Received 28 April 2024; revised 9 June 2024; accepted 26 June 2024. Date of publication 5 July 2024; date of current version 29 July 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.3424226

Reliability-Based Decoding of Low-Density Lattice Codes Using Gaussian and Eisenstein Integers

WARANGRAT WIRIYA (Member, IEEE), AND BRIAN M. KURKOSKI (Member, IEEE)

School of Information Science, Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Nomi 923-1292, Japan

CORRESPONDING AUTHOR: B. M. KURKOSKI (e-mail: kurkoski@jaist.ac.jp)

This work was supported by the JSPS Kakenhi under Grant JP 21H04873.

ABSTRACT This paper proposes reliability-based decoding for complex low-density lattice codes (CLDLC) which can be applied to both Gaussian and Eisenstein integers. Two major contributions are: first, a decoding algorithm for CLDLC using a likelihood-based reliability function is used to determine the number of complex Gaussian functions at the variable node. This allows each message to be approximated by a variable number of Gaussian functions depending upon its reliability. An upper bound on the Kullback-Leibler (KL) divergence of the approximation is formed to find selection thresholds via linear regression. Second, a construction of CLDLC using Eisenstein integers is given. Compared to Gaussian integers, this reduces the complexity of CLDLC decoding by exploiting the structure of the Eisenstein integers. The proposed CLDLC decoding algorithm has higher performance and lower complexity compared to existing algorithms. When the reliability-based algorithm is applied to Eisenstein integer CLDLC decoding, the complexity is reduced to $\mathcal{O}(n \cdot t \cdot 1.35^{d-1})$ at the volume-to-noise ratio of 6 dB, for lattice dimension n, with degree d inverse generator matrix and t decoding iterations. Decoding CLDLC using Eisenstein integers has lower complexity than CLDLC using Gaussian integers when $n \geq 49$.

INDEX TERMS Complex low-density lattice codes, low-density lattice codes, parametric decoder, lattice decoder, Eisenstein integers, Gaussian integers.

I. INTRODUCTION

A. BACKGROUND

CODED MODULATION increases the spectral efficiency of wireless communication systems. Lattice codes are elegant and powerful structures for coded modulation that not only can achieve the capacity of the additive white Gaussian noise (AWGN) channel but are also a key ingredient to many multi-terminal schemes that exploit linearity properties [1], [2], [3]. There is an always-increasing demand for increased spectrum efficiency, massive connectivity, and higher data rates; in post-5G or 6G wireless networks, lattice codes are a potential candidate to achieve these goals [4], [5], [6], [7].

Low-density lattice codes (LDLC) defined over the real numbers were proposed by Sommer et al. [8]. These lattice codes can be encoded and decoded efficiently in high-dimensional Euclidean space, and error-free decoding is possible within 0.6 dB of the unconstrained power channel capacity [9]. LDLC lattices have a sparse inverse generator matrix. Based on this property, LDLC can be decoded on principles similar to low-density parity-check (LDPC) codes using the belief propagation (BP) decoding algorithm [10]. The messages in the iterative decoding process are probability density functions for each lattice symbol, and the algorithm has a complexity that is linear in lattice dimension n.

These real-valued LDLC were extended to the complex numbers by Yona and Feder [11]. Such complex low-density lattice codes (CLDLC) provide several advantages. For instance, CLDLC are naturally matched to complex-valued channels, and they are a suitable construction for the compute-and-forward (C&F) paradigm as an alternative strategy for wireless networks [12]. Another advantage is that CLDLC increases the coding gain compared to real LDLC [11]. An n-dimensional CLDLC is not simply a $2n$-dimensional real LDLC, but rather the CLDLC is directly generated over the domain of complex numbers. CLDLC outperforms $2n$-dimensional real LDLC because the real-valued parity check matrix normally suffers from short loops. CLDLC lattices can be decoded using the belief propagation

(BP) algorithm [10], where the messages in the iterative processing of CLDLC are complex Gaussian functions. BP decoding of CLDLC confronts the same issue as real-valued LDLC, that an infinite Gaussian mixture must be approximated for the decoder implementation.

B. RELATED WORKS

1) *Infinite Gaussian mixture approximation for CLDLC*: Yona and Feder [11] used a high-complexity Gaussian mixture reduction (GMR) algorithm to approximate Gaussian mixture messages based on Gaussian integers (GI). Their decoder sorts the whole list of Gaussian mixtures and the Gaussian mixtures which satisfy a given condition will be grouped. Each incoming message at the variable node is approximated by 9 complex Gaussian functions. The GMR algorithm must be performed at every multiplication at the variable node, on each iteration. Due to a large number of approximated Gaussian functions, the complicated GMR algorithm, and the large number of uses of the GMR algorithm at the variable node, this proposed algorithm may not be suitable for hardware implementation.
2) *Infinite Gaussian mixture approximation for the real-valued LDLC*: There has been substantial past work on complexity reduction of decoding for the real-valued LDLC, such as mixture reduction [9], [13], searching and sorting [14], using fixed two or fixed three Gaussian functions [15], use of $2d-2$ Gaussian functions [16] and a reliability function at the variable node [17].
3) *Lattice codes based on Eisenstein and Gaussian integers*: The E_6 lattice and the Coxeter-Todd lattice K_{12} are well-known lattices based on Eisenstein integers and have the highest known packing density among lattices in their respective dimensions [18]. Signal constellations based on Eisenstein integers were introduced in [19] and [20]. Lattice codes based on Construction A over Eisenstein integers for compute-and-forward were proposed in [21], and extensions to the ring of algebraic integers of a general imaginary quadratic field which includes Gaussian integers have been proposed in [22] and [23]. In addition, there are several works that introduced Gaussian integers for CLDLC, for example, [24].
4) *Real-valued lattice codes*: Polar lattices were introduced in [25] which are formed from binary polar codes using Construction D. The design was guided by channel capacity. The decoding algorithm is based on successive cancelation (SC) decoding for two or more component codes. Afterward, a design technique for polar code lattices of finite dimension using use the explicit finite-length code properties was proposed in [26]. The result based on the successive cancelation list (SCL) decoding algorithm was also provided for small dimensions of polar lattices. In addition, low-density construction-A (LDA) lattices were introduced in [27]. LDA lattices are similar to LDLC in the sense that they both have sparse matrices and are decoded using belief propagation. However, they are quite different in important ways — CLDLC lattices design is based on the selection of non-zero coefficients, whereas LDA design is based on the design of a p-ary LDPC code. CLDLC decoding is based on message-passing using complex Gaussian mixtures while LDA decoding is based on message-passing using p-ary probability vectors.
5) *Other codes based on Gaussian and Eisenstein integers*: Besides lattice codes, there are several codes that also can be constructed based on Gaussian and Eisenstein integers. Huber introduced a construction of block codes over Gaussian and Eisenstein integers [28] and [29]. Afterward, other works such as [30] and [19] proposed LDPC codes over Gaussian integers, and signal constellations based on Eisenstein Integers, respectively. These codes consider finite alphabets and modulo arithmetic over Gaussian and Eisenstein integers.

C. CONTRIBUTION

To reduce the complexity of CLDLC decoder proposed by Yona and Feder [11], this paper proposes two major contributions. The first is reliability-based CLDLC decoding. In reliability-based decoding, the number of Gaussian functions in the check-to-variable message is adaptively selected depending upon the reliability. It provides a small number of Gaussian functions in each check-to-variable message; in particular, it allows the use of a single Gaussian function when the message has high reliability. Fewer Gaussian functions mean lower decoder complexity. Reliability-based decoding uses a threshold to select the number of Gaussian functions. This threshold can be found using the Kullback-Leibler (KL) divergence between the approximation and the true distribution, but explicitly computing the divergence is inefficient. Instead, we form an upper bound on this KL divergence, and linear regression is used to efficiently estimate this threshold.

The second contribution is a CLDLC construction using Eisenstein integers (EI). This new construction reduces the complexity of message-passing decoders where an infinite Gaussian mixture is represented by a finite mixture. The advantage of the Eisenstein integers over the Gaussian integers is that the hexagonal Voronoi cells of the Eisenstein integer lattice have the tightest packing in two dimensions. As a result, the quality of the message-passing approximation is improved and mixtures have a smaller number of Gaussian functions. This smaller number of Gaussian functions leads to lower decoder complexity. While the extension of CLDLC to Eisenstein integers is straightforward, the contribution is that the Eisenstein integer construction lowers decoding complexity, compared to Gaussian integers.

Reliability-based decoding is applied to decoding both Eisenstein integer CLDLC (EI-CLDLC) and Gaussian integer CLDLC (GI-CLDLC) constructions in this paper. For decoding GI-CLDLC, reliability-based decoding has lower complexity, and slightly better performance, than the

previous CLDLC decoder of Yona and Feder [11]. Moreover, EI-CLDLC decoding has lower complexity than GI-CLDLC decoding for dimension $n \geq 49$.

D. NOTATION AND DEFINITIONS

We use $\mathbb{C}$, $\mathbb{R}$, and $\mathbb{Z}$ to denote the fields of complex numbers, real numbers, and integers, respectively. We use boldface lowercase with bar $\bar{\mathbf{x}}$ to denote column vectors and boldface uppercase $\mathbf{H}$ to denote matrices. We use three interchangeable ways of writing complex numbers: $z = a + bi$, vector $\mathbf{z} = [a\ b]^T$ (without a bar) where $[\cdot]^T$ is the transpose, and $re^{i\theta}$ where $r = \sqrt{a^2 + b^2}$ and $\theta = \tan^{-1}(\frac{b}{a})$. Also, $x_{Re} = a$ and $x_{Im} = b$ denote the real and imaginary parts of the complex number x. The conjugate transpose or Hermitian transpose of $\mathbf{G}$ is $\mathbf{G}^\dagger$, and the complex conjugate of c is c^*. The n-by-n identity matrix is $\mathbf{I}_n$.

The rest of the paper is organized as follows. Section II describes the preliminaries of CLDLCs, its Latin square construction for the check matrix and the overall picture of the reliability-based parametric BP decoding algorithm describe in Sections II-B and II-C, respectively. Section II-D describes operations over complex Gaussian mixtures and the moment matching approximation. Section III explains the proposed approximation of Gaussian mixtures at variable nodes. The approximation using Gaussian and Eisenstein integers, the reliability of the check-to-variable message, the number of Gaussian function selections based on its reliability, the threshold function formed by an upper bound on KL divergence, and the detail of Gaussian mixture approximation at the variable node are explained in Section III-A to Section III-E. Section IV gives the numerical evaluation of decoder error rate, complexity, and convergence properties for various lattice dimensions. The last section, Section V summarizes the paper.

II. PRELIMINARIES OF COMPLEX LOW-DENSITY LATTICE CODES

This section provides the preliminaries of CLDLC. Section II-A describes the overall picture of CLDLC and the unconstrained power complex additive white Gaussian noise (CAWGN) system. Section II-B explains a Latin square construction which is the inverse generator matrix for encoding. Section II-C describes the overall picture of the reliability-based parametric BP decoding algorithm. (The proposed infinite Gaussian mixture approximation at variable nodes is explained in Section III.) The last subsection describes operations on complex Gaussian functions and the moment matching approximation which are used in Section II-C.

A. CLDLC AND UNCONSTRAINED POWER CAWGN SYSTEM

An n-dimensional complex lattice Λ is a discrete additive subgroup of $\mathbb{C}^n$, defined by a non-singular square generator matrix $\mathbf{G}$. A lattice point $\bar{\mathbf{x}}$ is an integral linear combination of basis vectors in $\mathbf{G}$. Each lattice point is constructed from:

$$\bar{\mathbf{x}} = \mathbf{G}\bar{\mathbf{b}}. \tag{1}$$

Complex-valued lattices are a generalization of real-valued lattices. In CLDLC, the inverse of $\mathbf{G}$, $\mathbf{H} = \mathbf{G}^{-1}$ is restricted to be sparse to develop a linear-time iterative decoding scheme [8], where $\mathbf{H}$ is called the check matrix or inverse generator matrix. For encoding, CLDLC can use Jacobi method as in the real-valued LDLC case [8]. We consider two types of complex integers $\mathbf{b}$, Gaussian integers (GI) $\mathbb{Z}[i]$ ($i = \sqrt{-1}$) and Eisenstein integers (EI) $\mathbb{Z}[\omega]$ where $\omega = \frac{-1+i\sqrt{3}}{2}$. A vector of GI is written $\bar{\mathbf{b}} = [b_{1,Re} + ib_{1,Im}, b_{2,Re} + ib_{2,Im}, \ldots, b_{n,Re} + ib_{n,Im}]$, and a vector of EI is written $\bar{\mathbf{b}} = [b_{1,Re} + \omega b_{1,Im}, b_{2,Re} + \omega b_{2,Im}, \ldots, b_{n,Re} + \omega b_{n,Im}]$, where $b_{Re}, b_{Im} \in \mathbb{Z}^n$.

Since we consider the unconstrained lattice only, a complex lattice point $\bar{\mathbf{x}} \in \Lambda$ is transmitted over a complex additive white Gaussian noise (CAWGN) channel, and the received sequence $\bar{\mathbf{y}} = (y_1, y_2, \ldots, y_n) \in \mathbb{C}^n$ is:

$$\bar{\mathbf{y}} = \bar{\mathbf{x}} + \bar{\mathbf{z}}, \tag{2}$$

where the vector $\mathbf{z}_j \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_2)$ is complex additive Gaussian noise, and $j = 1, 2, \ldots, n$. $\mathcal{CN}$ denotes complex-normal distribution, σ^2 is noise variance of each element of complex number.

The Voronoi region of a lattice point is the set of points in $\mathbb{C}^n$ that are closest to the lattice point. For a square generator matrix $\mathbf{G}$, the Voronoi region volume $V(\Lambda)$ equals $\det(\mathbf{G}^\dagger\mathbf{G})$ for lattices with Gaussian integers, and $V(\Lambda)$ equals $(\frac{\sqrt{3}}{2})^n \det(\mathbf{G}^\dagger\mathbf{G})$ for lattices with Eisenstein integers [31]. We consider here the unconstrained power system. this paper uses the volume-to-noise ratio (VNR) as an analog to the signal-to-noise ratio. The VNR for Gaussian and Eisenstein integer lattices is defined as:

$$VNR = \frac{V(\Lambda)^{2/n}}{2\pi e\sigma^2}. \tag{3}$$

The Poltyrev capacity, or unconstrained lattice capacity, is $\sigma^2 = \frac{1}{2\pi e}$ [32], which corresponds to $VNR = 0$ dB. In the sequel, we normalize the Voronoi region volume such that $V(\Lambda) = 1$.

The maximum likelihood lattice point estimate $\hat{\bar{\mathbf{x}}}$ is:

$$\hat{\bar{\mathbf{x}}} = \arg\max_{\bar{\mathbf{x}}\in\Lambda} \Pr(\bar{\mathbf{y}}|\bar{\mathbf{x}}). \tag{4}$$

If $\bar{\mathbf{x}} = \hat{\bar{\mathbf{x}}}$ the correct codeword is received, or an error occurred otherwise.

B. LATIN SQUARE CONSTRUCTION

A Latin square construction for check matrix $\mathbf{H}$ was proposed by Sommer et al. for real number LDLC [8] and extended by Yona and Feder [11] to CLDLC. A Latin square CLDLC can be constructed by designing a check matrix $\mathbf{H}$ having a constant number of non-zero values (degree) d in every row and column. Let $\mathcal{H} = \{h_1, h_2, h_3, \ldots, h_d\}$ be a generating sequence, where $h_i \in \mathbb{C}$ and $i = 1, 2, \ldots, d$. The non-zero entries of each row and each column are $h_1, h_2, h_3, \ldots, h_d$, so that each row (respectively, column) is a permutation

of any other row (column), except for complex rotations, including sign changes.

Sommer et al. showed that a sufficient condition to achieve exponential convergence of the message variance in the BP decoder is to select the generator sequence $|h_1| \geq |h_2| \geq \cdots \geq |h_d|$ such that:

$$\alpha = \frac{\sum_{j=2}^{d} |h_j|^2}{|h_1|^2} < 1, \tag{5}$$

where $|\cdot|$ denotes the Euclidean norm [8]. An example of a Latin square check matrix $\mathbf{H}$ is shown in (6),

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & h_2 & 0 & 0 & h_3 & 0 \\ h_2^* & 1 & 0 & 0 & 0 & 0 & 0 & h_3 \\ 0 & -h_3^* & 1 & 0 & 0 & 0 & 0 & h_2 \\ 0 & 0 & -h_3^* & 1 & 0 & 0 & h_2 & 0 \\ -h_3 & 0 & 0 & 0 & 1 & h_2 & 0 & 0 \\ 0 & 0 & -h_2^* & h_3 & 0 & 1 & 0 & 0 \\ 0 & -h_2^* & 0 & 0 & h_3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & h_2 & h_3 & 0 & 1 \end{bmatrix}. \tag{6}$$

The Latin square check matrix $\mathbf{H}$ has the lattice dimension $n = 8$, row and column degree $d = 3$. The generating sequence $\mathcal{H}$ is $|h_1| = 1$ and $|h_2| = |h_3| = \frac{1}{\sqrt{d}}$, for example, $\mathcal{H} = \{1, \sqrt{\frac{2}{9}} + i\sqrt{\frac{1}{9}}, \sqrt{\frac{2}{9}} + i\sqrt{\frac{1}{9}}\}$, and in the sequel, we normalize $V(\Lambda) = 1$.

C. CLDLC RELIABILITY-BASED PARAMETRIC BP DECODER

For CLDLC decoding, this paper proposes to approximate the infinite complex Gaussian mixture function with a finite number of complex Gaussian functions using the reliability of check-to-variable messages and a threshold function. This approximation takes place at the variable node. This subsection describes the overall picture of the CLDLC reliability parametric BP decoding algorithm. Details of the approximation at the variable node are explained in Section III.

CLDLC message-passing decoding can be performed on a bipartite graph, where rows of $\mathbf{H}$ correspond to check nodes, and columns of $\mathbf{H}$ correspond to variable nodes. The variable-to-check messages are $q_k(\mathbf{z})$ and the check-to-variable messages are $R_k(\mathbf{z})$. The proposed CLDLC reliability parametric BP decoding algorithm is as follows.

- *Initialization:* For the CAWGN channel (2), the initial variable-to-check message is

$$y_i(\mathbf{z}) = \mathcal{N}\Big(\mathbf{z}; \mathbf{y}_i, \sigma^2 \mathbf{I}_2\Big), \tag{7}$$

  for $i = 1, 2, \ldots, n$. Received messages are sent to connected check nodes in the initial step.
- *Check-to-variable message:* The check node incoming messages are $q_k(\mathbf{z})$, where $k = 1, \ldots, d-1$ single Gaussian functions and output is $k = d$. In the first iteration, $q_k(\mathbf{z})$ is the channel message $y_k(\mathbf{z})$. (The subscript i of $y_i(\mathbf{z})$ changed to k because the incoming messages of one specific check node are considered.) The corresponding non-zero coefficients from $\mathcal{H}$ are $h_1, \ldots, h_d$. The output $R_d(\mathbf{z})$ is:

$$R_d(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_d, \mathbf{V}_d). \tag{8}$$

  where mean $\mathbf{m}_d$ and covariance matrix $\mathbf{V}_d$ can be found from

$$\mathbf{m}_d = -[h_d]^{-1} \sum_{k=1}^{d-1} [h_k]\mathbf{m}_k \text{ and}$$
$$\mathbf{V}_d = \sum_{k=1}^{d-1} \left[\frac{h_k}{h_d}\right] \mathbf{V}_k \left[\frac{h_k}{h_d}\right]^T, \tag{9}$$

  where $[h_d]$, $[h_k]$, and $[\frac{h_k}{h_d}]$ can be expressed as $[h_d] = \begin{bmatrix} Re\{h_d\} & -Im\{h_d\} \\ Im\{h_d\} & Re\{h_d\} \end{bmatrix}$, $[h_k]$ expression is same as $[h_d]$ but the subscript changes to k, and $[\frac{h_k}{h_d}] = \begin{bmatrix} Re\{\frac{h_k}{h_d}\} & -Im\{\frac{h_k}{h_d}\} \\ Im\{\frac{h_k}{h_d}\} & Re\{\frac{h_k}{h_d}\} \end{bmatrix}$.
- *Variable-to-check message:* The messages $R_k(\mathbf{z})$ coming from the check nodes are expanded to a periodic function $\widetilde{R}_k(\mathbf{z})$, but with a finite number of Gaussian functions depending on their reliability. Then, the moment matching approximation *MM* is used to find the message of single-Gaussian function $f_d(\mathbf{z})$ sent back to the check node:

$$f_d(\mathbf{z}) = y_i(\mathbf{z}) \prod_{k=1}^{d-1} \widetilde{R}_k(\mathbf{z}) \text{ and}$$
$$q_d(\mathbf{z}) = MM(f_d(\mathbf{z})). \tag{10}$$

  The efficient implementation of the variable node is a contribution of this paper, Section III describes details of this implementation. Section III-A shows how to select Gaussian functions near the channel value y_i. The specific number of Gaussian functions selected should be as small as possible which are selected using reliability and thresholds as shown in Section III-B to Section III-D. Section II-C gives the variable node function (10) in detail which includes the periodic expansion.
- *Final decision:* At the last iteration, the product without omitting any message is performed:

$$q_i^{final}(\mathbf{z}) = y_i(\mathbf{z}) \prod_{k=1}^{d} \widetilde{R}_k(\mathbf{z}). \tag{11}$$

  The lattice point estimate $\hat{\mathbf{x}}$ and the integer estimate $\hat{\mathbf{b}}$ are:

$$\hat{\mathbf{x}}_i = \arg\max_{\mathbf{z}} q_i^{final}(\mathbf{z}), \text{ and } \hat{\mathbf{b}} = \lfloor \mathbf{H}\hat{\mathbf{x}} \rceil, \tag{12}$$

  where $\lfloor \cdot \rceil$ denotes the rounding to the nearest integer.

D. OPERATIONS ON COMPLEX GAUSSIAN MIXTURES

This subsection describes operations on complex Gaussian functions which are messages in belief propagation decoding, described in Section II-C. The probability density function (PDF) of a complex Gaussian function with 2×1 mean vector $\mathbf{m}$ and 2×2 covariance matrix $\mathbf{V}$ is:

$$\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{V}) = \frac{1}{2\pi\sqrt{|\mathbf{V}|}} e^{-\frac{1}{2}(\mathbf{z}-\mathbf{m})^T \mathbf{V}^{-1}(\mathbf{z}-\mathbf{m})}. \tag{13}$$

From the AWGN channel assumption, the messages in belief propagation decoding are Gaussian mixtures. The message $f(\mathbf{z})$ is a mixture of N Gaussian functions,

$$f(\mathbf{z}) = \sum_{j=1}^{N} c_j \mathcal{N}\big(\mathbf{z}; \mathbf{m}_j, \mathbf{V}_j\big), \tag{14}$$

where $c_j \geq 0$ are mixing coefficients with $\sum_{j=1}^{N} c_j = 1$, $\mathbf{m}_j$ and $\mathbf{V}_j$ are mean and covariance of the Gaussian mixture. In this way, each Gaussian mixture can be described by a set of triples $(\mathbf{m}_1, \mathbf{V}_1, c_1), \ldots, (\mathbf{m}_N, \mathbf{V}_N, c_N)$.

Let $f(\mathbf{z}) = \sum_{j=1}^{N} f_j(\mathbf{z})$ and $g(\mathbf{z}) = \sum_{k=1}^{M} g_k(\mathbf{z})$ be two Gaussian mixtures, where $f_j(\mathbf{z}) = c_1 \mathcal{N}(\mathbf{z}; \mathbf{m}_1, \mathbf{V}_1)$ and $g_k(\mathbf{z}) = c_2 \mathcal{N}(\mathbf{z}; \mathbf{m}_2, \mathbf{V}_2)$. The product of two Gaussian mixtures $f(\mathbf{z}) \cdot g(\mathbf{z})$ is a mixture of NM Gaussian functions. Each mixture element is the pairwise product of two components $f_j(\mathbf{z})$ and $g_k(\mathbf{z})$, a Gaussian function $s(\mathbf{z}) = c\mathcal{N}(\mathbf{z}; \mathbf{m}, \mathbf{V})$ with mean $\mathbf{m}$, variance $\mathbf{V}$ and mixing coefficient c given by:

$$\mathbf{V} = \Big(\mathbf{V}_1^{-1} + \mathbf{V}_2^{-1}\Big)^{-1}, \quad \mathbf{m} = \mathbf{V}\Big(\mathbf{V}_1^{-1}\mathbf{m}_1 + \mathbf{V}_2^{-1}\mathbf{m}_2\Big), \tag{15}$$

$$c = \frac{c_1 c_2}{2\pi\sqrt{|\mathbf{V}_1 + \mathbf{V}_2|}} e^{-\frac{1}{2}(\mathbf{m}_1-\mathbf{m}_2)^T(\mathbf{V}_1+\mathbf{V}_2)^{-1}(\mathbf{m}_1-\mathbf{m}_2)}. \tag{16}$$

Let $f(\mathbf{z})$ be a Gaussian mixture. The moment matching approximation (MM) which minimizes the Kullback-Leiber divergence between $f(\mathbf{z})$ and $q(\mathbf{z})$ is used to approximate $f(\mathbf{z})$ with a single Gaussian function $q(\mathbf{z})$ [24]. The MM approximation finds the single Gaussian function $q(\mathbf{z})$ which has the same mean $\mathbf{m}$ and variance $\mathbf{V}$ as $f(\mathbf{z})$, given by:

$$\mathbf{m} = \sum_{j=1}^{N} c_j \mathbf{m}_j, \quad \mathbf{V} = \sum_{j=1}^{N} c_j \Big(\mathbf{V}_j + (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T\Big). \tag{17}$$

This operation is denoted as:

$$q(\mathbf{z}) = MM(f(\mathbf{z})). \tag{18}$$

III. RELIABILITY-BASED GAUSSIAN MIXTURE APPROXIMATION FOR CLDLC DECODING

This section explains the proposed algorithm for infinite Gaussian mixture approximation at variable nodes of CLDLC decoder in Section II-C. The number of finite Gaussian functions in the check-to-variable message is adaptively selected depending upon its reliability. The message reliability is compared to the threshold functions to select the number of finite Gaussian functions. This approximation is done for both Gaussian integers and Eisenstein integers. Section III-A describes the approximation using Gaussian integers and Eisenstein integers. Section III-B describes the use of message reliability and the threshold function to approximate the different numbers of finite Gaussian functions. Section III-C and D explain the reliability function of the check-to-variable message and the threshold function formed by an upper bound on KL divergence, respectively. The last subsection describes the approximation at the variable node in Section II-C.

A. GAUSSIAN MIXTURE APPROXIMATION USING GAUSSIAN AND EISENSTEIN INTEGERS

This subsection describes an approximation of an infinite Gaussian mixture using a finite Gaussian mixture; this is done for both Gaussian integers and Eisenstein integers. This is used at the variable node of the belief-propagation decoding algorithm for CLDLC lattices in Section II-C. The advantage of Eisenstein integers over Gaussian integers is that Eisenstein integers give a more accurate approximation for a fixed number of Gaussian functions in the mixture. If a fixed number of Gaussian functions (corresponding to either Gaussian integers or Eisenstein integers) which have means closest to the channel value are selected in the approximation, for example, three Gaussian functions. Three Eisenstein integers will together give a higher likelihood than three GI. This is because Eisenstein integers have hexagonal packing, which is tighter than the cubic packing of GI (the hexagonal packing is the tightest in two dimensions). Due to the tighter packing, the distance to the channel value will be lower on average.

In exact BP decoding, the messages $R_k(\mathbf{z})$ which come from the check nodes are expanded over all integers at the variable nodes [8]. (For convenience, the subscript k of $R_k(\mathbf{z})$ will be omitted, and call it $R(\mathbf{z})$ from now on.) The check-to-variable message is an infinite complex Gaussian mixture, which for GI is given by

$$R(\mathbf{z}) = \sum_{j \in \mathbb{Z}[i]} \mathcal{N}\left(\mathbf{z}; \mathbf{m}_c + \frac{j}{h}, \mathbf{V}_c\right), \tag{19}$$

and for EI, the sum is over $\mathbb{Z}[\omega]$. The infinite periodic Gaussian functions $R(\mathbf{z})$ with period $1/h$ has mean $\mathbf{m}_c$ and variance $\mathbf{V}_c$. The channel message $Y(\mathbf{z})$ is a single complex Gaussian function with mean $\mathbf{m}_a$ and variance $\mathbf{V}_a = \sigma^2 \mathbf{I}_2$. The exact product of $Y(\mathbf{z})$ and $R(\mathbf{z})$ at the variable node is also an infinite complex Gaussian mixture. This infinite Gaussian mixture $Y(\mathbf{z})R(\mathbf{z})$ must be reduced to a finite number of Gaussian functions $Y(\mathbf{z})\widetilde{R}(\mathbf{z})$ in practice. $\widetilde{R}(\mathbf{z})$ is the summation in (19) restricted to some finite subset $\mathcal{B}$, given by

$$\widetilde{R}(\mathbf{z}) = \sum_{j \in \mathcal{B}} \mathcal{N}\left(\mathbf{z}; \mathbf{m}_c + \frac{j}{h}, \mathbf{V}_c\right), \tag{20}$$

where $\mathcal{B} \subset \mathbb{Z}[i]$ in the GI case and $\mathcal{B} \subset \mathbb{Z}[\omega]$ in the EI case.

Choosing a finite subset of integers $\mathcal{B}$ of $\widetilde{R}(\mathbf{z})$ which is sufficient to represent the infinite Gaussian mixtures $R(\mathbf{z})$ is

the key for an accurate approximation. The periodic Gaussian functions far from the channel message have near-zero mixing coefficients and can be safely ignored. Therefore, $R(\mathbf{z})$ can be restricted using some finite integer set $\mathcal{B}$ which are near the channel message $Y(\mathbf{z})$. How to select $\mathcal{B}$ is the subject of the next subsection.

Next, we present an approximation of $R(\mathbf{z})$ based on a subset of Gaussian integers $\mathcal{B} \subset \mathbb{Z}[i]$ and Eisenstein integers $\mathcal{B} \subset \mathbb{Z}[\omega]$ at the variable node. For Gaussian integers, the three cases of $|\mathcal{B}| = 1, |\mathcal{B}| = 2$ and $|\mathcal{B}| = 4$ Gaussian functions near $Y(\mathbf{z})$ are considered, where $|\mathcal{B}|$ is the number of elements of $\mathcal{B}$ or the number of Gaussian functions in the approximation. Each case of $|\mathcal{B}|$ is used to approximate check-to-variable messages which have different reliability. If a check-to-variable message has high reliability, $|\mathcal{B}| = 1$ is sufficient to represent the infinite Gaussian mixtures. But if the check-to-variable message has intermediate or low reliability $|\mathcal{B}| = 2$ or 4 will be applied, respectively. For Eisenstein integers, the three cases of $|\mathcal{B}| = 1, |\mathcal{B}| = 2$ and $|\mathcal{B}| = 3$ Gaussians near $Y(\mathbf{z})$ are considered. If the check-to-variable message has high and intermediate reliability, $|\mathcal{B}| = 1$ and 2 will be applied as in the case of Gaussian integers. The difference is if the check-to-variable message has low reliability, $|\mathcal{B}| = 3$ is sufficient to approximate the infinite Gaussian mixtures.

- *High-reliability:* For Gaussian integers, let the single Gaussian function set be $\mathcal{B} = \{\mathbf{b}_0\}$, where $\mathbf{b}_0$ is

$$\mathbf{b}_0 = \lceil -[h]\ (\mathbf{m}_c - \mathbf{m}_a) \rfloor. \tag{21}$$

  For Eisenstein integers, the single Gaussian function $\mathcal{B} = \{\mathbf{b}_0\}$ can be found using coset decoding of the hexagonal lattice [18]. First, $\widetilde{\mathbf{b}}_0 = -[h]\ (\mathbf{m}_c - \mathbf{m}_a)$ is defined as the soft value of $\mathbf{b}_0$. Then, the soft value $\widetilde{\mathbf{b}}_0$ is quantized to an Eisenstein integer $\mathbf{b}_0$ following [31]. The resulting Gaussian function is $\widetilde{R}(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_c + [h]^{-1}\mathbf{b}_0, \mathbf{V}_c)$.
- *Intermediate reliability:* Two Gaussian integers are selected, $\mathcal{B} = \{\mathbf{b}_0, \mathbf{b}_1\}$. For Gaussian integers, $\mathbf{b}_0$ is found from (21). $\mathbf{b}_1$ is found from the minimum Euclidean distance between the soft value $\widetilde{\mathbf{b}}_0$ and the eight integers nearest $\mathbf{b}_0$. Define $\zeta_0 = \mathbf{b}_0 + \{[-1\ -1]^T, [-1\ \ 0]^T, [-1\ \ 1]^T, [0\ -1]^T, [0\ \ 1]^T, [1\ -1]^T, [1\ 0]^T, [1\ 1]^T\}$ so that $\mathbf{b}_1$ is:

$$\mathbf{b}_1 = \min_{\mathbf{x} \in \zeta_0} ||\mathbf{x} - \widetilde{\mathbf{b}}_0||^2. \tag{22}$$

  For Eisenstein integers, $\mathbf{b}_0$ is from the previous item (high-reliability) and $\mathbf{b}_1$ is found from the minimum Euclidean distance between $\widetilde{\mathbf{b}}_0$ and the six integers nearest $\mathbf{b}_0$. Define as $\varepsilon_0 = \mathbf{b}_0 + \{[-1\ \ 0]^T, [1\ \ 0]^T, [-\frac{1}{2}\ \ \frac{\sqrt{3}}{2}]^T, [\frac{1}{2}\ \ \frac{\sqrt{3}}{2}]^T, [-\frac{1}{2}\ -\frac{\sqrt{3}}{2}]^T, [\frac{1}{2}\ -\frac{\sqrt{3}}{2}]^T\}$, so that $\mathbf{b}_1$ can be calculated using (22) but $x \in \zeta_0$ is substituted to $x \in \varepsilon_0$. The resulting Gaussian mixture is $\widetilde{R}(\mathbf{z}) = \sum_{j \in \{\mathbf{b}_0, \mathbf{b}_1\}} \mathcal{N}(\mathbf{z}; \mathbf{m}_c + [h]^{-1}j, \mathbf{V}_c)$. Note that eight and six integers nearest $\mathbf{b}_0$ for Gaussian and Eisenstein integer approximation are determined by the deep hole. The definition of deep hole is explained at the end of this subsection. For Gaussian integers, there are four deep holes around the integer which are connected to eighth integers. For Eisenstein integers, there are three deep holes around the integer which are connected to six integers.
- *Low reliability:* For Gaussian integers, four Gaussian integers are selected, $\mathcal{B} = \{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$. $\mathbf{b}_0$ and $\mathbf{b}_1$ can be found from equation (21) and (22). $\mathbf{b}_2$ can be computed as

$$\mathbf{b}_2 = \min_{\mathbf{x} \in \zeta_1} ||\mathbf{x} - \widetilde{\mathbf{b}}_0||^2, \tag{23}$$

  where $\zeta_1 = \zeta_0 - \mathbf{b}_1$, and $-$ denotes set subtraction. $\mathbf{b}_3$ can be found as,

$$\mathbf{b}_3 = \min_{\mathbf{x} \in \zeta_2} ||\mathbf{x} - \widetilde{\mathbf{b}}_0||^2, \tag{24}$$

  where $\zeta_2 = \zeta_1 - \mathbf{b}_2$. The resulting Gaussian mixture is $\widetilde{R}(\mathbf{z}) = \sum_{j \in \{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}} \mathcal{N}(\mathbf{z}; \mathbf{m}_c + [h]^{-1}j, \mathbf{V}_c)$. For Eisenstein integers, three Eisenstein integers are selected, $\mathcal{B} = \{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2\}$, where $\mathbf{b}_0$ and $\mathbf{b}_1$ can be found as above items (high and intermediate reliability), and $\mathbf{b}_2$ can be computed as (23) but $x \in \zeta_1$ is substituted to $x \in \varepsilon_1$, where $\varepsilon_1 = \varepsilon_0 - \mathbf{b}_1$. The resulting Gaussian mixture is $\widetilde{R}(\mathbf{z}) = \sum_{j \in \{\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2\}} \mathcal{N}(\mathbf{z}; \mathbf{m}_c + [h]^{-1}j, \mathbf{V}_c)$.

For Gaussian integers, example $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2$ and $\mathbf{b}_3$ are shown in Fig. 1-(a). In summary, $\mathbf{b}_0$ is the closest to the soft value $\widetilde{\mathbf{b}}_0$, $\mathbf{b}_1$ is second closest, etc. For Eisenstein integers, example $\mathbf{b}_0, \mathbf{b}_1$, and $\mathbf{b}_2$ of Eisenstein integers are shown in Fig. 1-(b). In the figures, T_1 and T_2 are thresholds to select between high, intermediate, and low reliability, and will be described in Section III-B.

The maximum value for $|\mathcal{B}|$ of 4 and 3 for Gaussian and Eisenstein integers is related to the number of lattice points connected to a deep hole. The deep hole is the furthest point from the lattice point on the Voronoi cell and are shown in Fig. 1-(a) and (b). If the check-to-variable message r is at the deep hole in the worst case, the number of high-likelihood integers $|\mathcal{B}|$ near r is 4 and 3 for Gaussian and Eisenstein integers, respectively.

B. RELIABILITY-BASED IMPLEMENTATION AT THE VARIABLE NODES

This subsection describes the reliability-based selection of Gaussian functions to be used at the variable node in Section II-C. From Section III-A, the infinite complex Gaussian mixtures can be approximated by 1, 2, or 4 Gaussian functions in the GI case, or 1, 2, or 3 Gaussian functions in the EI case. This subsection addresses how to select the number of Gaussian functions. We define the reliability, $1/\rho$ of the check-to-variable messages which is used to choose the number of Gaussian functions for each incoming message at the variable node. Two thresholds $T_1 < T_2$ will be set at the variable node. If the reliability is high, that is ρ satisfies $\rho \leq T_1$, then one Gaussian function will be selected. If the reliability is intermediate, that is, ρ satisfies

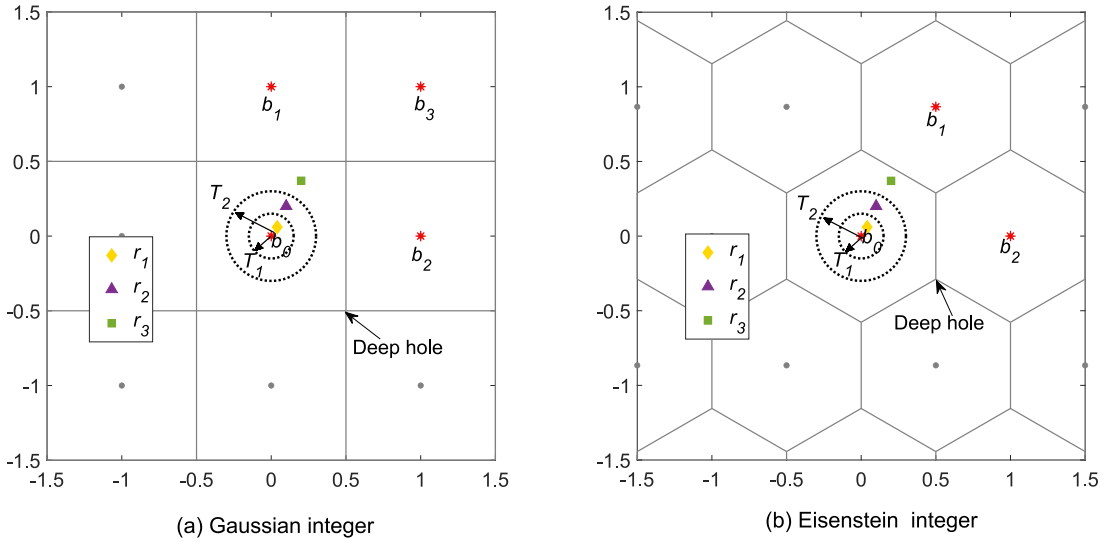


FIGURE 1. Gaussian approximation based on the reliability of check-to-variable message for (a) Gaussian integers, (b) Eisenstein integers for three example check-to-variable messages r_1, r_2 and r_3.

$T_1 < \rho \leq T_2$, then two Gaussian functions will be selected. Otherwise, the reliability is low and four Gaussian functions or three Gaussian functions will be selected for Gaussian integers or Eisenstein integers, respectively. The thresholds T_1, T_2 are selected such that the Kullback-Leiber divergence of the resulting approximation is no greater than a target value.

C. RELIABILITY FUNCTION

The reliability is based on the likelihood of the soft check-to-variable message $\widetilde{\mathbf{b}}_0 = -[h]\ (\mathbf{m}_c - \mathbf{m}_a)$ given its nearest integer approximation $\mathbf{b}_0$, as given by (21). When the channel is noiseless, $\widetilde{\mathbf{b}}_0$ will be equal to $\mathbf{b}_0$, and the highest likelihood value is obtained. The likelihood of $\widetilde{\mathbf{b}}_0$ given $\mathbf{b}_0$ can represent the inverse magnitude of the noise or the reliability of the message. If there is a large difference between $\widetilde{\mathbf{b}}_0$ and $\mathbf{b}_0$, it means that the message has low reliability or high noise magnitude. The likelihood is:

$$\Pr\left(\widetilde{\mathbf{b}}_0|\mathbf{b}_0\right) = \frac{1}{2\pi\sqrt{|\mathbf{V}|}} e^{-\frac{1}{2}\left(\widetilde{\mathbf{b}}_0-\mathbf{b}_0\right)^T \mathbf{V}^{-1}\left(\widetilde{\mathbf{b}}_0-\mathbf{b}_0\right)}. \quad (25)$$

For any given Gaussian function with variance $\mathbf{V}$, $\widetilde{\mathbf{b}}_0$ and $\mathbf{b}_0$ are the only variables in (25), the other parameters are constant. We define ρ as the relevant part of the likelihood:

$$\rho = |\widetilde{\mathbf{b}}_0 - \mathbf{b}_0| = |(-[h]\ (\mathbf{m}_c - \mathbf{m}_a)) - \lceil -[h]\ (\mathbf{m}_c - \mathbf{m}_a) \rfloor|, \quad (26)$$

which satisfies $0 \leq \rho \leq 1$. A smaller value of the magnitude of the Gaussian noise means higher reliability, so the reliability is written as $1/\rho$. Fig. 1-(a) and (b) show three example check-to-variable messages r_1, r_2 and r_3 which have reliabilities ρ_1, ρ_2 and ρ_3, respectively; both Gaussian integers and Eisenstein integers are shown. $\rho_1 < \rho_2 < \rho_3$ represent low, intermediate and high intensity of the Gaussian noise, respectively.

D. THRESHOLD AND BOUND ON KULLBACK-LEIBLER DIVERGENCE

This subsection describes how to obtain the thresholds T_1, T_2, using the Kullback-Leibler (KL) divergence. In particular, the smallest number of Gaussian functions are chosen such that an estimate of the resulting KL divergence does not exceed a target value. The KL divergence between two Gaussian mixtures does not have a closed-form solution in general. Our approach is to form an upper bound on the KL divergence and use this to upper bound the thresholds, T_1, T_2. Then, linear regression is used as an approximation.

The KL divergence represents the similarity between two probability density functions (PDFs) [33], [34]. If the two PDFs are the same, the divergence is zero. Our main interest is the KL divergence between the infinite Gaussian mixture $Y(\mathbf{z})R(\mathbf{z})$ and the finite Gaussian mixture $Y(\mathbf{z})\widetilde{R}(\mathbf{z})$:

$$D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) = \int_{\mathbb{C}} Y(\mathbf{z})R(\mathbf{z}) \log \frac{Y(\mathbf{z})R(\mathbf{z})}{Y(\mathbf{z})\widetilde{R}(\mathbf{z})} d\mathbf{z}, \quad (27)$$

where $Y(\mathbf{z})$ is channel message with mean m_a and variance $\mathbf{V}_a$. $R(\mathbf{z})$ and $\widetilde{R}(\mathbf{z})$ are given in (19) and (20). The KL divergence of the Gaussian mixture approximation depends on 5 parameters: h, m_c, m_a, $\mathbf{V}_c$ and $\mathbf{V}_a$. If h, $\mathbf{V}_c$ and $\mathbf{V}_a$ are fixed, the KL divergence depends on $m_c - m_a$ only.

Let κ denote the maximum allowed KL divergence in (27) — we seek approximations with KL divergence not greater than κ. Without loss of generality, set $m_a = 0$ and $b_0 = 0$; then the KL divergence depends on m_c only. The region $\mathcal{R}$ are the values of m_c where the KL divergence is less than or equal to the target value κ, that is:

$$\mathcal{R} = \{m_c \in \mathbb{C} \mid KL(m_c) \leq \kappa\}. \quad (28)$$

The threshold T is the radius of the largest disc that is fully contained in $\mathcal{R}$. The region $\mathcal{R}$ is not a disc, but we restrict the effective region to be the disc $|m_c| \leq T$, for ease

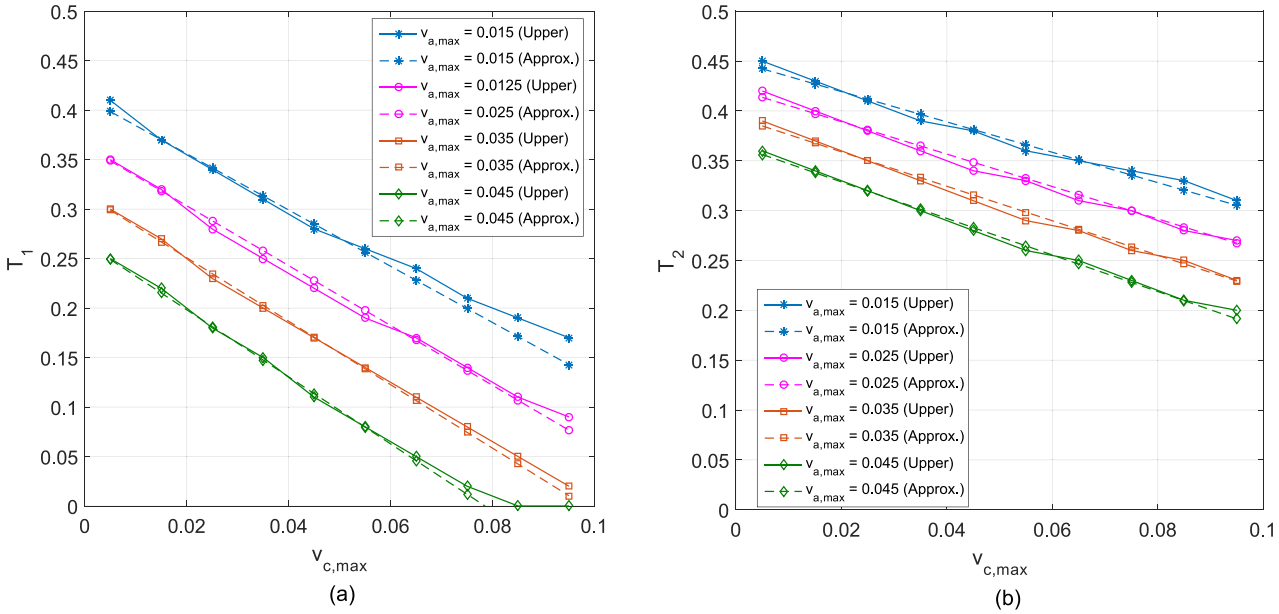


FIGURE 2. (a) T_1, and (b) T_2 of Gaussian integer approximation, for $k = 4$ and $\kappa = 10^{-2}$. Solid lines represent T_1 and T_2 calculated from the upper bound. Dashed lines represent T_1 and T_2 approximated by linear regression.

of computation. Interpret T as T_1 if $\widetilde{R}(\mathbf{z})$ is one Gaussian function, and as T_2 if $\widetilde{R}(\mathbf{z})$ is two Gaussian functions.

Next, in order to bound the KL divergence, assume that $\mathbf{V}_c$ and $\mathbf{V}_a$ have covariance zero, and the larger of the two variances is used as this has the greater effect on the KL divergence. Accordingly, the maximum element of the covariance matrix $\mathbf{V}_c$ and $\mathbf{V}_a$ is $v_{c,\max}$ and $v_{a,\max}$ respectively.

Since the target is to find the thresholds T_1 (one Gaussian function) and T_2 (two Gaussian functions), $\widetilde{R}(\mathbf{z})$ is an l-Gaussian mixture where l equals one or two. To form a bound, k Gaussian functions are selected from $R(\mathbf{z})$ where $k \in \mathbb{Z}^+$ (ideally, $k \to \infty$, but since most resulting Gaussian mixtures have near-zero mixing coefficients, it is more effective to consider k most relevant Gaussian functions). Using these assumptions, an upper bound on (27) can be formed.

Proposition 1: Let $R_k(\mathbf{z})$ consist of k Gaussian functions with mean m_c as in (19). If $\widetilde{R}(\mathbf{z})$ is either a single Gaussian function approximation or two-Gaussian function approximation, then:

$$D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) \leq \frac{(k-1)v_{a,max}}{a|h|^2\big(v_{c,max}+v_{a,max}\big)\left(1+e^{b\frac{2\left(m_{c,Re}h_{Re}-m_{c,Im}h_{Im}\right)-1}{2|h|^2\left(v_{c,max}+v_{a,max}\right)}}\right)^c}. \quad (29)$$

where $(a, b, c) = (2, -1, 1)$ for the single-Gaussian function approximation and $(a, b, c) = (1, 1, 2)$ for the two-Gaussian function approximation.

The proof is given in Appendix. This can be used to form an upper bound on T_1 and T_2. The value of T_1 and T_2 is $0 \leq \{T_1, T_2\} \leq 1/h$ because T_1 and T_2 are calculated from the check-to-variable message after stretching by $1/h$. We are interested in the value of m_c for which the bound attains κ; this value is T_1 $(l = 1)$ or T_2 $(l = 2)$. In (27), $D(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z}))$ has the worst case when $m_{c,Re} = m_{c,Im}$, so we assume $m_{c,Re} = m_{c,Im}$ in (29). For $l = 1, 2$ we have T_l as:

$$T_l = \frac{|h|}{(h_{Re}-h_{Im})}\left(\frac{1}{2}+\frac{1}{b}|h|^2\big(v_{a,max}+v_{c,max}\big)\log\left(\sqrt[c]{\left(\frac{(k-1)\big(v_{a,max}\big)}{a\kappa|h|^2\big(v_{c,max}+v_{a,max}\big)}\right)}-1\right)\right). \quad (30)$$

where (a, b, c) are as given in Proposition 1 for $l = 1$ single Gaussian function and $l = 2$ two-Guassian functions approximation. Note that from (26), $0 \leq \rho \leq 1$ applies to the check-to-variable message before stretching by $1/h$. So that T_1 and T_2 is comparable with ρ, scalar $|h|$ is included in (30).

While (30) is not linear, we observed that it is roughly linear for parameters of interest, so linear regression is used to estimate T_1 and T_2. Fig. 2 (a) and (b) show T_1 and T_2 obtained from the upper bound and its linear approximation, for Gaussian integers. The linear approximation equation of T_1 and T_2 used for decoding is given in (31)

$$T = p_1 - p_2 v_{a,\max} - p_3 v_{c,\max} - p_4 v_{a,\max} v_{c,\max}, \quad (31)$$

where the set of coefficient $\{p_1, p_2, p_3, p_4\}$ is provided in Table 2.

The constant value κ is set at 10^{-2}. This value was chosen to give a favorable performance-complexity trade-off, that is, it is the largest value of κ for which there is no noticeable degradation in decoder symbol error rate (SER), as determined by substantial simulations.

For the number of Gaussian functions k in the infinite Gaussian mixture $R(\mathbf{z})$, we selected $k = 4$ and 3 for Gaussian

TABLE 1. Approximation of T_1 and T_2 obtained using linear regression.

Threshold	Gaussian integer $\{p_1, p_2, p_3, p_4\}$	Eisenstein integer $\{p_1, p_2, p_3, p_4\}$
T_1	$\{0.43, 4.96, 0.64, 24.04\}$	$\{0.44, 5.65, 1.24, 10.34\}$
T_2	$\{0.61, 7.81, 1.48, 0.99\}$	$\{0.43, 5.56, 1.05, -4.29\}$

integer and Eisenstein integer, respectively. ($k = 4$ and 3 are sufficient to accurately approximate the infinite Gaussian mixtures because these are the number of neighbors of a deep hole for Gaussian integer and Eisenstein integer; see the explanation at the end of Section III-A.) In order to implement a practical algorithm, the same threshold for all check-to-variable messages independently of coefficient h is desired, so $h = 1$ was selected because this gives the smallest (i.e., most pessimistic) threshold value.

E. GAUSSIAN APPROXIMATION AT VARIABLE NODE

This subsection describes the approximation at the variable node in Section II-C. Unlike existing decoding algorithms based on real and complex numbers [8], [9], [11], [13], [14], [15], [16], the variable node function of the proposed algorithm adaptively selects the number of Gaussian functions using the reliability of the check-to-variable message. At each iteration, the variable node function is as follows.

- *Step 1, calculate reliability:* At a variable node, the input messages are $R_i(\mathbf{z})$ with mean $\mathbf{m}_i$ and variance $\mathbf{V}_i$, for $i = 1, 2, \ldots, d$. Find the corresponding reliabilities $\rho_1, \rho_2, \ldots, \rho_d$ using (26).
- *Step 2, find the number of Gaussian functions in message expansion:* For $i = 1, 2, \ldots, d$, the reliability ρ_i of each message is compared to the threshold T_1 and T_2. If the reliability $\rho_i \leq T_1$, then the number of Gaussian functions in the message expansion is $|\mathcal{B}_i| = 1$. If $T_1 < \rho_i \leq T_2$, then $|\mathcal{B}_i| = 2$, otherwise, $|\mathcal{B}_i| = 3$ or 4 for Gaussian or Eisenstein integers, respectively.
- *Step 3, message expansion:* Each message $R_1(\mathbf{z}), R_2(\mathbf{z}), \ldots, R_d(\mathbf{z})$ is expanded to a periodic function using $|\mathcal{B}_i|$ Gaussian functions from step 2. The periodic Gaussian function is $\widetilde{R}_i(\mathbf{z}) = \sum_{b \in \mathcal{B}_i} \mathcal{N}(\mathbf{z}; \mathbf{m}_i + [h_i]^{-1}\mathbf{b}, \mathbf{V}_i)$. The set $\mathcal{B}_i$ for Gaussian and Eisenstein integers is found in Section III-A.

Note that a forward-backward recursion is applied at the variable node to reduce the number of operations inside the variable node. This recursion is similar to the existing forward-backward recursion of [15] (real-valued case) but it is different [11] (complex-valued case) in how the channel value $\mathbf{y}_i$ is handled, in [15] the channel message $\mathbf{y}_i$ is multiplied at the last step of the recursion while [11] the channel message $\mathbf{y}_i$ is multiplied at the first step of the recursion.

Finally, we summarize all the steps of CLDLC reliability-based BP decoding algorithm in Algorithm 1. This algorithm illustrates clearly that the contributions of this work are in the message expansion part of variable nodes.

Algorithm 1 CLDLC Reliability-Based BP Decoder

Require: 1) Received messages from the CAWGN channel which are the initial variable-to-check messages $y_i(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{y}_i, \sigma^2\mathbf{I}_2)$ in (7), where $i = 1, 2, \ldots, n$. 2) Preparing the threshold T_1 and T_2 lookup table for different value of variance. This can be computed from (31) and Table 1.

Ensure: Estimate the integers $\hat{\overline{\mathbf{b}}}$ in (12).

$\hat{\overline{\mathbf{b}}} \leftarrow \overline{\mathbf{b}}$

***Compute initialization*:**

{

The check node incoming messages are $q_k(\mathbf{z})$, where $k = 1, \ldots, d-1$ single Gaussian functions and output is $k = d$, then $q_k(\mathbf{z}) = y_k(\mathbf{z})$.

}

while iter $\neq$ max.iter **do**

***Compute check-to-variable messages*:**

{

$R_d(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_d, \mathbf{V}_d)$ in (8)

}

***Compute variable-to-check message*:**

{

The proposed approximation at the variable node

{

Step 1, calculate check-to-variable message reliability:
Find the corresponding reliability ρ_i of $R_i(\mathbf{z})$ in (26)
,where $i = 1, 2, \ldots, d$

Step 2, find the number of Gaussian functions in message expansion:
if $\rho_i \leqslant T_1$ **then**
the message expansion, $|\mathcal{B}_i| = 1$
else if $T_1 < \rho_i \leqslant T_2$ **then**
$|\mathcal{B}_i| = 2$
else
$|\mathcal{B}_i| = 4$ for Gaussian integers
$|\mathcal{B}_i| = 3$ for Eisenstein integers
end if

Step 3, message expansion:
from $|\mathcal{B}_i|$ in step 2, the message can be expanded as:
$\widetilde{R}_i(\mathbf{z}) = \sum_{b \in \mathcal{B}_i} \mathcal{N}(\mathbf{z}; \mathbf{m}_i + [h_i]^{-1}\mathbf{b}, \mathbf{V}_i)$.
The set $\mathcal{B}_i$ for Gaussian and Eisenstein integers is found in Section III-A.

}

After expanding $R_k(\mathbf{z})$ to $\widetilde{R}_k(\mathbf{z})$,
find the message $q_d(\mathbf{z})$ sent back to the check node,
$f_d(\mathbf{z}) = y_i(\mathbf{z}) \prod_{k=1}^{d-1} \widetilde{R}_k(\mathbf{z})$ and $q_d(\mathbf{z}) = MM(f_d(\mathbf{z}))$

}

end while

***Final decision*:**

{

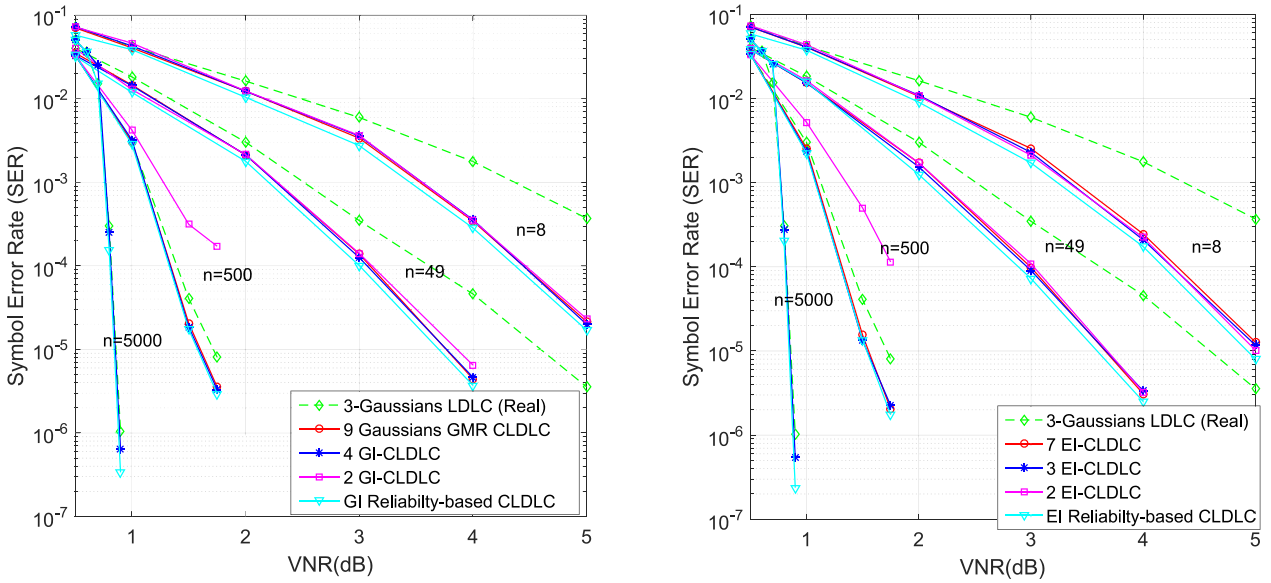
$q_i^{final}(\mathbf{z}) = y_i(\mathbf{z}) \prod_{k=1}^{d} \widetilde{R}_k(\mathbf{z})$

$\hat{\mathbf{x}}_i = \arg\max_{\mathbf{z}} q_i^{final}(\mathbf{z})$, and $\hat{\overline{\mathbf{b}}} = \lfloor \mathbf{H}\hat{\mathbf{x}} \rceil$

IV. NUMERICAL RESULTS

A. ERROR RATE FOR RELIABILITY-BASED DECODER

The error rate of the reliability-based parametric CLDLC decoder was evaluated on the unconstrained input power CAWGN channel. We compared five decoding algorithms: 1) real-valued LDLC with the 3 Gaussian function decoder [15], 2) CLDLC based on GMR algorithm with



(a) CLDLC based on Gaussian integer

(b) CLDLC based on Eisenstein integer

FIGURE 3. The performance comparison in terms of VNR vs symbol error rate of (a) GI-CLDLC, and (b) EI-CLDLC.

9 Gaussian functions [11]; CLDLC decoding with a fixed number of Gaussian functions: 3) 4 Gaussian functions (4 GI-CLDLC) and 4) 2 Gaussian functions CLDLC (2 GI-CLDLC), which are the extension of real-valued decoding [15] to the complex case. Finally, 5) the proposed reliability-based CLDLC decoder. Here, complex-valued lattices with dimension $n = 8$, 49, 500, and 5,000 were used, and the matrix $\mathbf{H}$ had row and column degrees d of 3, 3, 5, and 7, respectively. The elements of the GI and EI vector $\overline{\mathbf{b}}$ in equation (1) are randomly chosen from the set of integers $b_{Re}, b_{Im} \in \mathcal{Z}$, where $\mathcal{Z} = \{-10, -9, \ldots, 9, 10\}$. For real-valued LDLC lattices, dimensions $n = 16$, 100, 1,000, and 10,000 were used, and the row and column degrees are the same as CLDLC and the elements of the real integer vector $\overline{\mathbf{b}}$ are randomly chosen from $\mathcal{Z}$. The inverse generator matrix was created with the generator sequence $\mathcal{H} = \{1, \frac{1}{\sqrt{d}}, \ldots, \frac{1}{\sqrt{d}}\}$, and the matrix was normalized such that $V(\Lambda) = 1$. For 9 Gaussian functions with GMR algorithm, we follow the settings of [11] using M = 10,K = 3,$VarRangeLen = 0.4$, and $CheckRangeLen = 0.05$, where M is the number of Gaussian functions in each list, K is the number of replications, *VarRangeLen* and *CheckRangeLen* denote each axis grouping range length. At each VNR, we simulated until the number of symbol errors and word errors reach 1,000 and 500 at least (must satisfy both conditions). The number of iterations t of BP decoding is 50 iterations.

Fig. 3 (a) shows the symbol error rate of each decoding algorithm for CLDLC based on Gaussian integers (a symbol error occurs when $\widehat{b_i} \neq b_i$). The result shows that the reliability-based decoding gives the best performance when $n \leq 500$. In addition, CLDLC outperforms real-valued LDLC when $n \leq 500$. For $n = 5000$, all CLDLC decoding algorithms based on Gaussian integers yield the same performance as the real-valued LDLC.

Fig. 3 (b) shows the symbol-error rate of decoding CLDLC based on Eisenstein integers. We also compared five algorithms: 1) 3 Gaussian functions LDLC, 2) 7 EI-CLDLC, 3) 3 EI-CLDLC, 4) 2 EI-CLDLC, and 5) EI reliability-based CLDLC decoder. The general tendency of the results are similar to CLDLC based on Gaussian integers. The reliability-based decoding algorithm based on Eisenstein integers gives the best performance among four CLDLC decoding algorithms, and EI-CLDLC outperforms the real-valued LDLC when $n \leq 500$. For $n = 5000$, all CLDLC decoding algorithms based on Eisenstein integers show the same performance as the real-valued LDLC.

The reason that when $n \leq 500$, CLDLC outperforms the real-valued LDLC, and for $n = 5000$ CLDLC shows the same performance as the real-valued LDLC was explained in Section II-D of [11] by Yona and Feder. In CLDLC, each narrow variance message (variance approach 0) in the check node has a mean that converges to a specific lattice point, known as hypothesis. When this narrow variance message is multiplied by the wide variance messages in the variable node, this process removes irrelevant hypothesis, aiding convergence to the correct lattice point. The wide variance messages alignment in 2-dimensional messages of CLDLC makes it easier to remove irrelevant hypothesis and converge to the correct hypothesis. For $n \leq 500$, the CLDLC alignment is better than the real-valued LDLC alignment, which leads to the performance improvement. However, when the lattice dimension increases, the CLDLC advantage becomes negligible because there is a good alignment even in the real-valued LDLC.

B. RELIABILITY-BASED DECODER COMPLEXITY

The complexity of the proposed reliability-based decoding algorithm and existing algorithms are described in this subsection.

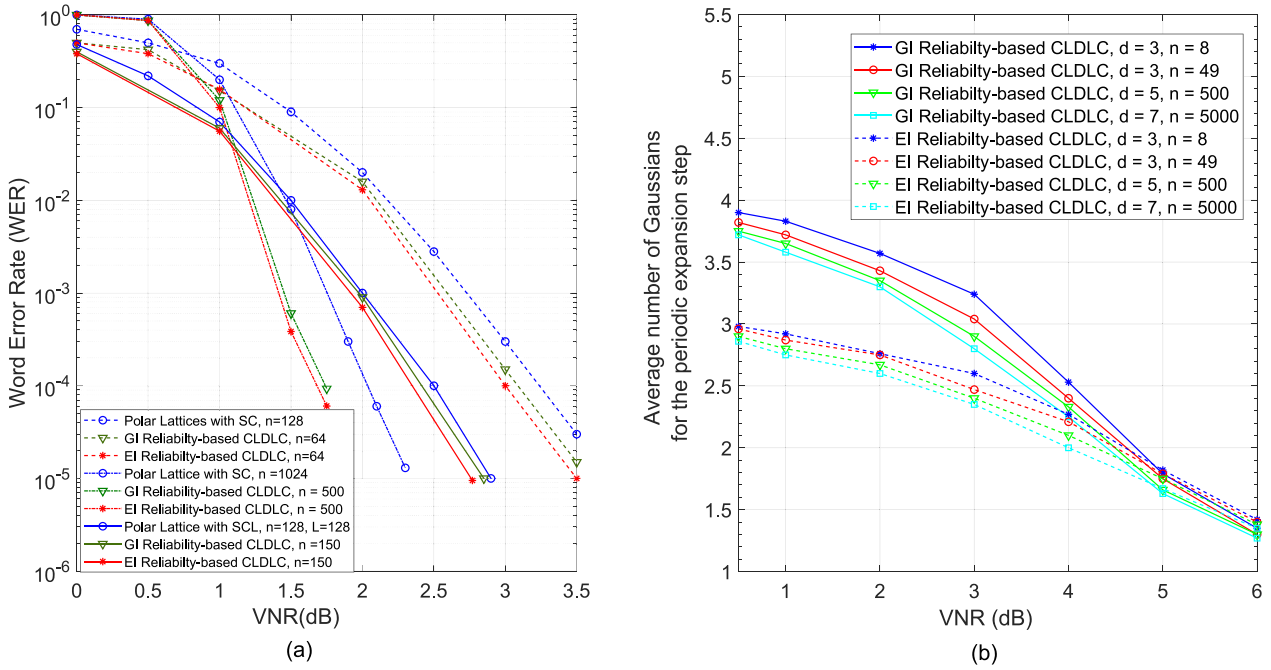


FIGURE 4. (a) The performance comparison between GI and EI reliability-based CLDLC with Polar lattices [25] in terms of VNR vs word error rate. (b) Average number of Gaussian functions $|\mathcal{B}|$ of the reliability-based parametric decoder in the periodic expansion step vs VNR for each lattice dimension, solid lines and dash lines show GI-CLDLC and EI-CLDLC, respectively.

For GMR decoding of CLDLC lattices [11], the storage requirement needed after the GMR algorithm is $\mathcal{O}(n \cdot d \cdot M)$. The computational complexity is $\mathcal{O}(n \cdot d \cdot t \cdot K^2 \cdot M^3)$, and is dominated by sorting and searching in tables, where n is the lattice dimension, d is the degree of the inverse generator matrix, t is the number of iterations, K is the number of replications and M is the number of Gaussian functions in each list. $K = 3$ and $M = 10$ were used in [11].

The three/two Gaussian functions decoding algorithm which is proposed by [15] reduces the number Gaussian mixtures in LDLC (real number case). It is straightforward to extend this algorithm to the complex case; we call this GI-CLDLC and EI-CLDLC decoding. The computational complexity of GI-CLDLC are $\mathcal{O}(n \cdot t \cdot 2^{d-1})$ and $\mathcal{O}(n \cdot t \cdot 4^{d-1})$ for 2 Gaussian function replications and 4 Gaussian function replications, respectively. For EI-CLDLC, the complexity are $\mathcal{O}(n \cdot t \cdot 2^{d-1})$ and $\mathcal{O}(n \cdot t \cdot 3^{d-1})$ for 2 Gaussian function replications and 3 Gaussian function replications, respectively. After MM algorithm, the storage requirement needed is $5 \cdot n \cdot d$ because the message passed between the check and variable nodes are single complex Gaussian functions which are represented by five parameters, 2×1 mean vector $\mathbf{m}$ and 2×2 covariance matrix $\mathbf{V}$.

The average computational complexity of the proposed reliability-based decoding algorithm is $\mathcal{O}(n \cdot t \cdot |\widetilde{\mathcal{B}}|^{d-1})$, where $|\widetilde{\mathcal{B}}|$ is an average number of Gaussian functions in the periodic expansion step. This complexity analysis assumes that $|\mathcal{B}|$ is independent for each edge. While not strictly independent, the sparse graph is locally tree-like, making this a good approximation. After the MM algorithm, the storage requirement is $5 \cdot n \cdot d \cdot |\widetilde{\mathcal{B}}|$ as a function of the 3 parameters n, d and VNR is shown in Fig. 4 (b). For example, for a fixed VNR, we can see that $|\widetilde{\mathcal{B}}|$ decreases when n and d increase. On the other hand, for a fixed n and d, $|\widetilde{\mathcal{B}}|$ decreases when VNR increases. The mean of $|\widetilde{\mathcal{B}}|$ ranges from 3.75 (at VNR $= 0.5$ dB) to 1.35 (at VNR $= 6$ dB) Gaussian functions on average for GI reliability-based, and 2.9 (at VNR $= 0.5$ dB) to 1.38 (at VNR $= 6$ dB) Gaussian functions on average for EI reliability-based. Significantly, when GI and EI reliability-based decoding are compared, the EI decoder has lower complexity at $n = 49$, 500 and 5000, when the VNR range is 1–4 dB. This is due to the tighter packing of the Eisenstein integers which allows fewer Gaussian functions for a good approximation, as was discussed earlier. The computational complexity of decoding algorithms in Fig. 3 (a) and (b) are summarized in Table 2.

The computational complexity based on $\mathcal{O}$ notation in Table 2 also can represent the time complexity and tell us whether the decoding algorithms have shorter or longer computation times compared to each other. Besides the time complexity based on $\mathcal{O}$ notation, we also represent the computation time of CLDLC and LDLC based on computer simulation for visualization or providing an image that which algorithms provide shorter or longer computation time compared to each other. These computation run-times are shown Fig. 5 (a) and (b) which are implemented by MATLAB2020a on a desktop with Intel Core I7-9700 3.0 GHz CPU, 32 GB RAM, and Windows 10 operating system. The run-time might be different for other programming languages, CPU, RAM, and operating systems, however, in terms of related run-times to each other (shorter or longer), this run-time would not change. The computation time of the

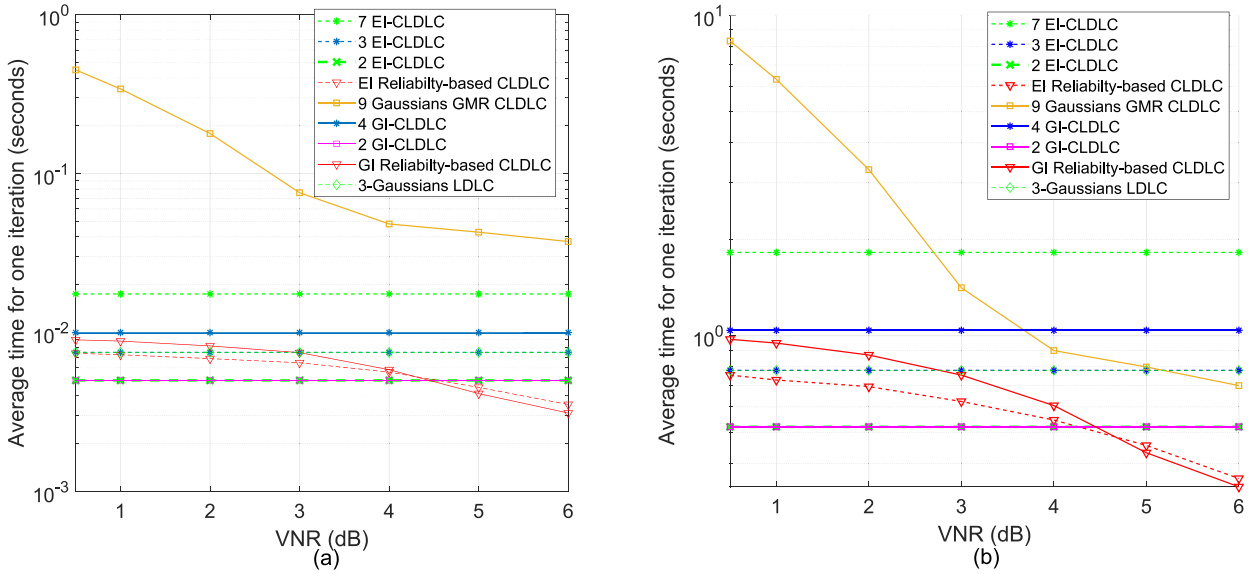


FIGURE 5. (a) Time comparison between 9 Gaussian functions GMR, 4 GI-CLDLC, 2 GI-CLDLC, GI reliability-based CLDLC, 7 EI-CLDLC, 3 EI-CLDLC, 2 EI-CLDLC, EI reliability-based CLDLC and 3 Gaussian functions LDLC for CLDLC dimension $n = 8$ and degree $d = 3$, and (b) for CLDLC dimension $n = 500$ and degree $d = 5$.

TABLE 2. Computational complexity of CLDLC and LDLC decoders.

Decoding algorithms	Computational complexity	Notes
9 Gaussians GMR CLDLC	$\mathcal{O}(n{\cdot}d{\cdot}t{\cdot}K^2{\cdot}M^3)$	$n = \{8, 49, 500\}$ $d = \{3, 3, 5\}, t = 50$ $K = 3,\ M = 10$
4 GI-CLDLC	$\mathcal{O}(n \cdot t \cdot 4^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$
2 GI-CLDLC	$\mathcal{O}(n \cdot t \cdot 2^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$
GI Reliability-based CLDLC	$\mathcal{O}(n \cdot t \cdot \lvert\widetilde{\mathcal{B}}\rvert^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$ see Fig. 4 (b) for $\lvert\widetilde{\mathcal{B}}\rvert$
7 EI-CLDLC	$\mathcal{O}(n \cdot t \cdot 7^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$
3 EI-CLDLC	$\mathcal{O}(n \cdot t \cdot 3^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$
2 EI-CLDLC	$\mathcal{O}(n \cdot t \cdot 2^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$
EI Reliability-based CLDLC	$\mathcal{O}(n \cdot t \cdot \lvert\widetilde{\mathcal{B}}\rvert^{d-1})$	$n = \{8, 49, 500, 5000\}$ $d = \{3, 3, 5, 7\}, t = 50$ see Fig. 4 (b) for $\lvert\widetilde{\mathcal{B}}\rvert$
3 Gaussians LDLC (Real)	$\mathcal{O}(n \cdot t \cdot 3^{d-1})$	$n = \{16, 100,$ $1000, 10000\}$ $d = \{3, 3, 5, 7\}, t = 50$

Note 1: $\lvert\widetilde{\mathcal{B}}\rvert$ changes which depends on lattice dimension n and VNR which is shown in Fig. 4 (b).

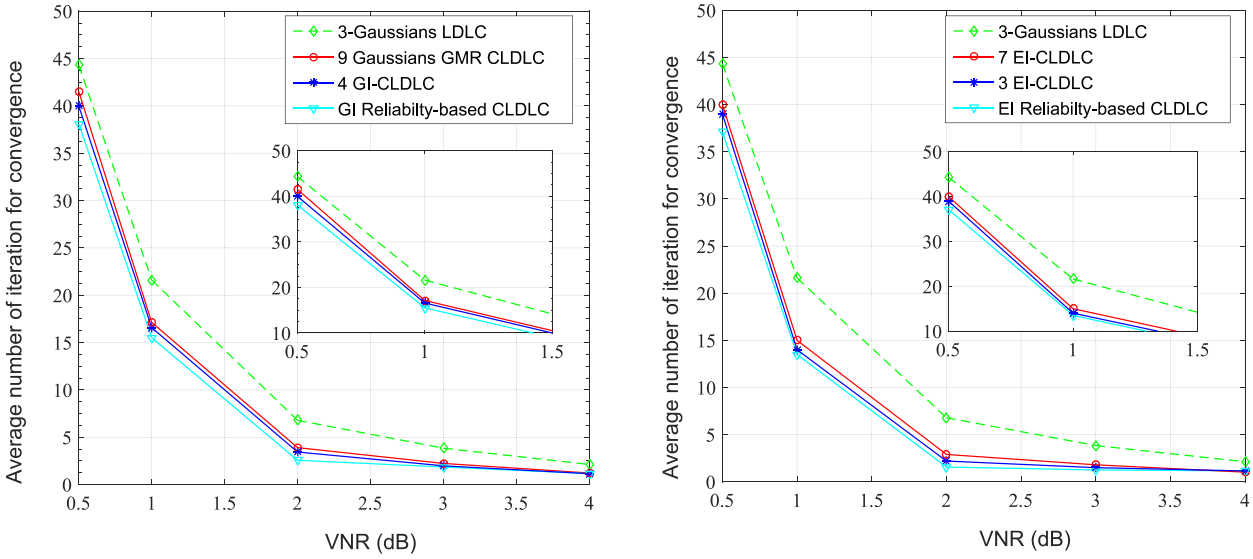
reliability-based decoder based on Gaussian integers and Eisenstein integers, 9 Gaussian functions GMR algorithm, 4 GI-CLDLC, 2 GI-CLDLC, 7 EI-CLDLC, 3 EI-CLDLC, 2 EI-CLDLC, and 3 Gaussian functions LDLC are shown here. We consider two lattice dimensions $n = 8$ and 500 with degree $d = 3$ and 5, respectively. For $n = 8$, 2 GI-CLDLC and 2 EI-CLDLC give comparable performance to other algorithms, as was shown in Fig. 3 (a) and (b), and also give the lowest computation time when VNR < 4.5 dB. When VNR ≥ 4.5 dB reliability-based decoders provide the lowest computation time.

For $n = 500$, Fig. 5 shows that 2 GI-CLDLC and 2 EI-CLDLC give the lowest computation time when VNR < 4.5 and 4.25 dB, respectively. However, the 2 Gaussian functions expansion loses performance when $n \geq 49$, as was shown in Fig. 3 (a) and (b). Therefore, reliability-based decoders provide the lowest computation time that also have good decoder error rates when $n \geq 49$. In addition, if we compare GI reliability-based decoding and EI reliability-based decoding, EI shows lower computation time when VNR ≤ 4.6 dB, again because the EI integers provide a good approximation with fewer Gaussian functions.

In Fig. 6, the average number of iterations required for decoder convergence is shown. We took a sample of 10,000 converged codewords (non-converging codewords are ignored) and evaluate the mean of the number of iterations required. The number of iterations reduces when the VNR increases. Fig. 6 (a) and (b) show the number of iterations for GI decoding and EI decoding, respectively. The complex-valued LDLC decoder needs fewer iterations for convergence compared to the real-valued LDLC. Both GI and EI reliability-based decoding require the fewest iterations for convergence.

C. COMPARISON OF ERROR RATE AND DECODER COMPLEXITY BETWEEN CLDLC AND POLAR LATTICES

Fig. 4 (a) shows word error rate (WER) comparison between GI and EI reliability-based CLDLC and Polar lattices with the SC [25], [26] and SCL decoding algorithms [26]. For



(a) CLDLC based on Gaussian integer

(b) CLDLC based on Eisenstein integer

FIGURE 6. Average number of iterations required for decoder convergence in terms of VNR, for CLDLC dimension $n = 500$ and degree $d = 5$, (a) GI-CLDLC and (b) EI-CLDLC.

Polar lattices under SC decoding, we consider $n = 128$ [26] and $n = 1024$ [25]. For Polar lattices under SCL decoding, we consider $n = 128$ and list size $L = 128$ [26]. We consider GI and EI reliability-based CLDLC $n = 64, 150, 500$ and $d = 3$ to compare the performance and complexity with Polar lattices. For GI reliability-based CLDLC, $|\widetilde{\mathcal{B}}|$ of $n = 64$ ranges from 3.7 (at VNR = 1 dB) to 2.7 (at VNR = 3.5 dB). $|\widetilde{\mathcal{B}}|$ of $n = 150$ ranges from 3.65 (at VNR = 1 dB) to 2.95 (at VNR = 3 dB). $|\widetilde{\mathcal{B}}|$ of $n = 500$ ranges from 3.62 (at VNR = 1 dB) to 3.35 (at VNR = 2 dB). For EI reliability-based CLDLC, $|\widetilde{\mathcal{B}}|$ of $n = 64$ ranges from 2.8 (at VNR = 1 dB) to 2.3 (at VNR = 3.5 dB). $|\widetilde{\mathcal{B}}|$ of $n = 150$ ranges from 2.76 (at VNR = 1 dB) to 2.4 (at VNR = 3 dB). $|\widetilde{\mathcal{B}}|$ of $n = 500$ ranges from 2.74 (at VNR = 1 dB) to 2.7 (at VNR = 2 dB).

For the same lattice dimensions, Polar lattices $n = 128$ compared to CLDLC $n = 64$ and Polar lattices $n = 1024$ compared to CLDLC $n = 500$, CLDLC outperforms Polar lattices under SC decoding while having higher decoding complexity. However, when compared to Polar lattices under SCL decoding, CLDLC provides worse performance and this also comes with lower complexity. The overall decoding complexity of SC and SCL decoding is $\mathcal{O}(n \log_2 n)$ and $\mathcal{O}(L \cdot n \log_2 n)$, respectively. To compare the complexity between CLDLC and Polar lattices, CLDLC and Polar lattices must have close WER performance to each other. There are two options for this. The first one is decreasing the list size L of SCL decoding and WER of Polar lattices will be close to CLDLC. The other way is increasing the lattice dimensions of CLDLC and WER of CLDLC will be close to Polar lattices. We selected the second option which is changing the lattice dimensions of CLDLC because the implementation and result of another list size L for Polar lattices under SCL decoding are not provided in [26], and it is not in the scope of this work to implement it. We selected the CLDC lattices $n = 150$ and $d = 3$ which provide nearly the same WER performance as Polar lattices with SCL decoding. The result shows that EI reliability-based CLDLC provides lower complexity than Polar lattices based on SCL decoding which have $n = 128$ and $L = 128$ while GI reliability-based CLDLC provides higher complexity.

V. CONCLUSION

We proposed the construction of CLDLC based on Eisenstein integers, to reduce the complexity of CLDLC decoding. We defined the reliability of the check-to-variable messages for choosing the number of finite Gaussian functions for each incoming message at the variable node. Each incoming message at the variable node can be approximated by a varying number of finite Gaussian functions, depending on its reliability. Therefore, the number of finite Gaussian functions will be minimized for each incoming message, reducing complexity. The reliability-based decoding algorithm was applied to CLDLC based on Gaussian integers as well.

Our results show that the reliability-based decoding algorithm for Eisenstein integers gives the lowest complexity when $n \geq 49$. In addition, reliability-based decoding algorithms based on both Eisenstein integers and Gaussian integers shows the best performance when $n \leq 500$. Eisenstein integers provides lower complexity than Gaussian integers because the hexagonal Voronoi cells of the Eisenstein integer lattice has the tightest packing in two dimensions, leading to a higher reliability than Gaussian integers for the same fixed number of Gaussian functions in the approximation.

APPENDIX
PROOF OF PROPOSITION 1

The proof of Proposition 1 is given. The check-to-variable messages $R(\mathbf{z})$ and $\widetilde{R}(\mathbf{z})$ have mean $\mathbf{m}_c$, variance $\mathbf{V}_c$, where

the k-Gaussian mixture $R(\mathbf{z})$ expressed as:

$$R(\mathbf{z}) = \sum_{i=0}^{k-1} \mathcal{N}(\mathbf{z}; \mathbf{m}_c + \mathbf{b}_i/h, \mathbf{V}_c) \tag{32}$$

and the l-Gaussian function approximation is $\widetilde{R}(\mathbf{z})$, expressed as:

$$\widetilde{R}(\mathbf{z}) = \sum_{j=0}^{l-1} \mathcal{N}\big(\mathbf{z}; \mathbf{m}_c + \mathbf{b}_j/h, \mathbf{V}_c\big). \tag{33}$$

The channel message has mean $\mathbf{m}_a$, variance $\mathbf{V}_a$, expressed as $Y(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{m}_a, \mathbf{V}_a)$. And it is assumed that $\mathbf{V}_c$ and $\mathbf{V}_a$ are diagonal covariance matrices.

Define f and g as $Y(\mathbf{z})R(\mathbf{z})$ and $Y(\mathbf{z})\widetilde{R}(\mathbf{z})$, respectively, and these products are:

$$\begin{aligned} f = Y(\mathbf{z})R(\mathbf{z}) &= \sum_{i=0}^{k-1} \pi_i \underbrace{\mathcal{N}(\mathbf{z}; \mathbf{m}_i, \mathbf{V})}_{f_i}, \text{ and} \\ g = Y(\mathbf{z})\widetilde{R}(\mathbf{z}) &= \sum_{j=0}^{l-1} \omega_j \underbrace{\mathcal{N}\big(\mathbf{z}; \mathbf{m}_j, \mathbf{V}\big)}_{g_j}, \end{aligned} \tag{34}$$

where $\mathbf{V}$, $\mathbf{m}_i$, π_i and ω_j are:

$$\begin{aligned} \mathbf{V} &= \Big(\mathbf{V}_c^{-1} + \mathbf{V}_a^{-1}\Big)^{-1}, \mathbf{m}_i = \mathbf{V}\Big(\mathbf{V}_c^{-1}(\mathbf{m}_c + \mathbf{b}_i/h)\Big), \\ \pi_i' &= \frac{1}{2\pi\sqrt{|\mathbf{V}_c + \mathbf{V}_a|}} e^{-\frac{1}{2}(\mathbf{m}_c+\mathbf{b}_i/h)^T(\mathbf{V}_c+\mathbf{V}_a)^{-1}(\mathbf{m}_c+\mathbf{b}_i/h)}, \\ \pi_i &= \frac{\pi_i'}{\sum_{i=0}^{k-1}\pi_i'}, \text{ and } \omega_j = \frac{\pi_j'}{\sum_{j=0}^{l-1}\pi_j'}. \end{aligned} \tag{35}$$

Without loss of generality, $\pi_0' \geq \pi_1' \geq \cdots \geq \pi_{k-1}'$ and $\mathbf{m}_a = 0$ is assumed, which implies $|\mathbf{m}_0| \leq |\mathbf{m}_1| \leq \cdots \leq |\mathbf{m}_{k-1}|$.

Then, the KL divergence between $Y(\mathbf{z})R_k(\mathbf{z})$ and $Y(\mathbf{z})\widetilde{R}(\mathbf{z})$ can be expressed as:

$$D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) = D\left(\sum_{i=0}^{k-1}\pi_i f_i || \sum_{j=0}^{l-1}\omega_j g_j\right). \tag{36}$$

This can be upper bounded using the variational upper bound on KL divergence [34]:

$$D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) \leq \sum_{i=0}^{k-1}\sum_{j=0}^{l-1}\pi_i\omega_j D\big(f_i||g_j\big). \tag{37}$$

In general, the KL divergence between two complex Gaussian functions $\hat{f}$ and $\hat{g}$ has a closed-form expression,

$$\begin{aligned} D\big(\hat{f}||\hat{g}\big) = \frac{1}{2}\Bigg(\log\frac{|\mathbf{V}_{\hat{g}}|}{|\mathbf{V}_{\hat{f}}|} + \mathrm{Tr}\Big[\mathbf{V}_{\hat{g}}^{-1}\mathbf{V}_{\hat{f}}\Big] - 2 \\ + \Big(\mathbf{m}_{\hat{f}} - \mathbf{m}_{\hat{g}}\Big)^T\mathbf{V}_{\hat{g}}^{-1}\Big(\mathbf{m}_{\hat{f}} - \mathbf{m}_{\hat{g}}\Big)\Bigg). \end{aligned} \tag{38}$$

Here, f and g have the same variance $\mathbf{V}$. Then, $\mathbf{V}$ and $\mathbf{m}$ in (35) is substituted into (38), so that:

$$\pi_i\omega_j D\big(f_i||g_j\big) = \begin{cases} 0, & \text{if } i = j \\ \pi_i\omega_j\frac{v_a|\mathbf{m}_i - \mathbf{m}_j|^2}{2(v_c+v_a)}, & \text{if} i \neq j. \end{cases} \tag{39}$$

Equation (37) has $l(k-1)$ non-zero terms, and $\pi_0\omega_1 D(f_0||g_1) = \pi_1\omega_0 D(f_1||g_0)$ are equal and the greatest, so the term $\pi_0\omega_1$ determines the upper bound.

For single Gaussian function approximation $\omega_0 = 1$, and π_1 is given by:

$$\pi_1 = \frac{\pi_1'}{\sum_{i=0}^{k-1}\pi_i'} = \left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_1/h|^2}{2(v_c+v_a)}}}{\sum_{i=0}^{k-1} e^{-\frac{|\mathbf{m}_c+\mathbf{b}_i/h|^2}{2(v_c+v_a)}}}\right), \tag{40}$$

Now make the restriction to $k = 2$

$$\begin{aligned} \pi_1 &\leq \left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_1/h|^2}{2(v_c+v_a)}}}{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_0/h|^2}{2(v_c+v_a)}} + e^{-\frac{|\mathbf{m}_c+\mathbf{b}_1/h|^2}{2(v_c+v_a)}}}\right) \\ &= \frac{1}{1 + e^{\frac{-(\mathbf{m}_c h + \mathbf{m}_c^* h^*)+1}{2|h|^2(v_c+v_a)}}} \\ &= \frac{1}{1 + e^{\frac{-2\left(m_{c,Re}h_{Re} - m_{c,Im}h_{Im}\right)+1}{2|h|^2(v_c+v_a)}}} \end{aligned} \tag{41}$$

where $\mathbf{b}_0 = 0$ and $\mathbf{b}_1$ is any integer at the minimum Euclidean distance of 1, $|\mathbf{b}_1| = 1$, so without loss of generality, take $\mathbf{b}_1 = -1$. Then, (41) is substituted into (39), where $|\mathbf{m}_i - \mathbf{m}_j|^2$ is upper bounded by $|\mathbf{m}_i - \mathbf{m}_j|^2 \leq 1/|h|^2$. Form the upper bound from (37) by replacing the $(k-1)$ non-zero terms with the greatest term, so the upper bound of single Gaussian function approximation is:

$$\begin{aligned} &D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) \leq \\ &\frac{(k-1)v_a}{2|h|^2(v_c+v_a)|\left(1 + e^{\frac{2(m_{c,Re}h_{Re} - m_{c,Im}h_{Im})+1}{2|h|^2(v_c+v_a)}}\right)}. \end{aligned} \tag{42}$$

For two Gaussian function approximation, the term $\pi_0\omega_1$ can be written as:

$$\begin{aligned} \pi_0\omega_1 &= \frac{\pi_0'}{\sum_{i=0}^{k-1}\pi_i'}\frac{\pi_1'}{\sum_{j=0}^{l-1}\pi_j'} \\ &= \left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_0/h|^2}{2(v_c+v_a)}}}{\sum_{i=0}^{k-1} e^{-\frac{|\mathbf{m}_c+\mathbf{b}_i/h|^2}{2(v_c+v_a)}}}\right)\left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_1/h|^2}{2(v_c+v_a)}}}{\sum_{i=0}^{l-1} e^{-\frac{|\mathbf{m}_c+\mathbf{b}_i/h|^2}{2(v_c+v_a)}}}\right) \\ \pi_0\omega_1 &\leq \left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_0/h|^2}{2(v_c+v_a)}}}{\sum_{i=0}^{l-1} e^{-\frac{|\mathbf{m}_c+\mathbf{b}_i/h|^2}{2(v_c+v_a)}}}\right)^2. \end{aligned} \tag{43}$$

Make the restriction to $l = 2$

$$\begin{aligned} \pi_0\omega_1 &\leq \left(\frac{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_0/h|^2}{2(v_c+v_a)}}}{e^{-\frac{|\mathbf{m}_c+\mathbf{b}_0/h|^2}{2(v_c+v_a)}} + e^{-\frac{|\mathbf{m}_c+\mathbf{b}_1/h|^2}{2(v_c+v_a)}}}\right)^2 \\ &= \left(\frac{1}{1 + e^{\frac{2\left(m_{c,Re}h_{Re} - m_{c,Im}h_{Im}\right)-1}{2|h|^2(v_c+v_a)}}}\right)^2, \end{aligned} \tag{44}$$

where $\mathbf{b}_0 = 0$ and, $\mathbf{b}_1 = -1$. Then, (44) is substituted into (39), where $|\mathbf{m}_i - \mathbf{m}_j|^2 \leq 1/|h|^2$. Form the upper bound

from (37) by replacing the $2(k-1)$ non-zero terms with the greatest term, so the upper bound of two Gaussian function approximation is:

$$D\big(Y(\mathbf{z})R(\mathbf{z})||Y(\mathbf{z})\widetilde{R}(\mathbf{z})\big) \leq \frac{(k-1)v_a}{|h|^2(v_c+v_a)\left(1+e^{\frac{2(m_{c,Re}h_{Re}-m_{c,Im}h_{Im})-1}{2|h|^2(v_c+v_a)}}\right)^2}. \quad (45)$$

REFERENCES

[1] R. Zamir, *Lattice Coding for Signals and Networks: A Structured Coding Approach to Quantization, Modulation, and Multiuser Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[2] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 273–278, Jan. 1998.

[3] U. Erez and R. Zamir, "Achieving 1/2 log (1+SNR) on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[4] Y. Tian, D. Wu, C. Yang, and A. F. Molisch, "Asymmetric two-way relay with doubly nested lattice codes," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 694–702, Feb. 2012.

[5] Y. Huang, N. E. Tunali, and K. R. Narayanan, "A compute-and-forward scheme for Gaussian bi-directional relaying with inter-symbol interference," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1011–1019, Mar. 2013.

[6] D. Fang, Y.-C. Huang, Z. Ding, G. Geraci, S.-L. Shieh, and H. Claussen, "Lattice partition multiple access: A new method of downlink non-orthogonal multiuser transmissions," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–6.

[7] A. Hindy and A. Nosratinia, "Lattice coding and decoding for multiple-antenna Ergodic fading channels," *IEEE Trans. Commun.*, vol. 65, no. 5, pp. 1873–1885, May 2017.

[8] N. Sommer, M. Feder, and O. Shalvi, "Low-density lattice codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1561–1585, Apr. 2008.

[9] B. Kurkoski and J. Dauwels, "Reduced-memory decoding of low-density lattice codes," *IEEE Commun. Lett.*, vol. 14, no. 7, pp. 659–661, Jul. 2010.

[10] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[11] Y. Yona and M. Feder, "Complex low density lattice codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2010, pp. 1027–1031.

[12] J. Zhu and M. Gastpar, "Gaussian multiple access via compute-and-forward," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 2678–2695, May 2017.

[13] B. Kurkoski and J. Dauwels, "Message-passing decoding of lattices using gaussian mixtures," in *Proc. IEEE Int. Symp. Inf. Theory*, 2008, pp. 2489–2493.

[14] Y. Yona and M. Feder, "Efficient parametric decoder of low density lattice codes," in *Proc. IEEE Int. Symp. Inf. Theory*, 2009, pp. 744–748.

[15] R. A. Parrao Hernandez and B. M. Kurkoski, "The three/two gaussian parametric LDLC lattice decoding algorithm and its analysis," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3624–3633, Sep. 2016.

[16] S. Liu, Y. Hong, E. Viterbo, A. Marelli, and R. Micheloni, "Efficient decoding of low density lattice codes," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1195–1199, Aug. 2019.

[17] W. Wiriya and B. M. Kurkoski, "Reliability-based parametric LDLC decoding," in *Proc. Int. Symp. Inf. Theory Appl. (ISITA)*, 2018, pp. 188–192.

[18] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, vol. 290. New York, NY, USA: Springer, 2013.

[19] J. Freudenberger and S. Shavgulidze, "Signal constellations based on Eisenstein integers for generalized spatial modulation," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 556–559, Mar. 2017.

[20] S. Stern, D. Rohweder, J. Freudenberger, and R. F. H. Fischer, "Multilevel coding over Eisenstein integers with ternary codes," in *Proc. 12th Int. ITG Conf. Syst. Commun. Coding (SCC)*, 2019, pp. 1–6.

[21] N. E. Tunali, Y.-C. Huang, J. J. Boutros, and K. R. Narayanan, "Lattices over Eisenstein integers for compute-and-forward," *IEEE Trans. Inf. Theory*, vol. 61, no. 10, pp. 5306–5321, Oct. 2015.

[22] Y.-C. Huang, K. R. Narayanan, and P.-C. Wang, "Adaptive compute-and-forward with lattice codes over algebraic integers," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2015, pp. 566–570.

[23] Y.-C. Huang, K. R. Narayanan, and P.-C. Wang, "Lattices over algebraic integers with an application to compute-and-forward," *IEEE Trans. Inf. Theory*, vol. 64, no. 10, pp. 6863–6877, Oct. 2018.

[24] Y. Wang, A. Burr, and D. Fang, "Complex low density lattice codes to physical layer network coding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2015, pp. 2060–2065.

[25] L. Liu, Y. Yan, C. Ling, and X. Wu, "Construction of capacity-achieving lattice codes: Polar lattices," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 915–928, Feb. 2018.

[26] O. R. Ludwinianand, N. Liu, K. Anwar, and B. M. Kurkoski, "Design of polar code lattices of finite dimension," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2021, pp. 1011–1016.

[27] N. Di Pietro, G. Zémor, and J. J. Boutros, "LDA lattices without dithering achieve capacity on the Gaussian channel," *IEEE Trans. Inf. Theory*, vol. 64, no. 3, pp. 1561–1594, Mar. 2017.

[28] K. Huber, "Codes over Gaussian integers," *IEEE Trans. Inf. Theory*, vol. 40, no. 1, pp. 207–216, Jan. 1994.

[29] K. Huber, "Codes over Eisenstein-Jacobi integers," in *Contemporary Mathematics*, vol. 168. Providence, RI, USA: Am. Math. Soc., 1994, p. 165.

[30] D. Rohweder, J. Freudenberger, and S. Shavgulidze, "Low-density parity-check codes over finite Gaussian integer fields," in *Proc. IEEE Int.Symp. Inf. Theory (ISIT)*, 2018, pp. 481–485.

[31] Q. T. Sun, J. Yuan, T. Huang, and K. W. Shum, "Lattice network codes based on Eisenstein integers," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2713–2725, Jul. 2013.

[32] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994.

[33] C. Nafornita, Y. Berthoumieu, I. Nafornita, and A. Isar, "Kullback-Leibler distance between complex generalized gaussian distributions," in *Proc. 20th Eur. Signal Process. Conf. (EUSIPCO)*, 2012, pp. 1850–1854.

[34] J. R. Hershey and P. A. Olsen, "Approximating the Kullback-Leibler divergence between gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2007, pp. IV–317–IV–320.

WARANGRAT WIRIYA (Member, IEEE) was born in Thailand. She received the B.S. and M.S. degrees from the Department of Telecommunication Engineering, King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 2010 and 2012, respectively, and the Ph.D. degree from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST) in 2023. She currently works with Technology Strategy and Compliance Division, Rakuten Mobile, Japan. Then, she worked with the Research and Development Team of Read/Write Head in Seagate Technology Ltd., Thailand, from 2012 to 2016. She received a scholarship from National Electronics and Computer Technology Center, Thailand, from 2010 to 2012, while she was a M.S. Student at KMITL. From 2016 to 2019, during the Ph.D. Program at JAIST, she received two scholarships, the Doctoral Research Fellow from JAIST, and KDDI Foundation Scholarship.

BRIAN M. KURKOSKI (Member, IEEE) was born in Portland, OR, USA. He received the B.S. degree from the California Institute of Technology in 1993, and the M.S. and Ph.D. degrees from the University of California at San Diego in 2000 and 2004, respectively. He worked at Two California Startups. He is a Professor with the Japan Advanced Institute of Science and Technology (JAIST) in Nomi, Japan. He received a JSPS Postdoctoral Fellowship from 2004 to 2006, while the University of Electro-Communications in Tokyo, Japan, where he continued as an Associate Professor from 2007 to 2012. He has been with JAIST since 2012. He was an Associate Editor for *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* from 2010 to 2014. He was the Chair of the Data Storage Technical Committee, a Technical Committee of the IEEE Communications Society from 2017 to 2018, and was a Secretary from 2013 to 2016. For the IEEE Information Theory Society, he is a member of the Board of Governors from 2021 to 2023, and has been the Chair of the Digital Presence/Online Committee since 2019. He was a General Co-Chair of the 2021 IEEE Information Theory Workshop.