low density parity check (LDPC) codes using the 1st order approximation is proposed in [10].

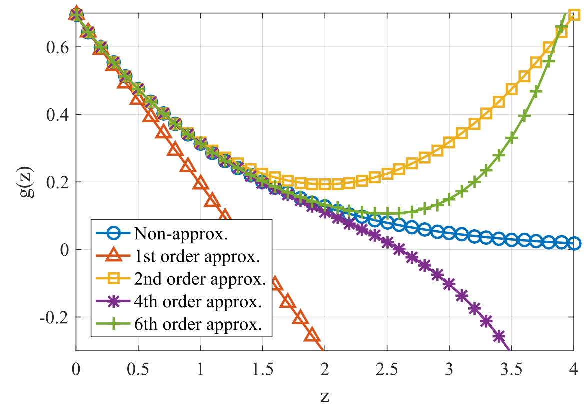

Fig. 3 The values of non-approximate function and 1st, 2nd, 4th, 6th order approximate functions of $g(z)$ using Maclaurin's series.

3.2 Approximate Belief Propagation Using Piecewise Linear Function

Based on [7], we can design some functions which offer almost perfect match to the original function using only linear functions. From this observation, we propose an approximate polar BP decoder using the following piecewise linear approximate function:

$$g(z) = \begin{cases} 0.68 - 0.38z & \text{if } 0 \leqslant z < 1, \\ 0.49 - 0.19z & \text{if } 1 \leqslant z < 2, \\ 0.21 - 0.05z & \text{if } 2 \leqslant z < 4, \\ 0 & \text{otherwise.} \end{cases}$$

As a special case of piecewise linear approximation, we can also consider an approximation using constant functions. This is also known as lookup table approximation. We propose another approximate polar BP decoder using the following stepwise linear approximate function:

$$g(z) = \begin{cases} 0.6 & \text{if } 0 \leqslant z < 2, \\ 0 & \text{otherwise.} \end{cases}$$

Figure 4 shows that the piecewise linear approximate function provides almost perfect match to the non-approximate function. In the next section, we will show that, interestingly, a polar decoder using the stepwise linear approximate function achieve the performance very close to that of the original BP decoder in spite of the difference between the stepwise linear approximate function and the original function.

We note that similar approximation approaches for LDPC codes are proposed in [11]–[13].

4. Simulation Results

We present simulation results comparing the performance of

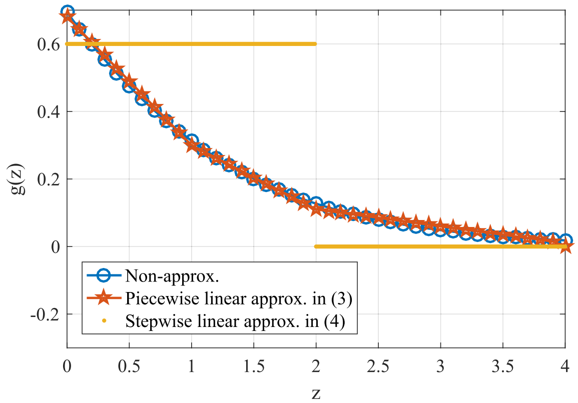

Fig. 4 The values of non-approximate function, stepwise linear approximate function, and piecewise linear approximate function of $g(z)$.

the proposed approximate polar BP decoders using Maclaurin's series, piecewise linear function, and stepwise linear function to that of the existing polar BP decoders; original BP decoder (also known as sum-product decoder) [1], MS decoder [4], and normalized MS decoder [5]. In the remaining of the paper, we denote them mMS, pMS, sMS, SP, MS, and nMS, respectively.

For the simulation, we use polar codes with code length 512, 1024 and code rate 1/2, 1/3, 2/3. The maximum number of iterations for all the decoders is set to 200. In particular, for nMS, we use $\alpha = 1.07$ which is an optimized value for a polar code with code length 512 and code rate 1/2. Figures 5–10 show the frame error rate (FER) versus E_b/N_0 of various polar BP decoders using binary phase shift keying (BPSK) modulation over an additive white Gaussian noise (AWGN) channel.

We first give the simulation results of polar codes with code length 512 in Figs. 5–7. Figure 5 shows the result with code rate 1/2. In the figure, the BP decoders have almost the same performance except for MS. In particular, MS suffers the performance degradation in low E_b/N_0 region. Figure 6 shows the result with code rate 1/3. We can see that the performance of nMS and sMS is slightly inferior to that of other BP decoders except for MS. Figure 7 shows the result with code rate 2/3. In this case, nMS has the worst performance in high E_b/N_0 region. On the other hand, the proposed decoders, mMS, pMS, and sMS, have the performance very close to that of SP.

Now, we give the simulation results of polar codes with code length 1024 in Figs. 8–10. Figure 8 shows the result with code rate 1/2. In low E_b/N_0 region, MS has the worst performance and, in high E_b/N_0 region, nMS has the worst performance even though it was optimized for code rate 1/2. From the result, we can expect that the optimization of α for nMS is affected by the code length as well as the code rate. Figure 9 shows the result with code rate 1/3. Similar to the case of code length 512, all the BP decoders except for MS have almost the same performance. Finally, Fig. 10 shows the result with code rate 2/3. In this case, nMS performs significantly worse. Although MS is the most approximate version, it has better performance than nMS in middle and high E_b/N_0 ranges. Comparing with the case of code length 512, the performance gap between our approximation schemes and existing approximation schemes is also increased.

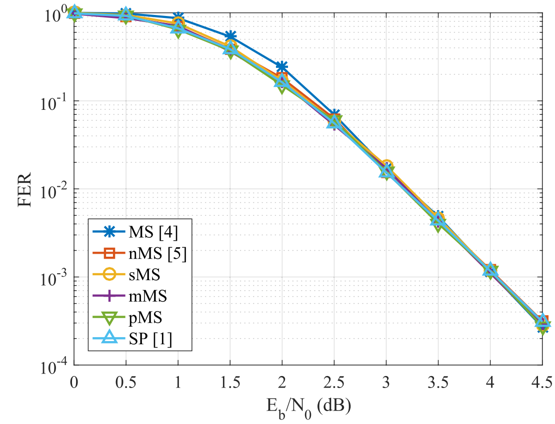

Fig. 5 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 512 and code rate 1/2.

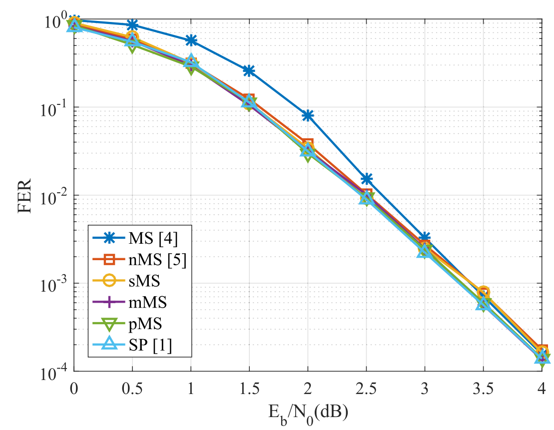

Fig. 6 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 512 and code rate 1/3.

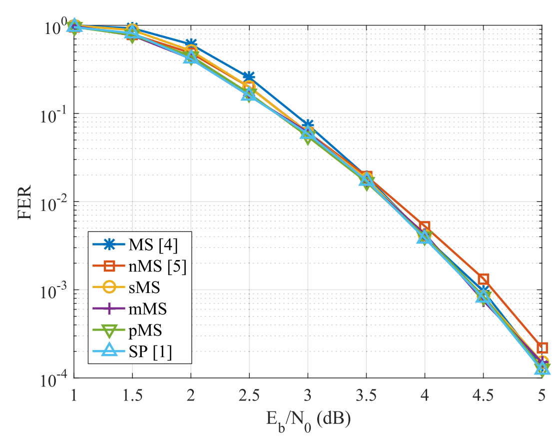

Fig. 7 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 512 and code rate 2/3.

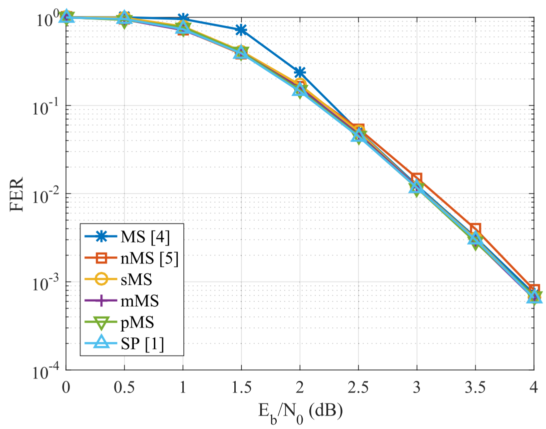

Fig. 8 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 1024 and code rate 1/2.

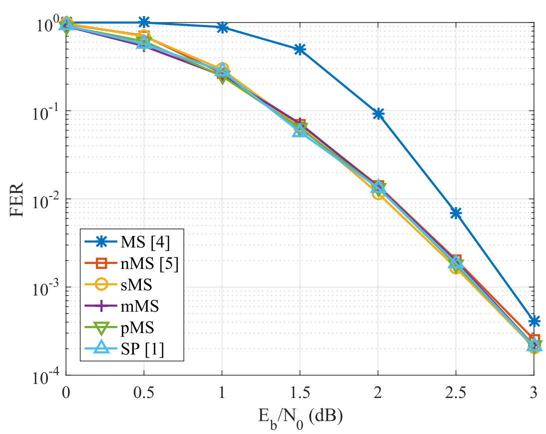

Fig. 9 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 1024 and code rate 1/3.

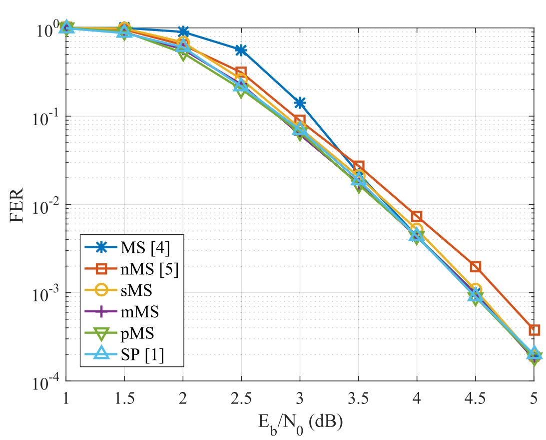

Fig. 10 FER vs. E_b/N_0 performance curve for various polar BP decoders with code length 1024 and code rate 2/3.

5. Conclusions

In this paper, we proposed three effective approximate BP decoders for polar codes using Maclaurin's series, piecewise linear function, and stepwise linear function. Simulation results show that the proposed decoders have the better performance than that of existing approximate polar BP decoders. Moreover, the proposed decoders achieve such performance without any optimization process according to the code parameters and channel condition. Therefore, it is anticipated that our approximate polar BP decoders could be a good alternative to existing approximate polar BP decoders. As a future work, one may consider a polar decoder modifying or combining some of the proposed approximation schemes.

References

[1] E. Arikan, "Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels," IEEE Trans. Inf. Theory, vol.55, no.7, pp.3051–3073, July 2009.

[2] I. Tal and A. Vardy, "List decoding of polar codes," IEEE Trans. Inf. Theory, vol.61, no.5, pp.2213–2226, May 2015.

[3] K. Niu and K. Chen, "CRC-aided decoding of polar codes," IEEE Commun. Lett., vol.16, no.10, pp.1668–1671, Oct. 2012.

[4] A. Pamuk, "An FPGA implementation architecture for decoding of polar codes," Proc. ICWCS, pp.437–441, Nov. 2011.

[5] B. Yuan and K.K. Parhi, "Early stopping criteria for energy-efficient low-latency belief-propagation polar code decoders," IEEE Trans. Signal Process., vol.62, no.24, pp.6496–6506, Dec. 2014.

[6] D. Zwillinger, CRC Standard Mathematical Tables and Formulae, 32th ed., CRC Press, 2012.

[7] A. Imamoto and B. Tang, "Optimal piecewise linear approximation of convex functions," Proc. WCECS, pp.1191–1194, Oct. 2008.

[8] J. Chen, A. Dholakia, E. Eleftheriou, M.P.C. Fossorier, and X. Hu, "Reduced-complexity decoding of LDPC codes," IEEE Trans. Commun., vol.53, no.8, pp.1288–1299, Aug. 2005.

[9] J. Chen and M.P.C. Fossorier, "Density evolution for two improved BP-based decoding algorithm of LDPC codes," IEEE Commun. Lett., vol.6, no.5, pp.208–210, May 2002.

[10] S. Papaharalabos and P.T. Mathiopoulos, "Simplified sum-product algorithm for decoding LDPC codes with optimal performance," IET Electron. Lett., vol.45, no.2, pp.116–117, Jan. 2009.

[11] X.-Y. Hu, E. Eleftheriou, D.-M. Arnold, and A. Dholakia, "Efficient implementations of the sum-product algorithm for decoding LDPC codes," Proc. IEEE GLOBECOM, pp.1036–1036E, Nov. 2001.

[12] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," IET Electron. Lett., vol.37, no.2, pp.102–104, Jan. 2001.

[13] J. Li and X.D. Zhang, "Reduced-complexity belief propagation decoding for low-density parity-check codes," IET Electron. Lett., vol.44, no.3, pp.220–222, Jan. 2008.