The Three/Two Gaussian Parametric LDLC Decoder

Ricardo Antonio Parrao Hernandez and Brian M. Kurkoski
School of Information Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan 923-1292
Email: ricardo.parrao@jaist.ac.jp, kurkoski@jaist.ac.jp

***Abstract*—Low density lattice codes (LDLC) can be decoded efficiently using iterative decoding, and approach the capacity of the AWGN channel. In the iterative LDLC decoder the messages are Gaussian mixtures. In any implementation, the Gaussian mixtures must be approximated. This work describes a three/two Gaussian parametric LDLC decoder. Internally at the variable node the periodic Gaussian mixtures are approximated with three or two Gaussians, while the messages between nodes are single Gaussians. The proposed approximation is more accurate than the previously-proposed approximation. Numerical results shows that for moderate dimension, e.g. $n = 1,000$, the two Gaussian approximation is sufficient for accurate performance. But for large dimension, e.g. $n = 10,000$, three Gaussians are needed.**

I. Introduction

Lattice codes are codes over the real numbers and can be seen as the Euclidean-space analog of linear codes. Shannon showed that codes with very long random Gaussian-distributed codewords can approach the AWGN capacity [1], and now it is known that lattice codes can also achieve the AWGN capacity [2] [3] [4].

Many high dimensional lattices can be constructed from codes. A practical way of constructing lattices is via Construction A [5]. Construction A considers mapping a p-ary code of length n into the Euclidean space. High dimensional lattices can be constructed by applying Construction A to low density parity check codes (LDPC), for example.

Another type of high dimensional lattice construction given by Sommer, Feder and Shalvi [6], called low density lattice codes (LDLC), has the property that the inverse generator matrix is sparse. The decoding of LDLC lattices can be implemented by using a belief propagation (BP) algorithm on a sparse graph. At a symbol error rate of 10^{-5}, the LDLC belief propagation decoder attains 0.6 dB from the unconstrained AWGN channel capacity [7].

Nazer and Gastpar [8] introduced a scheme called compute-and-forward based on physical-layer network coding (PLNC). Where there are multiple users and multiple relays, the idea is that the transmitted signals at each user are n-dimensional lattice points and each relay decodes an integer combination of these lattice points from the noisy observations, which is again a lattice point, instead of decoding the transmitted signals individually. Recent studies show that LDLC lattices are a suitable and practical approach for PLNC [9]. But how to efficiently and practically decode LDLC lattices is still an open question.

This work was supported by JSPS Kakenhi Grant Number 26289119.

In the LDLC belief propagation decoder the messages passed between check and variable nodes are continuous functions. In any implementation, these continuous functions must be approximated. In the original implementation [6], these messages were approximated by a discretely quantized function. The amount of quantization, typically 1024 bins, is impracticably large, but this gives the best-known performance. For the AWGN channel, the messages are precisely represented using a mixture containing an infinite number of Gaussians. This again is impractical, so it is natural to approximate this with a finite mixture of Gaussians.

Various parametric LDLC decoding algorithms have been proposed. In [10] a decoder using a Gaussian mixture reduction algorithm was introduced, where all possible pairs of Gaussians on a list are searched and the closest pairs are replaced with a single Gaussian. Further, using a single Gaussian as the message between the variable and check nodes leads to reduced memory requirements with a minor performance penalty [11]. This was followed by work by Yona and Feder [12], where the Gaussian mixture approximation is made by taking the dominating Gaussian in the mixture. This process is done by searching in tables, sorted in terms of the mixing coefficients. These relatively complicated processes of sorting and searching need to be performed at every message multiplication at the variable node. In [13] a single-Gaussian moment matching (SGMM) approximation was used internally at the variable node for every incoming message, and density evolution noise thresholds were presented. But finite-dimensional results were not given.

This paper presents a parametric decoding algorithm for LDLC lattices. In the proposed algorithm the infinite Gaussian mixtures are approximated with three or two Gaussians, which are nearby to the channel message. Accordingly, we call this the "three/two Gaussian parametric LDLC decoder". Approximations with a higher number of Gaussians improves the performance, but with a modest increase in complexity.

In this paper, we consider the accuracy of the approximation by evaluating the KL divergence. This gives insight into the performance-complexity tradeoff for LDLC decoding. The three/two Gaussian approximation is more accurate than the

SGMM used in [13]. The accuracy is confirmed using numerical simulations. For dimension $n = 1000$ the approximation using two Gaussians presents similar performance to the quantized algorithm. And for dimension $n = 10,000$ the approximation using three Gaussians presents better performance than the approximation using two Gaussians.

II. Background

A. Lattice and Low Density Lattice Codes

A lattice Λ is an additive subgroup of $\mathbb{R}^n$. A matrix $\mathbf{G}$, whose columns are the basis vectors, is called the generator of the lattice. A lattice point is defined as:

$$\mathbf{x} = \mathbf{G}\mathbf{b}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{Z}^n$ are column vectors.

A lattice codeword $\mathbf{x}$ is transmitted over the AWGN channel. Then it is received as:

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

where $\mathbf{z}$ is the additive Gaussian noise with 0 mean and variance σ^2, and $z_i \sim \mathcal{N}(0, \sigma^2)$ for $i = 1, 2 \ldots, n$.

A low density lattice code (LDLC), introduced by Sommer et al. [6], is an n-dimensional lattice code defined by a non-singular generator matrix $\mathbf{G}$ satisfying the condition that the inverse generator matrix $\mathbf{H} = \mathbf{G}^{-1}$ is sparse.

The non-zero elements of the inverse generator H are called the generator sequence h, with $h_1 \geq h_2 \geq \cdots \geq h_d > 0$ where d denotes the degree of the inverse generator matrix. The signs of the generator sequence entries are randomly changed to "$-$" with probability one-half.

B. Products of Gaussian mixtures

Let $f(w)$ be a mixture of N Gaussians,

$$f(w) = \sum_{i=1}^{N} c_i \mathcal{N}(w; m_i, v_i), \tag{3}$$

with mean m_i, variance v_i and mixing coefficients $c_i > 0$ for $i = 1, 2 \ldots, N$ and $\sum_{i=1}^{N} c_i = 1$.

The product of two Gaussian mixtures $f(w) = \sum_{i=1}^{N} f_i(w)$ and $g(w) = \sum_{j=1}^{M} g_j(w)$ is $f(w) \cdot g(w)$. The product of two components $f_i(w) = c_1 \mathcal{N}(w; m_1, v_1)$ and $g_j(w) = c_2 \mathcal{N}(w; m_2, v_2)$ is a single Gaussian $s(w) = c\mathcal{N}(w; m, v)$ with mean m, variance v and mixing coefficient c given by:

$$\frac{1}{v} = \frac{1}{v_1} + \frac{1}{v_2} \tag{4}$$

$$\frac{m}{v} = \frac{m_1}{v_1} + \frac{m_2}{v_2} \tag{5}$$

$$c = \frac{c_1 c_2}{\sqrt{2\pi(v_1 + v_2)}} e^{-\frac{(m_1 - m_2)^2}{2v_1 + 2v_2}}. \tag{6}$$

The Gaussian product $f(w) \cdot g(w)$ is the mixture of the $N \cdot M$ products obtained using the pair-wise operation above.

C. Moment Matching Approximation

The "moment matching approximation" is the single-Gaussian approximation of a Gaussian mixture $f(w)$, given by (3), with a single Gaussian $q(w) = \mathcal{N}(w; m, v)$ which minimizes the Kullback-Leibler divergence between $f(w)$ and $q(w)$. The moment-matching approximation (MM) finds the single Gaussian $q(w)$ which has the same mean m and variance v as $f(w)$. The mean m and variance v is given by:

$$m = \sum_{i=1}^{N} c_i m_i \tag{7}$$

$$v = \sum_{i=1}^{N} c_i m_i^2 - \Big(\sum_{i=1}^{N} c_i m_i\Big)^2. \tag{8}$$

This operation is denoted as:

$$q(w) = \mathrm{MM}\big(f(w)\big). \tag{9}$$

III. Three/Two Gaussian Approximation

In this section we describe an approximation of the product of a single Gaussian and a Gaussian mixture, which is key for understanding the behavior of the three/two Gaussian parametric LDLC decoding algorithm. Analysis is performed by evaluating the Kullback-Leiber divergence.

The single Gaussian $Y(w)$ represents the channel message and has mean m_a and variance v_a. The Gaussian mixture $R(w)$ represents the check-to-variable node messages and is a periodic mixture of Gaussians with period $\frac{1}{|h|}$ and parameters m_c and v_c, and is given by:

$$R(w) = \sum_{i=-\infty}^{\infty} \mathcal{N}(w; m_c + \frac{i}{h}, v_c). \tag{10}$$

And let $\widetilde{R}(w)$ be the summation in (10) restricted to some finite integer set $\mathcal{B}$.

We want to approximate an infinite Gaussian mixture $Y(w)R(w)$ with $Y(w)\widetilde{R}(w)$, which consists of a finite number of Gaussians. In Fig. 1-(a), $Y(w)$ and $R(w)$ are illustrated. In Fig. 1-(b) the true product $Y(w)R(w)$ and the single-Gaussian moment matching (MM) approximation of $Y(w)R(w)$ is shown. The true product and the MM approximation are visually similar, but a difference exists in the tails, which results in a poor approximation. This poor approximation in the Gaussian messages causes errors to accumulate as the LDLC iterative decoding progresses.

A. Gaussian Neighbors Selection

Here we consider the $|\mathcal{B}| = 3$ and $|\mathcal{B}| = 2$ Gaussians which are neighbors near m_a. Let the 2-Gaussian set be $\mathcal{B} = \{b_1, b_2\}$, and let the 3-Gaussian set be $\mathcal{B} = \{b_0, b_1, b_2\}$, with $b_0 = b_1 - 1$ and $b_2 = b_1 + 1$.

For the 2-Gaussian set, we select two Gaussians from the periodic mixture, which are close to the mean m_a. Find b_1 such that:

$$\frac{b_1}{h} + m_c < m_a < \frac{b_1 + 1}{h} + m_c, \tag{11}$$

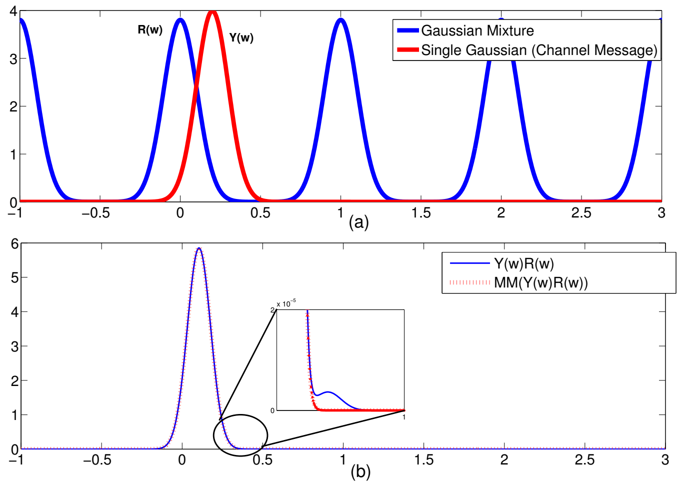

Fig. 1. Multiplication of a Gaussian mixture and a single Gaussian. and the approximation with a Single Gaussian. This operation takes place at the variable node.

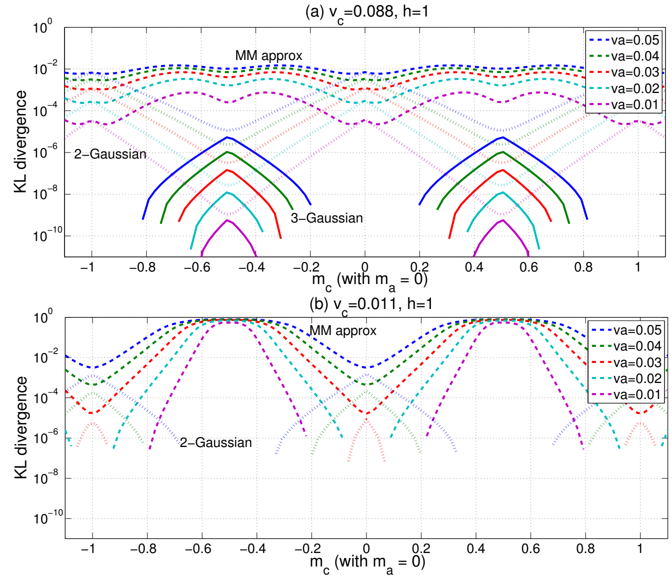

Fig. 2. KL divergence for $MM(Y(w)R(w))$ (dashed-line), 3-Gaussian approximation (solid-line) and 2-Gaussian approximation (dotted-line)

for $h > 0$, which is:

$$b_1 = \lfloor -h(m_c - m_a) \rfloor. \tag{12}$$

And for the 3-Gaussian set we choose the nearest Gaussian, and the two neighbors of nearest Gaussian. That is:

$$b_0 = b_1 - 1, \tag{13}$$
$$b_1 = \lceil h(m_c - m_a) \rfloor, \text{ and} \tag{14}$$
$$b_2 = b_1 + 1, \tag{15}$$

where $\lceil \cdot \rfloor$ denotes rounding to nearest integer.

The resulting mixtures are:

$$\widetilde{R}(w) = \mathcal{N}(w; \frac{b_1}{h} + m_c, v_c) + \mathcal{N}(w; \frac{b_2}{h} + m_c, v_c), \tag{16}$$

for the 2-Gaussian case. And:

$$\widetilde{R}(w) = \mathcal{N}(w; \frac{b_0}{h} + m_c, v_c) + \mathcal{N}(w; \frac{b_1}{h} + m_c, v_c) + \mathcal{N}(w; \frac{b_2}{h} + m_c, v_c). \tag{17}$$

for the 3-Gaussian case. Since $\widetilde{R}(w)$ consists of three or two Gaussian then $Y(w)\widetilde{R}(w)$ also consists of three or two Gaussians.

B. Kullback-Leibler Divergence Analysis

The Kullback-Leibler (KL) divergence is a measurement of the dissimilarity between two probability distributions, and it is equal to zero when the two distributions are the same. The KL divergence between $Y(w)R(w)$ and the approximation $Y(w)\widetilde{R}(w)$ is given by:

$$\int_{-\infty}^{\infty} Y(w)R(w) \log \frac{Y(w)R(w)}{Y(w)\widetilde{R}(w)} dw. \tag{18}$$

While selecting a sufficiently small number of Gaussians we want to minimize the KL divergence. Towards that end, the KL divergence is characterized for these approximations. The KL divergence is a five-parameter function, because the single Gaussian $Y(w)$ is described by m_a and v_a and the Gaussian mixture $R(w)$ by m_c, v_c and h. But the KL divergence (18) depends only on the difference $m_c - m_a$, so set $m_a = 0$ without loss of generality, reducing the number of parameters to four. Deriving analytically the KL divergence is not possible, so we evaluate the KL divergence numerically.

In Fig. 2 the KL divergence for the single-Gaussian moment matching approximation (dashed-line), three-Gaussian approximation (solid-line) and two-Gaussian approximation (dotted-line) are shown, using typically observed values for v_a and v_c under LDLC decoding. We present all values for m_c, but not all are equally likely because m_c is not uniformly distributed. The worse case is when $h = 1$. Fig. 2-(a) shows $v_c = 0.088$, corresponding to an early decoding iteration. Here, even the MM approximation presents a KL divergence of less than 10^{-2}. Empirically we have observed that a KL divergence of greater than 10^{-2} is a poor approximation for the proposed LDLC decoding algorithm. But KL divergence of less than 10^{-3} at least gives visually similar Gaussian functions.

Fig 2-(b) shows $v_c = 0.011$, corresponding to intermediate iterations of LDLC decoding, where the MM presents worse KL divergence. The KL divergence for two-Gaussian approximation is always less than 10^{-2} and the three-Gaussian approximation is even better. This suggests that the two-Gaussian approximation may be sufficient. The simulation results will show that this is often true, but when the dimension is very large, the three-Gaussian approximation is more reliable.

IV. Three/Two Gaussian Parametric Decoding

A parametric LDLC decoding algorithm is presented here. The variable-to-check message along edge k is a single Gaussian denoted $f_k(w)$. The check-to-variable message along edge k is a single Gaussian denoted $\widetilde{p}_k(w)$. Single Gaussians are represented by its mean and variance. Internally at the variable node, messages are represented by mixtures of multiple Gaussians.

A. Three/Two Gaussian Parametric Decoding Description

For the AWGN channel, the received message is

$$y(w) = \mathcal{N}(w; y_k, \sigma^2). \tag{19}$$

Check Node: The incoming messages are d single Gaussians $f_i(w) = \mathcal{N}(w; m_i, v_i)$ for $i = 1, 2 \ldots, d$. The output message $\widetilde{p}_i(w)$ at the convolution step is a single Gaussian with mean $\tilde{m}$ and variance $\tilde{v}$ given by:

$$\widetilde{m}_i = -\frac{1}{h_i} \sum_{j=1}^{d \backslash i} h_j m_j \tag{20}$$

$$\widetilde{v}_i = \frac{1}{h_i^2} \sum_{j=1}^{d \backslash i} h_j^2 v_j \tag{21}$$

The computation of $\widetilde{m}_i$ and $\widetilde{v}_i$ can be performed using a forward-backward recursion.

Variable node: The messages coming from the check nodes are single Gaussian $\mathcal{N}(w; \widetilde{m}_i, \widetilde{v}_i)$. Then the expansion step (periodic with period $1/|h_i|$ if $\mathcal{B} = \mathbb{Z}$) is approximated by:

$$\widetilde{R}_i(w) = \sum_{b \in \mathcal{B}} \mathcal{N}(w; m_i(b), \tilde{v}_i), \tag{22}$$

where the mean of each Gaussian is:

$$m_i(b) = \widetilde{m}_i + \frac{b}{h_i}, \tag{23}$$

for $b \in \mathcal{B}$ and the set $\mathcal{B}$ represents a subset of the integers, e.g. $|\mathcal{B}| = 3$ or 2, as described earlier. The message $f_i(w)$ sent back to the check node is a single Gaussian approximated by:

$$f_i(w) = \mathrm{MM}\Big(y_k(w) \prod_{j=1}^{d \backslash i} \widetilde{R}_j(w) \Big), \tag{24}$$

where $y_k(w) = \mathcal{N}(w; y_k, \sigma^2)$ is the channel message, and $\widetilde{R}_i(w)$ is the approximation of the periodic expansion. To maintain low storage and low complexity, a single Gaussian is used in the messages between the variable and check nodes, and this single Gaussian message is found by moment matching.

B. Forward-backward recursion

The computation of the output at the variable node $f_k(w)$ can be implemented by a forward-backward recursion. This recursion is distinct from previously described forward-backward approaches in how the channel value $Y(w)$ is handled [10]— in the three/two Gaussian parametric decoding algorithm the channel message is multiplied last.

The forward recursion is defined as:

$$\alpha_k(w) = \alpha_{k-1}(w) \cdot \widetilde{R}_k(w) \tag{25}$$

for $k = 2, 3, \ldots, d$, with $\alpha_1(w)$ initialized as equal to $\widetilde{R}_1(w)$. The backward recursion $\beta_k(w)$ is similarly computed for $k = d-1, d-2, \ldots, 1$ with β_d initialized as equal to $\widetilde{R}_d(w)$. Then, combine the forward and backward recursion to obtain:

$$\widetilde{f}_k(w) = \alpha_{k-1}(w) \cdot \beta_k(w). \tag{26}$$

Finally the single Gaussian output at the variable node is calculated using the moment matching approximation:

$$f_k(w) = \mathrm{MM}\Big(y_k(w) \cdot \widetilde{f}_k(w) \Big). \tag{27}$$

C. Three/Two Gaussian Parametric Decoding Complexity

In this section a description of the complexity is given. The complexity of the three/two Gaussian parametric decoding algorithm is dominated by the forward and backward algorithm which is $\mathcal{O}(n \cdot t \cdot 3^{d-1})$ and $\mathcal{O}(n \cdot t \cdot 2^{d-1})$ for 3-Gaussian approximation and 2-Gaussian approximation respectively, where t is the number of iterations, n is the lattice dimension and d is the degree of the LDLC inverse generator matrix. The storage required is $4 \cdot n \cdot d$, means and variances needed for the nd variable messages and nd check messages. Internally at the variable node, the storage is temporary and does not depend on n.

The complexity of the quantized BP decoding algorithm [6] is $\mathcal{O}(n \cdot t \cdot d \cdot \frac{L}{\Delta})$ where Δ is the probability density function resolution and L is the range length, and is dominated by the discrete Fourier transform operations. The complexity for [10] is $\mathcal{O}(n \cdot d \cdot t \cdot K^2 \cdot M^4)$, dominated by the moment matching algorithm, and for [12] is $\mathcal{O}(n \cdot d \cdot t \cdot K \cdot M^3)$, dominated by sorting and searching in tables, where K is the number of replications and M the number of Gaussian used in the mixtures.

D. Three/Two Gaussian Parametric Decoding Algorithm

In this section the three/two Gaussian parametric decoding algorithm is summarized.

$Input$: The received message $\mathbf{y} = \mathbf{Gb} + \mathbf{z}$, the channel variance σ^2, the inverse generator $\mathbf{H}$ and the maximum number of iterations $iter\_max$.

$Output$: The estimated information $\hat{b}$.

1) Variable node k, for $k = 1, 2, \ldots, n$, sends to all connected check nodes the message y_k and σ^2 from the channel.
2) At the check node every message $\widetilde{p}_i(w)$, a single Gaussian, for $i = 1, 2, \ldots, d$, to be sent to the variable node is computed. The mean is computed as in equation (20) and the variance is given in equation (21).
3) At the variable node k, the ith message, for $i = 1, 2, \ldots, d$, to be sent to the check node is calculated. The expansion step is calculated by selecting three or two Gaussians in the mixture as described in Sec. III-A.
4) The selected mixtures are multiplied, except the message i, to calculate

$$f_i(w) = \mathrm{MM}\Big(y_k(w) \prod_{j=1}^{d \backslash i} \widetilde{R}_j(w) \Big). \tag{28}$$

5) Steps 2-4 are repeated until the maximum number of iteration $iter\_max$ is reached.

6) The final estimation is made by combining all message in variable node, where $\hat{x}_k$ is the mean of:

$$\text{MM}\Big(y_k(w)\prod_{j=1}^{d}\tilde{R}_j(w)\Big). \quad (29)$$

7) Finally the received message is estimated by

$$\hat{\mathbf{b}} = \lceil H\hat{\mathbf{x}} \rfloor \quad (30)$$

V. Numerical Results

The all-zeros codeword was simulated over the AWGN channel. The inverse generator matrix was generated as in [6], with the generator sequence $\mathbf{h} = \{1, \frac{1}{\sqrt{d}}, \ldots, \frac{1}{\sqrt{d}}\}$, where $\alpha = \frac{\sum_{i=2}^{d} h_i^2}{h_1^2} < 1$, a necessary condition for exponential convergence of the message variance for the BP decoder. The inverse generator was further normalized in order to satisfy $\sqrt[n]{|det(\mathbf{H})|} = 1$.

Different lattice dimensions $n = 100$, $n = 1000$ and $n = 10,000$ were simulated, and the inverse generator H has degree $d = 3$ for dimension $n = 100$, and $d = 7$ for dimension $n = 1000$ and $n = 10000$. The symbol error rate (SER) versus the gap from the unconstrained capacity was evaluated. The unconstrained AWGN channel capacity corresponds to $\sigma^2 = \frac{1}{2\pi e}$.

The convergence in message variance is dominated by the edge with the generator sequence h with the greatest absolute value (e.g $h \approx \pm 1$), called the "dominant message". Three cases were simulated, (a) three Gaussian only, (b) two Gaussian only and (c) when the dominant edge only has three Gaussians and others edge have two Gaussians.

As shown in Fig. 3 the three/two Gaussian parametric decoding algorithm performs nearly as well as the quantized algorithm [6] in the SER case. For dimension $n = 1000$ the approximation using two Gaussians is sufficient for accurate performance. For $n = 10,000$ the approximation using three Gaussians presents a better performance than the approximation using two Gaussians.

We hypothesize that the gap that appears is due to use the single Gaussian message at the variable node output, which is maintained to keep the storage requirements low. This could be improved if the message between check and variable nodes are approximated with a greater number of Gaussians, but a greater memory is needed.

VI. Conclusion

In this work we presented the three/two Gaussian parametric decoding algorithm for low density lattice codes, which is a reliable and efficient decoding algorithm. The three/two Gaussian parametric decoding algorithm maintains a low storage requirement. This is because the messages between variable and check nodes are only single Gaussian functions.

Approximating infinite Gaussian mixtures internally at the variable node with three or two Gaussians results in a better approximation than a single Gaussian approximation, with respect to the Kullback-Leibler divergence. The three/two Gaussian parametric decoding algorithm presents nearly similar

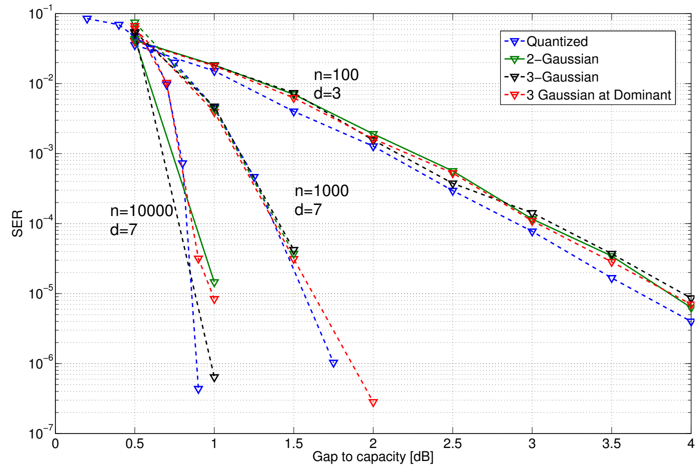

Fig. 3. SER vs the gap from capacity for dimension $n = 100$, $n = 1000$, $n = 10,000$

performance compared to the quantized decoding algorithm. These characteristics of the three/two Gaussian parametric decoder algorithm makes it a suitable candidate for hardware implementation.

References

[1] C. E. Shannon, "Probability of error for optimal codes in a Gaussian channel," *Bell System Technical Journal, The*, vol. 38, pp. 611–656, 1959.

[2] R. de Buda, "The upper error bound of a new near-optimal code," *Info. Theory, IEEE Trans. on*, vol. 21, pp. 441–445, Jul. 1975.

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

[4] T. Linder, C. Schlegel, and K. Zeger, "Corrected proof of de Buda's theorem," *Info. Theory, IEEE Trans. on*, vol. 39, no. 5, pp. 1735–1737, Sep 1993.

[5] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York, NY, USA: Springer-Verlag, 1999, iSBN 0-387-98585-9.

[6] N. Sommer, M. Feder, and O. Shalvi, "Low density lattice codes," in *Info. Theory, 2006 IEEE International Symposium on*. Seattle, WA, USA: IEEE, Jul. 2006.

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

[8] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference with structured codes," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*, July 2008, pp. 772–776.

[9] B. Chen, D. Jayakody, and M. Flanagan, "Cooperative relaying with low-density lattice coding and joint iterative decoding," in *Turbo Codes and Iterative Information Processing (ISTC), 2014 8th International Symposium on*, Aug 2014, pp. 254–258.

[10] B. Kurkoski and J. Dauwels, "Message-passing decoding of lattices using Gaussian mixtures," in *Info.Theory, 2008. ISIT 2008. IEEE International Symposium on*, July 2008, pp. 2489–2493.

[11] ——, "Reduced-memory decoding of low-density lattice codes," *Communications Letters, IEEE*, vol. 14, no. 7, pp. 659–661, July 2010.

[12] Y. Yona and M. Feder, "Efficient parametric decoder of low density lattice codes," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. Seoul, Korea: IEEE, Jun.-Jul. 2009, pp. 744–748.

[13] B. Kurkoski, K. Yamaguchi, and K. Kobayashi, "Single-Gaussian messages and noise thresholds for decoding low-density lattice codes," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*. Seoul, Korea: IEEE, Jun.–Jul. 2009, pp. 734–738.