

# Ultra-Light Decoder for Turbo Product Codes

A. Al-Dweik, *Senior Member, IEEE*, H. Mukhtar, *Member, IEEE*, E. Alsusa, *Senior Member, IEEE*, and J. Dias, *Senior Member, IEEE*

**Abstract**—This work presents a novel low complexity decoder for turbo product codes (TPCs). The new decoder, denoted as ultra-light decoder (ULD), can perform soft decision decoding without an algebraic hard decision decoder, which is the core of conventional soft decision decoders of block codes. Moreover, the unique structure of the ULD enables the design of a new approach to compute the minimum Euclidean distance (ED) at each decoding iteration. Therefore, the ULD offers significant complexity and delay reduction as compared to conventional TPC decoders. Reducing the complexity and delay will enable using codes with high code rates to improve the system spectral efficiency, or use powerful codes with low code rates to reduce the transmission power. The system bit error rate (BER) is presented for binary and  $M$ -ary modulation schemes over additive white Gaussian noise (AWGN) channels, and the coding gain is given for Rayleigh fading channels. The obtained numerical results show that the ULD offers coding gain that is comparable to conventional TPC decoders under various system and channel conditions, but with significantly lower complexity.

**Index Terms**—5G, Turbo codes, product codes, error control coding, error correction, iterative decoding, soft decision decoding, complexity reduction.

## I. INTRODUCTION

In the literature, complexity reduction of turbo product codes (TPCs) have received extensive attention as reported in [1], and the references listed therein. Generally speaking, the complexity and error performance bounds of TPCs can be obtained from [2], where it is shown that the complexity of the near-optimal soft-input soft-output (SISO) decoder of Pyndiah [3] can be considered as the upper bound, while its bit error rate (BER) is the lower bound. On the contrary, the complexity of the hard-input hard-output (HIHO) decoder [2] can be considered as a lower bound while its BER is the upper bound. The complexity of the SISO decoder [3] is mainly determined by two main parts, namely, the hard decision decoding (HDD) process that is required to generate the candidate codewords for the Chase-II decoder, and the extrinsic information calculation, which is required to produce the soft output after each decoding half-iteration. Therefore, all complexity reduction schemes summarized in [1] have mostly focused on these two parts. For example, the decoder proposed by Lu *et al.* [4] managed to reduce the number of required HDD operations to only one operation. The authors in [5] proposed a low complexity TPC decoder by modifying the HIHO decoder at the expense of a significant coding gain reduction. An efficient TPC decoder [6] is developed by utilizing a low complexity algorithm to compute the syndromes within the HDD process, which

is required to generate the candidate codewords. Computing the syndromes recursively enabled reducing the number of operations required to compute the Euclidean distance (ED). Although the number of operations required to compute ED is reduced and the HDD processes have been simplified, a major part of the HDD process is still required to be computed. Moreover, computing the syndromes recursively increases the latency of the decoder.

To the best of the authors' knowledge, there is no SISO decoding algorithm exists in the literature that does not require HDD. Therefore, this letter presents an ultra-light decoder (ULD) for TPCs where the requirement for HDD is absolutely eliminated. The proposed ULD is generally similar to the near-optimal decoder reported in [3]. However, the new decoder does not use algebraic hard-decision decoder to perform the soft decision decoding (SDD) for each row/column in the received signal matrix. More specifically, the new decoder is based on replacing the HDD at the receiver side by an encoder similar to the one used at the transmitter, which can be implemented using simple XOR operations [7]. The unique structure of the new decoder allows significant reduction of the computational complexity required for the extrinsic information calculation process, by using a highly efficient ED calculator. Therefore, the computational complexity and delay of the proposed decoder can be reduced.

## II. PROPOSED ULTRA-LIGHT TPC-DECODER (ULD)

The encoder of the proposed ULD is similar to the conventional TPC encoder [3]. Two-dimensional (2D) TPCs are constructed by serially concatenating two linear block codes  $C^i$  ( $i = 1, 2$ ). The two constituent codes  $C^i$  have the parameters  $(n_i, k_i, d_{\min}^{(i)})$  which describe the codeword length, number of information bits, and minimum Hamming distance, respectively [1]. To build a product code,  $k_1 \times k_2$  information bits are placed in a matrix of  $k_1$  rows and  $k_2$  columns. The  $k_1$  rows are encoded by code  $C^1$  and a matrix of size  $k_1 \times n_1$  is generated. Then, the  $n_1$  columns are encoded by the  $C^2$  code and a two-dimensional codeword of size  $n_2 \times n_1$  is obtained. The parameters of the product code  $C$  are  $(n_1 \times n_2, k_1 \times k_2, d_{\min}^{(1)} \times d_{\min}^{(2)})$ . The code rate which is the number of information bits divided by the codeword size is calculated as  $\zeta = \frac{k_1 \times k_2}{n_1 \times n_2}$  for regular TPCs. Since  $C^1$  and  $C^2$  are typically identical, the TPC codeword  $\mathbf{C}$  can be obtained as

$$\mathbf{C} = [\mathbf{c}_1^T \quad \mathbf{c}_2^T \quad \dots \quad \mathbf{c}_k^T] \times \mathbf{G} \quad (1)$$

where  $\mathbf{c}_i = \mathbf{m}_i \times \mathbf{G}$ ,  $\mathbf{G} \in \mathbb{B}^{k \times n}$  is the generator matrix of  $C$ ,  $\mathbb{B}$  is the set of binary numbers,  $\mathbf{c} = [m_1, m_2, \dots, m_k, p_1, p_2, \dots, p_{n-k}]$ ,  $m_i$  and  $p_j$  correspond to the information and parity bits, respectively, and  $(\cdot)^T$  denotes the transpose operation.

### A. Ultra-Light TPC Decoder

Without loss of generality, we assume that the TPC codeword  $\mathbf{C}$  is modulated using binary phase shift keying (BPSK) to produce

A. Al-Dweik and J. Dias are with the Electrical and Computer Engineering, Khalifa University, Abu Dhabi, UAE. (E-mail: {arafat.dweik, jorge.dias}@kustar.ac.ae).

H. Mukhtar is with the Department of Electrical Engineering, University of Dubai, Dubai, UAE. (E-mail: hhadam@ud.ac.ae).

E. Alsusa is with the School of Electrical and Electronic Engineering, University of Manchester, Manchester M13 9PL, U.K. (E-mail: e.alsusa@manchester.ac.uk).

the modulated TPC codeword matrix  $\mathbf{A}$ , which is transmitted over an additive white Gaussian noise (AWGN) channel. Therefore, the received matrix can be written as  $\mathbf{R} = \mathbf{A} + \mathbf{Z}$ , where  $\mathbf{Z}$  is a matrix of AWGN samples with zero mean and  $N_0/2$  variance. For a given constituent codeword  $\mathbf{c}$ , the received vector  $\mathbf{r}$  can be expressed as  $\mathbf{r} = \mathbf{a} + \mathbf{z}$ , where  $\mathbf{z} \in \mathbb{R}^{1 \times n}$  is the AWGN vector, and  $\mathbf{a}$  is the BPSK modulated version of  $\mathbf{c}$ . Then, the ultra-light SDD process can be described as follows:

- 1) Drop the last  $n - k$  elements in  $\mathbf{r}$ , which correspond to the parity bits. The new vector is denoted as  $\mathbf{r}_m$ .
- 2) Mark the indices (locations) of the least reliable  $p$  samples in  $\mathbf{r}_m$  [3]. For AWGN and Rayleigh fading channels, the reliability of the  $i$ th element is equal to  $|r_i|$ .
- 3) Compute the hard decision bits of  $\mathbf{r}_m$  such that  $\mathbf{b}_m = \frac{1}{2}(\text{sign}(\mathbf{r}_m) + 1)$ .
- 4) Generate  $2^p$  vectors (error patterns) each of which has  $k$  bits,  $\mathbf{e}_i = [e_{i1}, e_{i2}, \dots, e_{ik}]$ ,  $i \in \{1, 2, \dots, 2^p\}$ . An error pattern is a vector whose entries are all zeros except the entries marked in the previous step. The values of the marked bits will be altered between zeros and ones until all possible  $2^p$  combinations are generated.
- 5) Generate  $2^p$  different test patterns by adding each error pattern to  $\mathbf{b}_m$ ,  $\mathbf{t}_i = \mathbf{e}_i \oplus \mathbf{b}_m$ .
- 6) Each of the  $2^p$  test patterns is then encoded by computing  $\mathbf{x}_i = \mathbf{t}_i \times \mathbf{G}$  to produce  $2^p$  candidate codewords. The successful candidate codeword  $\mathbf{d}$  is the one that has the minimum squared ED (SED) to  $\mathbf{r}$ ,

$$\mathbf{d} = \arg \min_{\mathbf{t}_i} \|\mathbf{r} - \mathbf{u}_i\|^2, i \in \{1, 2, \dots, 2^p\} \quad (2)$$

where  $\mathbf{u}_i$  is obtained by mapping the elements of  $\mathbf{x}$  from  $\{0, 1\}$  to  $\{-1, +1\}$ . As it can be noted from the described algorithm, no HDD operations are needed to generate the candidate codewords. Once the first half iteration is completed, we generate soft information for each bit in  $\mathbf{d}$  to enable SDD for the next iteration. The soft information after the first half iteration is computed as [3],

$$\tilde{\mathbf{r}}(v) = \mathbf{r} + \alpha(v)\mathbf{w}(v) \quad (3)$$

where  $v$  is the half iteration number. The values of the scaling factor  $\alpha$  are different from the ones used in [3], and they are obtained by minimizing the BER at each half iteration,

$$\alpha = [0, 0.6, 0.7, 0.9, 0.9, 0.8, 0.6, 0.9].$$

The extrinsic information  $\mathbf{w}$  can be computed as

$$\mathbf{w}(v+1) = \frac{\dot{\mathbf{r}}(v) - \tilde{\mathbf{r}}(v)}{\|\dot{\mathbf{r}}(v) - \tilde{\mathbf{r}}(v)\|_\infty} \quad (4)$$

The max norm  $\|x\|_\infty = \max(|x_1|, |x_2|, \dots, |x_n|)$ ,

$$\dot{\mathbf{r}}(v) = \frac{1}{4} \left( \|\mathbf{r} - \mathbf{d}^{(2)}\|^2 - \|\mathbf{r} - \mathbf{d}^{(1)}\|^2 \right) \times \mathbf{d}^{(1)} \quad (5)$$

and  $\mathbf{d}^{(1)}$  and  $\mathbf{d}^{(2)}$  are the closest and next closest candidate codewords to  $\mathbf{r}$ , respectively. Note that in (5) all bits in  $\dot{\mathbf{r}}(v)$  have the same soft value and hence less number of comparison operations is required to compute  $\dot{\mathbf{r}}(v)$  when compared to the conventional decoder [3]. In cases where it is not possible to find  $\mathbf{d}^{(2)}$ , then

$$\mathbf{w}(v+1) = \beta(v) \times \mathbf{d}, \quad \beta \geq 0$$

where  $\beta$  for 8 half iterations,  $v \in \{1, 2, \dots, 8\}$  is given by

$$\beta = [0.2, 0.2, 0.4, 0.4, 0.6, 0.6, 0.8, 0.8]. \quad (6)$$

The scaling factor  $\beta$  is obtained using exhaustive search with the aim of minimizing the BER. The SDD and soft output generation are applied to all rows/columns in a similar fashion to the conventional SISO decoding, till the maximum number of iterations is reached.

As it can be noted from the system description, the row/column decoding process will be successful if all the errors in  $\mathbf{b}_m$  are part of the least reliable  $p$  bits. Otherwise, the transmitted codeword  $\mathbf{c}$  will never be part of the resultant candidate codewords. Therefore, some performance degradation should be expected as compared to the Chase-II decoder, where the algebraic decoding operations can overcome this problem in certain scenarios. Nevertheless, the BER performance loss is tolerable, as shown in Section III. Moreover, although the decoding process was described for the BPSK modulation, the same approach can be adopted for higher modulation orders by computing the Log Likelihood Ratios (LLRs) for each bit.

### B. ULD Complexity

As it can be noted from the ULD design, the main complexity reduction is achieved by completely eliminating the need for the algebraic hard-decision decoder, which is replaced by the encoder that can be implemented using simple XOR operations [8]. Moreover, computing the soft-output using (5) is much simpler than the conventional decoder because the condition that  $d_i^{(1)} \neq d_i^{(2)}$  is eliminated. Consequently, there is no need to keep updating  $\mathbf{d}^{(2)}$  in (5) based on the values of  $d_i^{(1)}$  and  $d_i^{(2)}$ .

Another significant complexity reduction can be achieved by noting that replacing the algebraic decoder by a systematic encoder enables the design of a new approach to compute the SED (2). In the proposed ULD, the candidate codewords are generated by encoding the test patterns. Therefore, the produced candidate codewords have the same information bits except for  $p$  bits altered based on an error pattern. Subsequently, the candidate codewords differ in known bit positions, namely, the error pattern altered bits and the parity bits. In contrast to conventional SDD, the candidate codewords are generated by decoding test patterns and the produced candidate codewords differences are unknown. Such feature enables the proposed ULD to minimize the large number of operations required to compute the SED. The new approach is described as follows.

Let  $\mathcal{D}_i$  denote the SED for each candidate codeword  $\mathbf{u}_i$ ,  $i \in \{1, 2, \dots, 2^p\}$ . The candidate codeword  $\mathbf{u}_i$  can be segmented into three disjoint parts. The first part corresponds to the information bits excluding the  $p$  least reliable locations, the second part corresponds to the  $p$  bits in the locations marked as the least reliable  $p$  bits, and finally the parity bits. The sets of indices of the first and second parts will be denoted as  $\mathbb{I}$  and  $\mathbb{Q}$ , respectively. Therefore,  $\mathcal{D}_i$  can be expressed as

$$\begin{aligned} \mathcal{D}_i &= \|\mathbf{r} - \mathbf{u}_i\|^2 \\ &= \sum_{j=1}^k (r_j - u_{i,j})^2 + \sum_{\substack{j=1 \\ j \in \mathbb{Q}}}^k (r_j - u_{i,j})^2 + \sum_{j=k+1}^n (r_j - u_{i,j})^2. \end{aligned} \quad (7)$$

TABLE I  
THE RELATIVE COMPLEXITY OF THE ULD AND [6].

	$(32,26,4)^2$		$(64,57,4)^2$		$(128,120,4)^2$		$(512,502,4)^2$	
	ULD	[6]	ULD	[6]	ULD	[6]	ULD	[6]
$p=3$	0.30	0.27	0.22	0.19	0.18	0.16	0.14	0.13
$p=4$	0.24	0.22	0.16	0.14	0.12	0.10	0.08	0.07
$p=5$	0.21	0.18	0.13	0.10	0.09	0.09	0.05	0.04

However, the first summation in (7) is independent of the candidate codeword index  $i$ , and hence we define  $\sum_{j=1, j \in \mathbb{I}}^k (r_j - u_{i,j})^2 = \sum_{j=1, j \in \mathbb{I}}^k (r_j - u_j)^2 \triangleq \mathcal{D}_{\mathbb{I}}$ , which is constant for all values of  $i$ . Therefore, for a given SDD operation, we compute  $\mathcal{D}_{\mathbb{I}}$  only once. For the second term, we need to compute the individual metrics when  $u_{i,j} = 1$  and  $-1$ , then compute the summation based on the values of the bits in the test pattern. Therefore, the second term should be computed  $2p$  times. The third term should be computed  $2^p$  times. To compare the complexity of the SED computation in the proposed decoder with the conventional approach (2), we use the number of times that the term  $(r_j - u_{i,j})^2$  is computed as a metric. For a single conventional SDD operation, the total number of computations required is  $2^p n$ , while it is  $(k - p) + 2p + 2^p (n - k)$  for the ULD. Thus, the relative complexity  $C_R$ , which is the ratio of the two terms, can be simplified to

$$C_R = \left(1 - \frac{k}{n} + \frac{k + p}{2^p n}\right) \times 100\%.$$

Table I presents  $C_R$  of the ULD as well as [6] for some of the codes reported in [3], and for typical values of  $p$ . As it can be noted from the results, the achieved complexity reduction depends on the code used and on  $p$ . Moreover, Table I shows that the ULD substantially outperforms [3], while [6] slightly outperforms the ULD, however, HDD is still required in [6]. Therefore, the ULD and [6] require almost the same number of ED operations, yet [6] requires most of the HDD operations which has to be performed  $2^p$  times for each row/column decoding.

### III. NUMERICAL EXAMPLES

The performance of the ULD is evaluated using the BER for different TPCs, and it is compared to the near-optimum SISO [3] and HIHO decoding. The BER for all systems is obtained using Monte Carlo computer simulation using BPSK and 16 quadrature amplitude modulation (QAM) over AWGN and Rayleigh fading channels. The proposed and [3] decoders are using  $p = 4$ , and the number of iterations for all decoding schemes is set to 4. Moreover, the coding gain for all cases is computed at  $\text{BER}=10^{-5}$ .

The BER of the  $\text{eBCH}(16, 11, 4)^2$  is shown in Fig. 1. As it can be noted from the figure, the ULD coding gain for the BPSK is about 5.5 dB, which is only 0.75 dB less than the coding gain of [3]. For the 16-QAM, the coding gain is about 4.7 dB, and hence it is 1.3 dB less than [3].

Fig. 2 shows the BER of the  $\text{eBCH}(32, 26, 4)^2$  TPC code. For this code, the ULD using BPSK has a coding gain of 5.54 dB, which is about 1 dB less than the coding gain of [3], while the coding gain for the 16-QAM is 5.54 dB, and hence it is 1.31 dB less than [3]. Moreover, the ULD coding gain advantage over the HIHO decoder is about 2.8 dB.

Fig. 3 and Fig. 4 show the BER of the  $\text{eBCH}(64, 57, 4)^2$  and  $\text{eBCH}(128, 120, 4)^2$  codes, respectively. For both codes, the ULD exhibited more than 2 dB of coding gain advantage over the HIHO

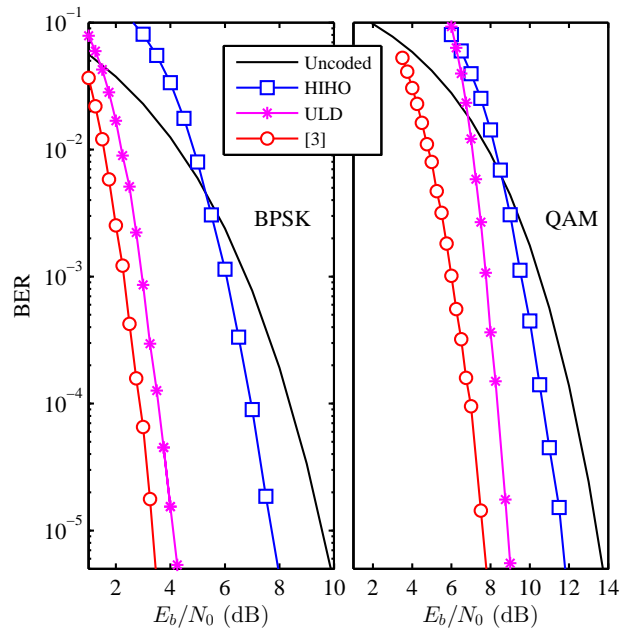


Fig. 1. BER of TPC  $\text{eBCH}(16, 11, 4)^2$ .

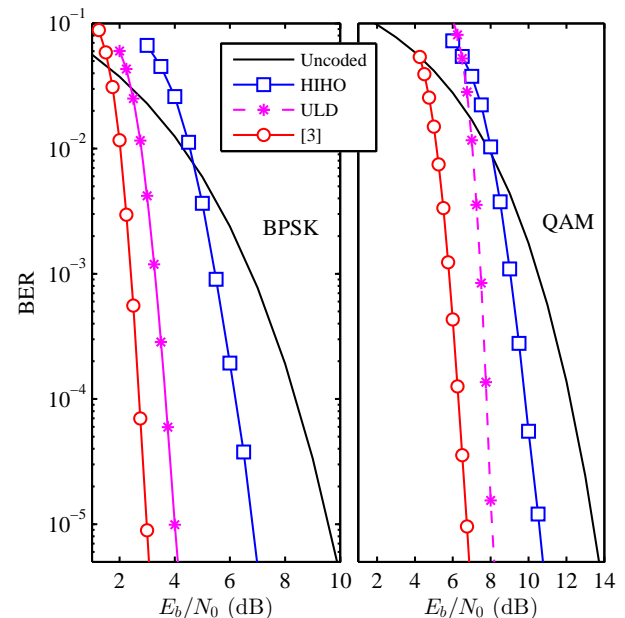


Fig. 2. BER of TPC  $\text{eBCH}(32, 26, 4)^2$ .

decoder and about 1 dB less than [3] for the BPSK. For the 16-QAM, the coding gain of the ULD is 5.54 dB, and hence it is 1.27 dB less than [3].

The coding gain for the considered codes in Rayleigh fading channels using BPSK is presented in Table II. As it can be noted from the results, the ULD coding gain degradation is 1.9 dB for the  $(16, 11, 4)^2$  code, which is about 5% less than [3]. For high code rates such as the  $(128, 120, 4)^2$ , the coding gain difference increases to 3.76 dB which is about 11.7% less than [3]. Consequently, the ULD can offer reliable BER performance in both AWGN and fading channels.

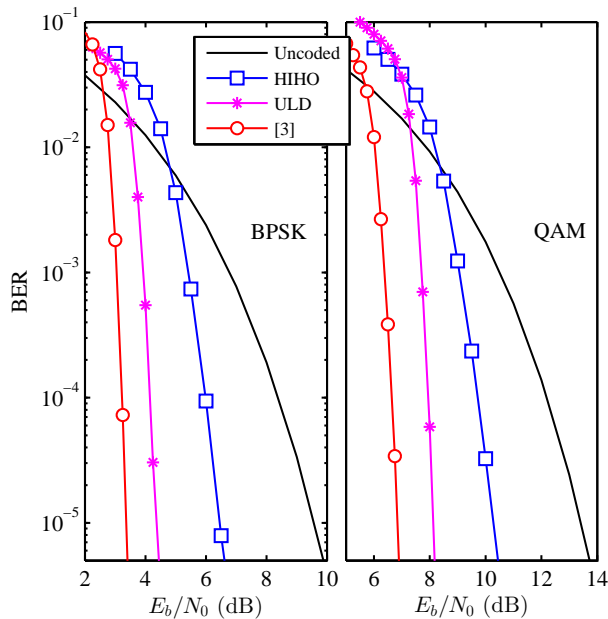


Fig. 3. BER of TPC eBCH(64, 57, 4)<sup>2</sup>.

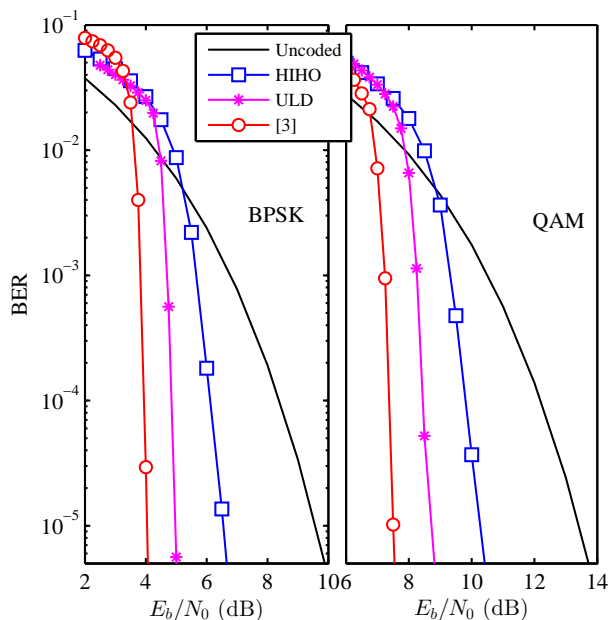


Fig. 4. BER of TPC eBCH(128, 120, 4)<sup>2</sup>.

TABLE III

RELATIVE DECODING STs OF THE ULD COMPARED TO [3].

	(16,11,4) <sup>2</sup>	(32,26,4) <sup>2</sup>	(64,57,4) <sup>2</sup>	(128,120,4) <sup>2</sup>
$p=3$	0.35	0.46	0.56	0.88
$p=4$	0.22	0.26	0.37	0.56
$p=5$	0.12	0.15	0.22	0.36

Generally speaking, it is difficult to compute the latency of large systems analytically, therefore, the simulation time (ST) is used to indicate latency. Table III shows the ST ratio of the ULD to [3] using typical values of  $p$ . As it can be noted from the Table, the ULD latency is smaller than [3] for all the considered codes and for all values of  $p$ . However, the difference becomes significant for  $p = 4$  and 5, particularly for short codes. It is worth noting that the ST may vary based on the simulation platform and script optimization. Nevertheless, it is typically considered as a useful indicator for the complexity and latency in several scenarios.

#### IV. CONCLUSION

In this letter, we presented an efficient decoder for turbo product codes. The new decoder is highly efficient due to the elimination of the hard decision decoding process used in conventional TPCs decoders. The elimination of the HDD process enabled the design of highly efficient ED calculator with a complexity that is significantly smaller than what is required by the conventional SISO decoders. In addition to the substantial complexity reduction, the new decoder will be much faster than other decoders, and hence will reduce the delay and buffering requirements at the receiver. Moreover, the error correction capability of the new decoder is comparable to the near-optimal decoder where the coding gain losses are only about 1 dB using BPSK in AWGN channels at error rate of  $10^{-5}$ . For higher order modulation, the coding gain loss increased to 1.3 dB when 16-QAM is used. The ULD BER performance was also evaluated in Rayleigh fading channels using BPSK modulation. The obtained results demonstrated the ULD coding gain loss was about 5% as compared to the near-optimum for short codes, and it was about 11.7% for long codes.

#### REFERENCES

- [1] H. Mukhtar, A. Al-Dweik, and A. Shami, "Turbo product codes: Applications, challenges, and future directions," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 3052–3069, Fourthquarter 2016.
- [2] A. Al-Dweik, S. Goff, and B. Sharif, "A hybrid decoder for block turbo codes," *IEEE Trans. Commun.*, vol. 57, no. 5, pp. 1229–1232, May 2009.
- [3] R. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, Aug. 1998.
- [4] P.-Y. Lu, E.-H. Lu, and T.-C. Chen, "An efficient hybrid decoder for block turbo codes," *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2077–2080, Dec. 2014.
- [5] A. Al-Dweik and B. Sharif, "Non-sequential decoding algorithm for hard iterative turbo product codes," *IEEE Trans. Commun.*, vol. 57, no. 6, pp. 1545–1549, June 2009.
- [6] C. Argon and S. McLaughlin, "An efficient Chase decoder for turbo product codes," *IEEE Trans. Commun.*, vol. 52, no. 6, pp. 896–898, June 2004.
- [7] R. F. T. El-Din, R. M. El-Hassani, and S. H. El-Ramly, "A novel high-speed systematic encoder for long binary cyclic codes," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 984–987, May 2013.
- [8] J. Cho and W. Sung, "Efficient software-based encoding and decoding of BCH codes," *IEEE Trans. Comput.*, vol. 58, no. 7, pp. 878–889, July 2009.

TABLE II

CODING GAIN IN FADING CHANNELS USING BPSK MODULATION

	Coding Gain (dB)			
	(16,11,4) <sup>2</sup>	(32,26,4) <sup>2</sup>	(64,57,4) <sup>2</sup>	(128,120,4) <sup>2</sup>
[3]	37.8	36.8	34.5	32.06
ULD	35.9	34.3	31.6	28.3
HIHO	26.0	25.4	24.1	22.9