Information Geometry of Turbo and Low-Density Parity-Check Codes

Shiro Ikeda. *Member, IEEE,* Toshiyuki Tanaka. *Member, IEEE,* Shun-ichi Amari. *Fellow, IEEE,*

***Abstract*— Since the proposal of turbo codes in 1993, many studies have appeared on this simple and new type of codes which give a powerful and practical performance of error correction. Although experimental results strongly support the efficacy of turbo codes, further theoretical analysis is necessary, which is not straightforward. It is pointed out that the iterative decoding algorithm of turbo codes shares essentially similar ideas with low-density parity-check (LDPC) codes, with Pearl's belief propagation algorithm applied to a cyclic belief diagram, and with the Bethe approximation in statistical physics. Therefore the analysis of the turbo decoding algorithm will reveal the mystery of those similar iterative methods. In this paper, we recapture and extend the geometrical framework initiated by Richardson to the information geometrical framework of dual affine connections, focusing on both of the turbo and LDPC decoding algorithms. The framework helps our intuitive understanding of the algorithms and opens a new prospect of further analysis. We reveal some properties of these codes in the proposed framework, including the stability and error analysis. Based on the error analysis, we finally propose a correction term for improving the approximation.**

***Index Terms*— belief propagation, information geometry, low-density parity-check (LDPC) codes, perturbation analysis, turbo codes.**

I. Introduction

THE properties of turbo codes have been extensively studied since it was introduced in 1993 [1], [2]. Although the encoding process and the iterative decoding algorithm are simple, theoretical analysis is not straightforward, and the main results so far obtained are mostly empirical. In addition to the experimental studies, clues have been sought in other methods. Since there are some iterative methods which are closely related to turbo codes, theoretical analysis of those methods were expected to give further understanding. One of them is another class of error correcting codes, low-density parity-check (LDPC) codes, which was originally proposed by Gallager [3], [4] and was rediscovered by MacKay [5]. Other methods have been found even in different fields, such as artificial intelligence and statistical physics. McEliece et al. showed that the turbo decoding algorithm is equivalent to Pearl's belief propagation algorithm [6], applied to a belief diagram with loops [7], and MacKay demonstrated that the LDPC decoding algorithm (the sum-product algorithm) is also equivalent to the belief propagation algorithm [5], while Kabashima and Saad pointed out that the iterative process of the Bethe approximation in statistical physics is the same as that of the belief propagation algorithm [8]–[10] (see also Yedidia et al. [11].). Although these results have shown that the turbo decoding algorithm shares the same idea with these methods, the efficacies of them are not fully understood theoretically, either.

Recently, some pathways for theoretical analysis of the decoding algorithms have been shown. One is the geometrical framework of the turbo decoding algorithm initiated by Richardson [12]. The existence of fixed points, a condition of the fixed point to be unique, and its local stability are studied in this framework. Another pathway is the density evolution [13] applied to the LDPC decoding algorithm. The density evolution describes the time evolution of message distribution. The prospects of these studies are promising, and further studies along these approaches are necessary.

In this article, we propose not only a new interpretation of the geometrical framework, but also an extension of it, with the help of information geometry [14], [15]. Information geometry studies intrinsic geometrical structures existing in families of probability distributions by using the two dual criteria of geometrical flatness (exponential or e–flatness and mixture or m–flatness) coupled with the Fisher information metric. We build a unified information geometrical framework to analyze the decoding algorithms of turbo and LDPC codes, which helps our intuitive understanding. The framework is general so that main results are applicable to related iterative algorithms.

The ideal goal of turbo and LDPC decodings is the maximization of the posterior marginals (MPM), which achieves the minimum bit error rate. However, since the exact MPM decoding is computationally intractable, it is approximated with iterative methods. The unified geometrical structure of the algorithms is elucidated by means of the e– and m–projections in information geometry together with the generalized Pythagorean theorem. Here, the Kullback-Leibler divergence, the Fisher information, and the skewness tensor play fundamental roles. The equilibrium of the iterative al-

Manuscript received February 20, 2003; revised November 20, 2003. This work was supported in part by the Grant-in-Aid for Scientific Research (14084208 and 14084209), MEXT, Japan. The material in this paper was presented in part at the Fifteenth Annual Conference on Neural Information Processing Systems, Vancouver, Canada, December 2001; the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002.

S. Ikeda is with the Department of Statistical Methodology, Institute of Statistical Mathematics, Tokyo, 106-8569, Japan. He is now also a visiting academic at the Gatsby Computational Neuroscience Unit, University College London, WC1N 3AR, United Kingdom, under the fellowship between the Royal Society and the Japan Society for the Promotion of Science (e-mail: shiro@ism.ac.jp).

T. Tanaka is with the Department of Electronics and Information Engineering, Tokyo Metropolitan University, Tokyo, 192-0397, Japan (e-mail: tanaka@eei.metro-u.ac.jp).

S. Amari is with the RIKEN Brain Science Institute, Saitama, 351-0198, Japan (e-mail: amari@brain.riken.jp).

gorithms is analyzed and its local stability condition is given in information geometrical terms. These are not only a new formulation and elucidation of Richardson's framework from a more general standpoint, but open a prospect to integrate wide varieties of iterative inference methods extensively studied in these years in the areas of information theory, statistical physics, statistical inference, neural networks, and artificial intelligence.

We further analyze the accuracy of soft decoding results of the iterative decoding algorithms in terms of the e– and m–curvatures. Hard decoding results are of primary interest in many studies, but for some applications, such as multiple user applications [16]–[19], the accuracy of soft decoding results is also important. In this paper, the error will be given by asymptotic expansion, so that the terms can be used to improve the results. We give an explicit algorithm for the improvement. The error analysis also gives insights into a design principle of LDPC codes, and shows why LDPC codes work so well. We finally touch upon the "free energy" in the statistical physics approach [10], [11].

The outline of the paper is as follows. In section II, we give the original schemes of turbo and LDPC codes. The basic strategy of the MPM decoding is given in section III. Section IV introduces the information geometry. Sections V and VI describe the information geometry of turbo and LDPC decodings, respectively. Decoding errors are analyzed in section VII, and finally conclusion is given with some discussions for future perspectives in section IX.

II. Original Definitions of Turbo and LDPC Codes

A. Turbo Codes

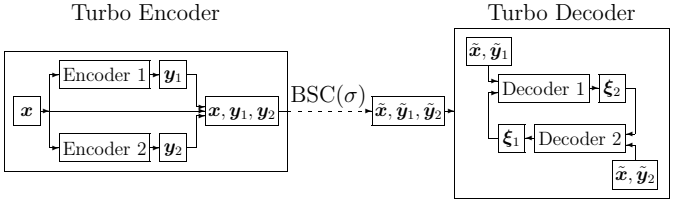

Fig. 1. Structure of turbo codes.

1) Encoding: The idea of turbo codes is illustrated in Fig.1. Let $\boldsymbol{x} = (x_1, \cdots, x_N)^T, x_i \in \{-1, +1\}$ be the information bits to be transmitted. We assume a binary symmetric channel (BSC) with bit-error rate σ, and it is easy to generalize the results to any memoryless channel (see Appendix I). Turbo codes use two encoders, Encoders 1 and 2 in the figure, which generate two sets of parity bits in the encoding process. We denote them by $\boldsymbol{y}_1 = (y_{11}, \cdots, y_{1L})^T$ and $\boldsymbol{y}_2 = (y_{21}, \cdots, y_{2L})^T, y_{1j}, y_{2j} \in \{-1, +1\}$. Each set of parity bits $\boldsymbol{y}_r$, $r = 1, 2$, is a function of $\boldsymbol{x}$ and is represented as $\boldsymbol{y}_r(\boldsymbol{x})$ when an explicit expression is necessary. The set of these codes $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2)$ are transmitted through the BSC, and a receiver observes their noisy version, $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$, $\tilde{x}_i, \tilde{y}_{1j}, \tilde{y}_{2j} \in \{-1, +1\}$.

2) Decoding: Turbo codes handle the case where the direct decoding with $(\tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ as a single set of parity bits is intractable, while the soft decoding with each of $\tilde{\boldsymbol{y}}_1$, $\tilde{\boldsymbol{y}}_2$ is tractable. Two decoders are used for the decoding, Decoders 1 and 2 in the figure. Decoder 1 infers the original information bits, $\boldsymbol{x}$, from $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1)$, and Decoder 2 does the same from $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_2)$. The inferences of these two decoders may differ initially, and a better inference is searched for through iterative information exchanges.

Let us define the following variables corresponding to the marginal log-likelihood ratios (see for example [12], [20]) with the use of the conditional probabilities $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ and $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, $r = 1, 2$,

$$lx_i = \frac{1}{2} \ln \frac{\sum_{\{\boldsymbol{x}: x_i = +1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}: x_i = -1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x})} = \frac{1}{2} \ln \frac{p(\tilde{x}_i|x_i = +1)}{p(\tilde{x}_i|x_i = -1)},$$

$$ly_{rj} = \frac{1}{2} \ln \frac{\sum_{\{\boldsymbol{x}: y_{rj} = +1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}: y_{rj} = -1\}} p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})} = \frac{1}{2} \ln \frac{p(\tilde{y}_{rj}|y_{rj} = +1)}{p(\tilde{y}_{rj}|y_{rj} = -1)},$$

$$L_r\boldsymbol{x} = F(l\boldsymbol{x}, l\boldsymbol{y}_r) = \frac{1}{2} \ln \frac{\sum_{\{\boldsymbol{x}: x_i = +1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}{\sum_{\{\boldsymbol{x}: x_i = -1\}} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})}. \quad (1)$$

Here, the factor $1/2$ is introduced to have consistency with our framework, and the function $F(l\boldsymbol{x}, l\boldsymbol{y}_r)$ is calculated efficiently by BCJR algorithm[21]. The turbo decoding algorithm makes use of two slack variables, $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2 \in \Re^N$, called the "extrinsic variables," for exchanging information between the decoders. The algorithm is given as follows. Its meaning will be explained later from the geometrical point of view.

Turbo decoding (Original)

1) Set $\boldsymbol{\xi}_1 = \mathbf{o}$ and $t = 1$.
2) Calculate $L_1\boldsymbol{x}^{(t)} = F((l\boldsymbol{x} + \boldsymbol{\xi}_1), l\boldsymbol{y}_1)$ from (1) and update $\boldsymbol{\xi}_2$ as follows.
$$\boldsymbol{\xi}_2 = L_1\boldsymbol{x}^{(t)} - (l\boldsymbol{x} + \boldsymbol{\xi}_1).$$
3) Calculate $L_2\boldsymbol{x}^{(t)} = F((l\boldsymbol{x} + \boldsymbol{\xi}_2), l\boldsymbol{y}_2)$ from (1) and update $\boldsymbol{\xi}_1$ as follows.
$$\boldsymbol{\xi}_1 = L_2\boldsymbol{x}^{(t)} - (l\boldsymbol{x} + \boldsymbol{\xi}_2).$$
4) Iterate 2 and 3 by increasing t by one, until $L_1\boldsymbol{x}^{(t)} = L_2\boldsymbol{x}^{(t)} = L_1\boldsymbol{x}^{(t+1)} = L_2\boldsymbol{x}^{(t+1)}$.

Ideally, steps 2 and step 3 would be iterated until convergence is achieved, but in practice, the number of iterations is fixed at less than 20.

B. LDPC Codes

1) Encoding: Figure 2 illustrates the structure of LDPC codes. Let $\boldsymbol{s} = (s_1, \cdots, s_M)^T$, $s_i \in \{0, 1\}$, be the information bits. Although we use notations different from those of turbo codes, it will soon become clear that the problems are formulated in a unified view, i.e., estimating $\boldsymbol{x}$ from an observed $\tilde{\boldsymbol{y}}$. To compose the generator and parity check matrices, two sparse matrices, $C_1 \in \{0, 1\}^{K \times M}$ and $C_2 \in \{0, 1\}^{K \times K}$ are prepared, where C_2 is invertible in the modulo 2 arithmetic.

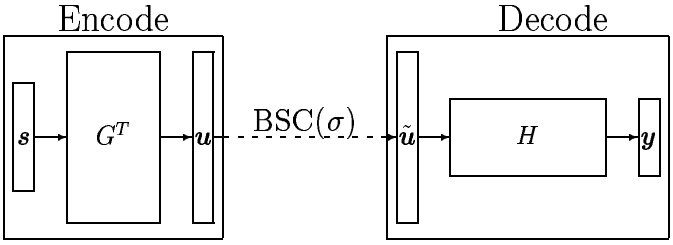

Fig. 2. Structure of LDPC codes.

They are shared by the sender and the receiver. The parity check matrix is

$$H = (C_1\ C_2), \quad H \in \{0,1\}^{K\times N},$$

where $N = M + K$. The generator matrix, $G^T \in \{0,1\}^{N\times M}$, is given by

$$G^T = \begin{pmatrix} E_M \\ C_2^{-1}C_1 \end{pmatrix} \mod 2,$$

where E_M is an identity matrix of size M. The codeword, $\boldsymbol{u} = (u_1, \cdots, u_N)^T$, is generated from $\boldsymbol{s}$:

$$\boldsymbol{u} = G^T\boldsymbol{s} \mod 2.$$

From the definition of G^T, the first M bits of $\boldsymbol{u}$ are identical to $\boldsymbol{s}$, and $\boldsymbol{u}$ is sent through a channel. We also assume a BSC with bit-error rate σ. Codeword $\boldsymbol{u}$ is disturbed and received as $\tilde{\boldsymbol{u}}$. Let $\boldsymbol{x} = (x_1, \cdots, x_N)^T$, $x_i \in \{0,1\}$ be the noise vector, and received codeword $\tilde{\boldsymbol{u}}$ is

$$\tilde{\boldsymbol{u}} = \boldsymbol{u} + \boldsymbol{x} \mod 2.$$

The LDPC decoding estimates noise vector $\boldsymbol{x}$, which yields an estimate of $\boldsymbol{s}$, since $\boldsymbol{s}$ is given by the first M bits of $\tilde{\boldsymbol{u}} + \boldsymbol{x}$ (mod 2). In the decoding process, the parity check matrix $H = \{h_{ij}\} = (C_1\ C_2) \in \{0,1\}^{K\times N}$ is used; it satisfies the equality $HG^T = O$. Syndrome vector $\boldsymbol{y} = (y_1, \cdots, y_K)^T$ is calculated by using $\boldsymbol{y} = H\tilde{\boldsymbol{u}}$. When noise is $\boldsymbol{x}$, the syndrome $\boldsymbol{y}$ is

$$\boldsymbol{y}(\boldsymbol{x}) = H\tilde{\boldsymbol{u}} = H(\boldsymbol{u} + \boldsymbol{x}) = HG^T\boldsymbol{s} + H\boldsymbol{x} = H\boldsymbol{x} \mod 2.$$

When $\tilde{\boldsymbol{y}}$ is the observed syndrome, the decoding problem is to estimate $\boldsymbol{x}$ that satisfies $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$.

2) Decoding: The detailed descriptions of the iterative decoding algorithm for LDPC codes are found elsewhere [3], [5], and we describe it briefly here. The decoding algorithm consists of two steps: the "horizontal step" and the "vertical step," which are iterated alternately. A set of probability distributions is updated in each step, that is, $\{q_{ri}^{(0)}, q_{ri}^{(1)}\}$ and $\{p_{ri}^{(0)}, p_{ri}^{(1)}\}$, respectively, where

$$q_{ri}^{(0)} + q_{ri}^{(1)} = 1, \quad p_{ri}^{(0)} + p_{ri}^{(1)} = 1,$$

for pairs of indices (r,i), $r = 1,\cdots,K$, $i = 1,\cdots,N$, such that $h_{ri} = 1$. The quantity $q_{ri}^{(x)}$ represents a guess of the probability that y_r is observed when $x_i = x$, where the distribution of $\boldsymbol{x}$ other than x_i is assumed to be given by $p_{ri}^{(x)}$. The sum of $q_{ri}^{(0)}$ and $q_{ri}^{(1)}$ is not necessarily 1, but it is normalized for simplicity. The quantity $p_{ri}^{(x)}$ is a guess of the probability of x_i to be x when y_r is observed. The updating rule is described below.

LDPC decoding (Original)

Initialization:

Set $p_{ri}^{(0)} = 1 - \sigma$ and $p_{ri}^{(1)} = \sigma$ for pairs of indices (r,i) such that $h_{ri} = 1$.

Horizontal step:

Update $\{q_{ri}^{(0)}, q_{ri}^{(1)}\}$ as follows. Note that summations and products are taken over pairs (r,i) for which $h_{ri} = 1$.

$$lq_{ri} = \ln \frac{\sum_{\boldsymbol{x}:x_i=1}\{p(\tilde{y}_r|\boldsymbol{x})\prod_{i':i'\neq i,h_{ri'}=1} p_{ri'}^{(x_{i'})}\}}{\sum_{\boldsymbol{x}:x_i=0}\{p(\tilde{y}_r|\boldsymbol{x})\prod_{i':i'\neq i,h_{ri'}=1} p_{ri'}^{(x_{i'})}\}},$$

$$q_{ri}^{(0)} = \frac{1}{e^{lq_{ri}}+1}, \quad q_{ri}^{(1)} = \frac{e^{lq_{ri}}}{e^{lq_{ri}}+1}.$$

Vertical step:

Update $\{p_{ri}^{(0)}, p_{ri}^{(1)}\}$ as follows.

$$lp_{ri} = \ln\frac{\sigma}{1-\sigma} + \ln\frac{\prod_{r':r'\neq r,h_{r'i}=1} q_{r'i}^{(1)}}{\prod_{r':r'\neq r,h_{r'i}=1} q_{r'i}^{(0)}},$$

$$p_{ri}^{(0)} = \frac{1}{e^{lp_{ri}}+1}, \quad p_{ri}^{(1)} = \frac{e^{lp_{ri}}}{e^{lp_{ri}}+1}.$$

Convergence:

Stop when the following lp_i, $i = 1,\cdots,N$, converges

$$lp_i = \ln\frac{\sigma}{1-\sigma} + \ln\frac{\prod_{r:h_{ri}=1} q_{ri}^{(1)}}{\prod_{r:h_{ri}=1} q_{ri}^{(0)}}.$$

When the algorithm achieves convergence, the estimate of $\boldsymbol{x}$ is obtained by the hard decision as

$$\hat{x}_i = \begin{cases} 1, & \text{for } lp_i \geq 0 \\ 0, & \text{for } lp_i < 0 \end{cases}, \quad i = 1,\cdots,N.$$

III. Formulation of MPM Decoding

A. Unified View of Turbo and LDPC Decoding

The goal for both of turbo and LDPC decodings is the MPM decoding. We first define the MPM decoding in a unified setting, and its specific form in each of turbo and LDPC decodings is explained in the following subsections. For the rest of the paper, we use the bipolar, i.e., $\{-1,+1\}$, expression for each bit x_i, y_i, $\tilde{x}_i$, and $\tilde{y}_i$ rather than the binary ($\{0,1\}$).

The decoding problem is generally solved based on the posterior distribution of $\boldsymbol{x}$ conditioned on the observed codeword or syndrome vector, i.e., $p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ in turbo codes and $p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$ in LDPC codes. The posterior distribution of $\boldsymbol{x}$ is expressed as

$$q(\boldsymbol{x}) = C\exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})), \qquad (2)$$

where $c_0(\boldsymbol{x})$ consists of the linear terms of $\{x_i\}$; $c_r(\boldsymbol{x})$, $r = 1,\cdots,K$, contain higher order interactions of $\{x_i\}$, and the terms depend on the observed information, $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{y}}$. In the case of turbo codes, $K = 2$, and $c_1(\boldsymbol{x})$ and $c_2(\boldsymbol{x})$ represent interactions in each of the two decoders, while in the case of LDPC codes, $c_r(\boldsymbol{x})$ represents each parity bit. In

the general graphical model, they correspond to cliques. We assume $c_r(\boldsymbol{x}) \neq c_{r'}(\boldsymbol{x})$ for $r \neq r'$. Decoding is to estimate the information bits, $\boldsymbol{x}$, based on $q(\boldsymbol{x})$. One natural approach is the MPM decoding. The MPM estimator minimizes the expected number of wrong bits in the decoded word. The MPM decoding in the bipolar case is achieved by taking the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$. Let $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T$ be the expectation of $\boldsymbol{x}$, and $\hat{\boldsymbol{x}}$ be the decoded MPM estimator. Then

$$\boldsymbol{\eta} = \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}, \quad \hat{\boldsymbol{x}} = \mathrm{sgn}(\boldsymbol{\eta}), \tag{3}$$

where $\mathrm{sgn}(\cdot)$ works in a bitwise manner. The $\boldsymbol{\eta}$ gives the "soft decoding," and the sign of each soft bit η_i gives the "hard decoding," $\hat{x}_i$.

Let $q(x_i)$ be the marginal distribution of one component x_i in $q(\boldsymbol{x})$, and let $\mathit{\Pi}$ denote the operator of the marginalization that maps $q(\boldsymbol{x})$ to a factorizable distribution having the same marginal distributions:

$$\mathit{\Pi} \circ q(\boldsymbol{x}) = \prod_{i=1}^{N} q(x_i).$$

The soft bit η_i depends only on the marginal distribution $q(x_i)$. Since $q(x_i)$ is a Bernoulli distribution, η_i has a one-to-one correspondence to $q(x_i)$. Therefore, the soft decoding is equivalent to the marginalization of $q(\boldsymbol{x})$. The marginalization of $q(\boldsymbol{x})$ generally needs summation over all possible $\boldsymbol{x}$ but one x_i, and it is computationally not tractable in the case of turbo and LDPC codes, where the length of $\boldsymbol{x}$ is more than a few hundred. Instead of marginalizing the entire $q(\boldsymbol{x})$ in (2), we make use of simple submodels, $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, $r = 1, \cdots, K$,

$$p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} + c_r(\boldsymbol{x}) - \varphi_r(\boldsymbol{\zeta}_r)), \tag{4}$$

where $\varphi_r(\boldsymbol{\zeta}_r)$ is the normalization factor. Each $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ includes only one nonlinear term $c_r(\boldsymbol{x})$, and the linear part $c_0(\boldsymbol{x})$ of $\boldsymbol{x}$ is adjusted further through $\boldsymbol{\zeta}_r$, which takes the effect of the other $c_{r'}(\boldsymbol{x})$'s, $r' \neq r$ into account by approximating them by the linear term $\boldsymbol{\zeta}_r \cdot \boldsymbol{x}$. We thus have K component decoders, each of which decodes $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r), r = 1, \cdots, K$. The parameter $\boldsymbol{\zeta}_r$ plays the role of a window through which information from the other decoders, $r' \neq r$, is exchanged. The idea is to adjust $\{\boldsymbol{\zeta}_r\}$ through iterative information exchange to approximate the overall $\mathit{\Pi} \circ q(\boldsymbol{x})$ with $\mathit{\Pi} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$. We assume that the marginalization or the soft decoding is tractable for any $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$.

B. Turbo Decoding

In this subsection, the concrete forms of (2) and (4) for turbo codes are derived. In turbo codes, the receiver observes a noisy version of $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2)$ as $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. We can easily derive the following relation from the assumption of a memoryless channel,

$$p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2 | \boldsymbol{x}) = p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}).$$

The Bayes posterior distribution $p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$ is defined with a prior distribution $\omega_0(\boldsymbol{x})$ of $\boldsymbol{x}$. In this paper, we consider the uniform prior, where $\omega_0(\boldsymbol{x}) = 1/2^N$, and the Bayes posterior distribution is derived as,

$$\begin{aligned} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) &= \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})\omega_0(\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})\omega_0(\boldsymbol{x})} \\ &= \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})}. \end{aligned} \tag{5}$$

Since we consider BSC, where each bit is flipped independently with probability σ, $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ and $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$ have the form of

$$\begin{aligned} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \exp(\beta\tilde{\boldsymbol{x}} \cdot \boldsymbol{x} - N\psi(\beta)), \quad \psi(\beta) = \ln(e^{-\beta} + e^{\beta}) \\ p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x}) = \exp(\beta\tilde{\boldsymbol{y}}_r \cdot \boldsymbol{y}_r(\boldsymbol{x}) - L\psi(\beta)), \quad r = 1, 2. \end{aligned}$$

Here, β is a positive real number called the inverse temperature in statistical physics and is related to σ by

$$\sigma = \frac{1}{2}(1 - \tanh\beta),$$

where $\beta \to 0$ as $\sigma \to 1/2$, and $\beta \to \infty$ as $\sigma \to 0$. Let us define

$$c_0(\boldsymbol{x}) = \beta\tilde{\boldsymbol{x}} \cdot \boldsymbol{x}, \quad c_r(\boldsymbol{x}) = \beta\tilde{\boldsymbol{y}}_r \cdot \boldsymbol{y}_r(\boldsymbol{x}), \quad r = 1, 2,$$

where $c_0(\boldsymbol{x})$ is linear in $\boldsymbol{x}$, and $\tilde{\boldsymbol{y}}_r \cdot \boldsymbol{y}_r(\boldsymbol{x})$ are polynomials in $\boldsymbol{x}$, representing higher order correlational components of many x_i's. The Bayes posterior distribution (5) is rewritten as

$$\begin{aligned} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) &= C\exp(c_0(\boldsymbol{x}) + \beta\tilde{\boldsymbol{y}}_1 \cdot \boldsymbol{y}_1(\boldsymbol{x}) + \beta\tilde{\boldsymbol{y}}_2 \cdot \boldsymbol{y}_2(\boldsymbol{x})) \\ &= C\exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})), \\ C &= \frac{1}{\sum_{\boldsymbol{x}} \exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x}))}, \end{aligned}$$

where C is the normalization factor. This distribution corresponds to $q(\boldsymbol{x})$ in (2), where $K = 2$.

In the turbo decoding algorithm, each of the two constituent decoders marginalizes its own posterior distribution of $\boldsymbol{x}$ derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x}) = p(\tilde{\boldsymbol{x}}|\boldsymbol{x})p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, where a prior distribution of the form

$$\begin{aligned} \omega(\boldsymbol{x}; \boldsymbol{\zeta}_r) = \exp(\boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \psi(\boldsymbol{\zeta}_r)), \\ \boldsymbol{\zeta}_r \in \Re^N, \quad \psi(\boldsymbol{\zeta}_r) = \sum_{i=1}^{N} \ln\bigl(e^{-\zeta_r^i} + e^{\zeta_r^i}\bigr), \end{aligned}$$

is used for taking information from the other decoder. The vectors $\boldsymbol{\zeta}_r$, $r = 1, 2$ correspond to the extrinsic variables in the original turbo decoding algorithm, that is $\boldsymbol{\zeta}_1 = \boldsymbol{\xi}_2$ and $\boldsymbol{\zeta}_2 = \boldsymbol{\xi}_1$. The prior distribution $\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ is a factorizable distribution in which the guess of the other decoder is represented. The posterior distribution of the decoder r is defined as

$$\begin{aligned} p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) = p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r; \boldsymbol{\zeta}_r) &= \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})\omega(\boldsymbol{x}; \boldsymbol{\zeta}_r)} \\ &= \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r)), \\ \varphi_r(\boldsymbol{\zeta}_r) &= \ln \sum_{\boldsymbol{x}} \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x}), \quad r = 1, 2. \end{aligned}$$

Here, $\varphi_r(\boldsymbol{\zeta}_r)$ is the normalization factor which is a function of $\boldsymbol{\zeta}_r$. It is clear that $\boldsymbol{\zeta}_r$ plays the role of the window of information exchange, and that the information is used as a prior. This distribution is of the form of (4).

C. LDPC Decoding

We reformulate the LDPC decoding problem in this subsection. The vectors $\boldsymbol{s}$, $\boldsymbol{u}$, $\tilde{\boldsymbol{u}}$, $\tilde{\boldsymbol{y}}$, and $\boldsymbol{x}$ are treated in the bipolar form, while G^T and H are still in the binary, i.e., $\{0,1\}$, form. Note that 0 in the binary form corresponds to $+1$ in the bipolar form, and vice versa. Each bit y_r of a syndrome vector $\boldsymbol{y}(\boldsymbol{x})$ is written as a higher order correlational product of $\{x_i\}$ in the bipolar form, that is, as a monomial in $\boldsymbol{x}$:

$$y_r(\boldsymbol{x}) = \prod_{j\in\mathcal{L}_r} x_j, \quad \mathcal{L}_r = \{j \mid h_{jr} = 1\},$$

where h_{jr} are elements of the parity-check matrix H.

We now consider the "softened" probability distribution of $\tilde{\boldsymbol{y}}$ conditioned on $\boldsymbol{x}$:

$$\begin{aligned} p(\tilde{\boldsymbol{y}}|\boldsymbol{x}) &= \exp(\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x}) - K\psi(\rho)) \\ &= \exp(c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x}) - K\psi(\rho)), \qquad (6) \\ c_r(\boldsymbol{x}) &= \rho\tilde{y}_r y_r(\boldsymbol{x}), \quad \rho\in\Re, \quad \rho > 0. \end{aligned}$$

In this article, we discuss the "soft constraint" which infers $\boldsymbol{x}$ based on the probability distribution $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ in (6) where a positive real number ρ is finite. More precisely, the MPM decoding is carried out by using $p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$ obtained from (6). However, the LDPC decoding algorithm generally uses the "hard constraint" which searches for the $\boldsymbol{x}$ that exactly satisfies the parity check equations:

$$\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x}).$$

As ρ becomes larger, the probability $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ is concentrated on $\boldsymbol{x}$ satisfying $\tilde{\boldsymbol{y}} = \boldsymbol{y}(\boldsymbol{x})$, and the "soft constraint" approaches the "hard constraint". See Appendix II where how hard decoding results depend on ρ is analyzed. Empirical studies have shown that the "soft constraint" with a fixed ρ has a sufficiently good performance [5]. The reason we introduce a finite ρ is to keep $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ strictly positive for any $\boldsymbol{x}\in\{-1,+1\}^N$. This is necessary to build a common information geometrical framework for turbo and LDPC decodings (see section IV-A).

Note that noise $\boldsymbol{x}$ is bitwise independent, and that its error rate is given by $\sigma = (1/2)(1-\tanh\beta)$. Consequently, we have the prior distribution $\omega_0(\boldsymbol{x})$:

$$\begin{aligned} \omega_0(\boldsymbol{x}) &= \exp(\beta\mathbf{1}_N\cdot\boldsymbol{x} - N\psi(\beta)) \\ &= \exp(c_0(\boldsymbol{x}) - N\psi(\beta)) \qquad (7) \\ c_0(\boldsymbol{x}) &= \beta\mathbf{1}_N\cdot x_i, \quad \mathbf{1}_N = (\underbrace{1,\cdots,1}_{N})^T. \end{aligned}$$

As a result, the Bayes posterior distribution becomes

$$\begin{aligned} p(\boldsymbol{x}|\tilde{\boldsymbol{y}}) &= \frac{p(\tilde{\boldsymbol{y}}|\boldsymbol{x})\omega_0(\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{y}}|\boldsymbol{x})\omega_0(\boldsymbol{x})} \\ &= C\exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + \cdots + c_K(\boldsymbol{x})). \end{aligned}$$

This is equivalent to $q(\boldsymbol{x})$ in (2).

In the horizontal and vertical steps of the LDPC decoding algorithm, marginalization is carried out based on distribution $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, which is calculated from $p(\tilde{y}_r|\boldsymbol{x})$ and prior $\omega(\boldsymbol{x};\boldsymbol{\zeta}_r)$. The parameter specifying the prior $\boldsymbol{\zeta}_r$ is obtained through the window for taking information from the other decoders r. We have

$$\begin{aligned} p(\tilde{y}_r|\boldsymbol{x}) &= \exp(c_r(\boldsymbol{x}) - \psi(\beta)), \\ \omega(\boldsymbol{x};\boldsymbol{\zeta}_r) &= \exp((\beta\mathbf{1}_N + \boldsymbol{\zeta}_r)\cdot\boldsymbol{x} - \psi(\beta\mathbf{1}_N + \boldsymbol{\zeta}_r)), \\ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) &= p(\boldsymbol{x}|\tilde{y}_r;\boldsymbol{\zeta}_r) = \frac{p(\tilde{y}_r|\boldsymbol{x})\omega(\boldsymbol{x};\boldsymbol{\zeta}_r)}{\sum_{\boldsymbol{x}} p(\tilde{y}_r|\boldsymbol{x})\omega(\boldsymbol{x};\boldsymbol{\zeta}_r)} \\ &= \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r\cdot\boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r)), \\ \varphi_r(\boldsymbol{\zeta}_r) &= \ln\sum_{\boldsymbol{x}}\exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r\cdot\boldsymbol{x}), \\ &\boldsymbol{\zeta}_r\in\Re^N, \quad r = 1,2,\cdots,K. \end{aligned}$$

This coincides with the formulation in (4). The above argument shows that the LDPC decoding problem falls into the general framework given in section III-A.

IV. Information Geometry of Probability Distributions

The preliminaries from information geometry [14], [15] are given in this section.

A. Manifolds of Probability Distributions: e–flat and m–flat Submanifolds

Consider the family of all the probability distributions over $\boldsymbol{x}$. We denote it by S:

$$S = \Big\{p(\boldsymbol{x}) \;\Big|\; p(\boldsymbol{x}) > 0, \boldsymbol{x}\in\{-1,+1\}^N, \sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1\Big\}.$$

This is the set of all the distributions over 2^N atoms $\boldsymbol{x}$. The family S has (2^N-1) degrees of freedom and is a (2^N-1)–dimensional manifold belonging to the exponential family [15], [22].

In order to prove this, we introduce random variables

$$\delta_{i_1\cdots i_N}(\boldsymbol{x}) = \begin{cases} 1, & \text{when } \boldsymbol{x} = (i_1,\cdots,i_N)^T \\ 0, & \text{otherwise} \end{cases},$$
$$\text{where } i_k\in\{-1,+1\},\ k = 1,\cdots,N.$$

Any $p(\boldsymbol{x})\in S$ is expanded in the following form:

$$p(\boldsymbol{x}) = \sum_{i_1\cdots i_N} p_{i_1\cdots i_N}\delta_{i_1\cdots i_N}(\boldsymbol{x}), \qquad (8)$$

where $p_{i_1\cdots i_N} = \Pr(x_1 = i_1,\cdots,x_N = i_N)$, which shows $p(\boldsymbol{x})\in S$ is parameterized by 2^N variables $\{p_{i_1\cdots i_N}\}$. Since $\sum_{\boldsymbol{x}} p(\boldsymbol{x}) = 1$, the family S has (2^N-1) degrees of freedom.

Similarly, $\ln p(\boldsymbol{x})$ is expanded:

$$\ln p(\boldsymbol{x}) = \sum_{i_1\cdots i_N} (\ln p_{i_1\cdots i_N})\delta_{i_1\cdots i_N}(\boldsymbol{x}).$$

Since the degrees of freedom are (2^N-1), we set $\boldsymbol{\theta} = \{\theta_{i_1\cdots i_N} \mid (i_1\cdots i_N)\neq(-1\cdots-1)\}$,

$$\theta_{i_1\cdots i_N} = \ln\frac{p_{i_1\cdots i_N}}{p_{-1\cdots-1}}$$

and rewrite (8) as

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \exp\Big(\sum_{i_1\cdots i_N}\theta_{i_1\cdots i_N}\delta_{i_1\cdots i_N}(\boldsymbol{x}) - \varphi(\boldsymbol{\theta})\Big),$$

where

$$\varphi(\boldsymbol{\theta}) = -\ln \Pr(x_1 = \cdots = x_N = -1).$$

This shows S is an exponential family whose natural, or canonical, coordinate system is $\boldsymbol{\theta}$.

The expectations of random variables $\delta_{i_1 \cdots i_N}(\boldsymbol{x})$ are

$$\eta_{i_1 \cdots i_N} = E_p[\delta_{i_1 \cdots i_N}(\boldsymbol{x})] = p_{i_1 \cdots i_N}.$$

They form another coordinate system of S that specifies $p(\boldsymbol{x})$,

$$\boldsymbol{\eta} = \{\eta_{i_1 \cdots i_N} \mid (i_1 \cdots i_N) \neq (-1 \cdots -1)\}.$$

Since S is an exponential family, it naturally has two affine structures: the exponential– or e–affine structure and the mixture– or m–affine structure. These structures were also adopted implicitly by Richardson [12] without resorting to the Riemannian structure and duality, stated in the following. When manifold S is regarded as an affine space in $\ln p(\boldsymbol{x})$, it is e–affine, and $\boldsymbol{\theta}$ gives the e–affine coordinate system. Similarly, when manifold S is regarded as an affine space in $p(\boldsymbol{x})$, it is m–affine, and $\boldsymbol{\eta}$ gives the m–affine coordinate system. They are dually coupled with respect to the Riemannian structure given by the Fisher information matrix, which will be introduced below.

First we define the e–flat and m–flat submanifolds of S.

e–flat submanifold:

Submanifold $M \subset S$ is said to be e–flat, when the following $r(\boldsymbol{x};t)$ belongs to M for all $t \in [0,1]$, $q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$.

$$\ln r(\boldsymbol{x};t) = (1-t)\ln q(\boldsymbol{x}) + t\ \ln p(\boldsymbol{x}) + c(t),$$

where $c(t)$ is the normalization factor. Obviously, $\{r(\boldsymbol{x};t) \mid t \in [0,1]\}$ is an exponential family connecting two distributions, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. In particular, when an e–flat submanifold is a one–dimensional curve, it is called an e–geodesic. The above $\{r(\boldsymbol{x};t) \mid t \in [0,1]\}$ is the e–geodesic connecting $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$. In terms of the e–affine coordinates, $\boldsymbol{\theta}$, a submanifold M is e–flat when it is linear in $\boldsymbol{\theta}$.

m–flat submanifold:

Submanifold $M \subset S$ is said to be m–flat when the following mixture $r(\boldsymbol{x};t)$ belongs to M for all $t \in [0,1], q(\boldsymbol{x}), p(\boldsymbol{x}) \in M$.

$$r(\boldsymbol{x};t) = (1-t)q(\boldsymbol{x}) + tp(\boldsymbol{x}).$$

When an m–flat submanifold is a one–dimensional curve, it is called an m–geodesic. Hence, the above mixture family is the m–geodesic connecting them. In terms of the m–affine coordinates, $\boldsymbol{\eta}$, a submanifold M is m–flat when it is linear in $\boldsymbol{\eta}$.

B. Kullback-Leibler divergence, Fisher Metric, and Generalized Pythagorean Theorem

Manifold S has a Riemannian metric given by the Fisher information matrix I. We begin with the Kullback-Leibler (KL) divergence, $D[\cdot;\cdot]$, defined by

$$D[q(\boldsymbol{x}); p(\boldsymbol{x})] = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) \ln \frac{q(\boldsymbol{x})}{p(\boldsymbol{x})}.$$

The KL–divergence satisfies $D[q;p] \geq 0$, and $D[q;p] = 0$ when and only when $q(\boldsymbol{x}) = p(\boldsymbol{x})$ holds for every $\boldsymbol{x}$. Although symmetry $D[q;p] = D[p;q]$ does not hold generally, it is regarded as an asymmetric squared distance.

Consider two nearby distributions $p(\boldsymbol{x};\boldsymbol{\theta})$ and $p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})$, specified by coordinates $\boldsymbol{\theta}$ and $\boldsymbol{\theta} + d\boldsymbol{\theta}$ in any coordinate system. From the Taylor expansion, their KL–divergence is given by the quadratic form

$$D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] = \frac{1}{2} d\boldsymbol{\theta}^T I(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

where $I(\boldsymbol{\theta})$ is the Fisher information matrix defined by

$$\begin{aligned} I(\boldsymbol{\theta}) &= \sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\theta}) \partial_{\boldsymbol{\theta}} \ln p(\boldsymbol{x};\boldsymbol{\theta}) (\partial_{\boldsymbol{\theta}} \ln p(\boldsymbol{x};\boldsymbol{\theta}))^T \\ &= -\sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\theta}) \partial_{\boldsymbol{\theta}\boldsymbol{\theta}} \ln p(\boldsymbol{x};\boldsymbol{\theta}), \end{aligned}$$

where $\partial_{\boldsymbol{\theta}}$ represents the gradient operator (differentiation with respect to the components of $\boldsymbol{\theta}$). When the squared distance of a small line element $d\boldsymbol{\theta}$ starting from $\boldsymbol{\theta}$ is given by the quadratic form

$$ds^2 = d\boldsymbol{\theta}^T G(\boldsymbol{\theta}) d\boldsymbol{\theta},$$

the space is called a Riemannian manifold with the Riemannian metric tensor $G(\boldsymbol{\theta})$, which is a positive-definite matrix depending on $\boldsymbol{\theta}$. In the present case, the Fisher information matrix $I(\boldsymbol{\theta})$ plays the role of the Riemannian metric $G(\boldsymbol{\theta})$. Hence, the infinitesimal KL–divergence is regarded as a half the squared Riemannian distance.

The Riemannian metric, giving a definition of the inner product to the tangent spaces, also defines the orthogonality of two intersecting curves. Let $p(\boldsymbol{x};\boldsymbol{\theta}_1(t))$ and $p(\boldsymbol{x};\boldsymbol{\theta}_2(t))$ be two curves intersecting at $t = 0$, that is, $\boldsymbol{\theta}_1(0) = \boldsymbol{\theta}_2(0)$. The tangent vectors of the curves at $t = 0$ are represented by $\dot{\boldsymbol{\theta}}_1(t)$ and $\dot{\boldsymbol{\theta}}_2(t)$ by using the coordinates, where $\dot{\boldsymbol{\theta}}_i(t) = d\boldsymbol{\theta}_i(t)/dt$. The two curves are said to be orthogonal at their intersection $t = 0$, when their inner product with respect to the Riemannian metric vanishes,

$$\langle \dot{\boldsymbol{\theta}}_1(0), \dot{\boldsymbol{\theta}}_2(0) \rangle = \dot{\boldsymbol{\theta}}_1(0)^T I(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}_2(0) = 0.$$

Now we state the generalized Pythagoras theorem and the projection theorem, which hold in a general dually flat manifold [15], and show the dual nature of the e– and m–structures with the Riemannian metric.

Theorem 1: Let $p(\boldsymbol{x})$, $q(\boldsymbol{x})$, and $r(\boldsymbol{x})$ be three distributions in S. When the m–geodesic connecting $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ is orthogonal at $q(\boldsymbol{x})$ to the e–geodesic connecting $q(\boldsymbol{x})$ and $r(\boldsymbol{x})$, the following relation holds

$$D[p(\boldsymbol{x}); r(\boldsymbol{x})] = D[p(\boldsymbol{x}); q(\boldsymbol{x})] + D[q(\boldsymbol{x}); r(\boldsymbol{x})].$$

Next we define the m–projection. The e–projection is also defined in a dual manner, by replacing $D[q(\boldsymbol{x}); p(\boldsymbol{x})]$ with $D[p(\boldsymbol{x}); q(\boldsymbol{x})]$, but we do not state the details here.

Definition 1: Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The point in M that minimizes the KL–divergence from $q(\boldsymbol{x})$ to M is denoted by

$$\Pi_M \circ q(\boldsymbol{x}) = \operatorname*{argmin}_{p(\boldsymbol{x}) \in M} D[q(\boldsymbol{x}); p(\boldsymbol{x})],$$

and is called the m–projection of $q(\boldsymbol{x})$ to M.
Finally, the m–projection theorem follows.

Theorem 2: Let M be an e–flat submanifold in S, and let $q(\boldsymbol{x}) \in S$. The m–projection of $q(\boldsymbol{x})$ to M is unique and given by the point in M such that the m–geodesic connecting $q(\boldsymbol{x})$ and $\mathit{\Pi}_M \circ q$ is orthogonal to M at this point.

C. Legendre Transformation and Local Structure

Let $\boldsymbol{\theta}$ be the e–affine coordinate system of S. Every exponential family has the form

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \exp(c(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi(\boldsymbol{\theta})).$$

The function $\varphi(\boldsymbol{\theta})$ is a convex function which is called the cumulant generating function in statistics, and the free energy in statistical physics. The m–affine coordinate system $\boldsymbol{\eta}$ is given by its gradient,

$$\boldsymbol{\eta} = \partial_{\boldsymbol{\theta}}\varphi(\boldsymbol{\theta}),$$

where $\partial_{\boldsymbol{\theta}} = \partial/\partial\boldsymbol{\theta}$ is the gradient operator. There is a dualistic structure described by the Legendre transformation; the dual potential, $\phi(\boldsymbol{\eta})$ is given by

$$\varphi(\boldsymbol{\theta}) + \phi(\boldsymbol{\eta}) - \boldsymbol{\theta}\cdot\boldsymbol{\eta} = 0$$

and is the negative of the Shannon entropy,

$$\phi(\boldsymbol{\eta}) = \sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\eta}) \ln p(\boldsymbol{x};\boldsymbol{\eta}).$$

The Fisher information matrix is given by the second derivative of φ,

$$I(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\varphi(\boldsymbol{\theta}),$$

which is positive-definite. We have shown that the square of the local distance is given by

$$\begin{aligned} D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] &= D[p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta}); d\boldsymbol{\theta}] \\ &= \frac{1}{2} d\boldsymbol{\theta}^T I(\boldsymbol{\theta}) d\boldsymbol{\theta}. \end{aligned}$$

The third derivative of the potential φ,

$$T = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}\boldsymbol{\theta}}\varphi(\boldsymbol{\theta}),$$

is called the skewness tensor. It is a symmetric tensor of order three, and its components are calculated as

$$T_{ijk} = E_p[(x_i - \eta_i)(x_j - \eta_j)(x_k - \eta_k)],$$

where $E_p[\cdot]$ denotes expectation with respect to $p(\boldsymbol{x})$. The KL–divergence is expanded as

$$D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] = \frac{1}{2} d\boldsymbol{\theta}^T I(\boldsymbol{\theta}) d\boldsymbol{\theta} + \frac{1}{6}(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}),$$

where

$$(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}) = \sum_{i,j,k} d\theta_i d\theta_j d\theta_k T_{ijk}(\boldsymbol{\theta})$$

in the component form. This shows the local asymmetry of the KL–divergence:

$$\begin{aligned} &D[p(\boldsymbol{x};\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta})] - D[p(\boldsymbol{x};\boldsymbol{\theta}+d\boldsymbol{\theta}); p(\boldsymbol{x};\boldsymbol{\theta})] \\ &= \frac{1}{3}(d\boldsymbol{\theta})^3 \circ T(\boldsymbol{\theta}). \end{aligned}$$

The skewness tensor plays a fundamental role in the analysis of decoding error.

D. Important Submanifolds and Marginalization

Now, we consider a submanifold, M_D, in which every joint distribution is decomposed as

$$p(\boldsymbol{x}) = \prod_{i=1}^{N} p(x_i), \quad p(\boldsymbol{x}) \in M_D.$$

All the bits of $\boldsymbol{x}$ are independent for every distribution in M_D. Since each bit takes one of $\{-1, +1\}$, $p(x_i)$ is a Bernoulli distribution, and $p(\boldsymbol{x})$ belongs to an exponential family of the form

$$\begin{aligned} p(\boldsymbol{x};\boldsymbol{\theta}) &= \prod_{i=1}^{N} p(x_i;\theta_i) = \prod_{i=1}^{N} \exp(\theta_i x_i - \varphi(\theta_i)) \\ &= \exp(\boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi(\boldsymbol{\theta})), \\ \varphi(\boldsymbol{\theta}) &= \sum_{i=1}^{N} \varphi(\theta_i) = \ln \sum_{i=1}^{N} (e^{-\theta_i} + e^{\theta_i}), \quad \boldsymbol{\theta} \in \Re^N. \end{aligned} \tag{9}$$

The submanifold M_D is N-dimensional, with its e–affine coordinate system $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_N)^T$, which are the natural or canonical parameters in M_D. The other parameter (m–affine coordinate system) is the expectation parameter $\boldsymbol{\eta} = (\eta_1, \cdots, \eta_N)^T$ defined by

$$\boldsymbol{\eta} = E_p[\boldsymbol{x}] = \sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x}.$$

This is equivalent to the soft decoding in (3). There is a simple one-to-one correspondence between $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$:

$$\partial_{\boldsymbol{\theta}}\varphi(\boldsymbol{\theta}) = \boldsymbol{\eta}, \quad \eta_i = \tanh(\theta_i), \quad \theta_i = \frac{1}{2}\ln\frac{1+\eta_i}{1-\eta_i}, \quad i = 1, \cdots, N.$$

Proposition 1: M_D is an e–flat submanifold of S.

Proof: M_D is a submanifold of S. Let $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \Re^N$ and $p(\boldsymbol{x};\boldsymbol{\theta}), p(\boldsymbol{x};\boldsymbol{\theta}') \in M_D$. For any $\boldsymbol{\theta}, \boldsymbol{\theta}'$,

$$\begin{aligned} \ln r(\boldsymbol{x};t) &= (1-t)\ln p(\boldsymbol{x};\boldsymbol{\theta}) + t \ln p(\boldsymbol{x};\boldsymbol{\theta}') + c(\boldsymbol{\theta},\boldsymbol{\theta}';t) \\ &= ((1-t)\boldsymbol{\theta} + t\boldsymbol{\theta}')\cdot\boldsymbol{x} + c(\boldsymbol{\theta},\boldsymbol{\theta}';t). \end{aligned}$$

Let $\boldsymbol{u}(t) = (1-t)\boldsymbol{\theta} + t\boldsymbol{\theta}'$, and $r(\boldsymbol{x};t) = \exp(\boldsymbol{u}(t)\cdot\boldsymbol{x} - \varphi(\boldsymbol{u}(t)))$ belongs to M_D. ■

We now define a number of e–flat submanifolds which play important roles in the decoding algorithms. The first is the submanifold of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ defined by

$$M_0 = \Big\{ p_0(\boldsymbol{x};\boldsymbol{\theta}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\theta})) \;\Big|\; \boldsymbol{x} \in \{-1,+1\}^N, \boldsymbol{\theta} \in \Re^N \Big\}.$$

Since $c_0(\boldsymbol{x})$ is linear in $\{x_i\}$, M_0 is identical to M_D. Let $c_0(\boldsymbol{x}) = \boldsymbol{\alpha}\cdot\boldsymbol{x}$, where $\boldsymbol{\alpha} = \beta\tilde{\boldsymbol{x}}$ for turbo codes and $\boldsymbol{\alpha} = \beta\mathbf{1}_N$ for LDPC codes. The new coordinate $\boldsymbol{\theta}$ is obtained by shifting the old one, $\boldsymbol{\theta}_{\mathrm{old}}$, in (9) by $\boldsymbol{\alpha}$:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathrm{old}} - \boldsymbol{\alpha}, \quad \varphi_0(\boldsymbol{\theta}) = \varphi(\boldsymbol{\theta} + \boldsymbol{\alpha}).$$

We use the new coordinates $\boldsymbol{\theta}$ as a coordinate system of M_0, in which information from the constituent decoders is

integrated. We define the expectation parameter as $\boldsymbol{\eta}_0(\boldsymbol{\theta})$, which is another coordinate system of M_0 and is dual to $\boldsymbol{\theta}$:

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} p_0(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x} = \partial_{\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}). \tag{10}$$

Next, we consider the submanifold primarily responsible for only one $c_r(\boldsymbol{x})$. The submanifold, M_r, $r = 1, \cdots, K$ ($K = 2$ for turbo codes), is defined by

$$M_r = \Big\{ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) = \exp(c_0(\boldsymbol{x}) + c_r(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} - \varphi_r(\boldsymbol{\zeta}_r)) \Big| \ \boldsymbol{x} \in \{-1,+1\}^N, \boldsymbol{\zeta}_r \in \Re^N \Big\}.$$

Here, $\boldsymbol{\zeta}_r$ is the e–affine coordinate system or the natural parameters of M_r, through which information of the other decoders is integrated. M_r is also an e–flat submanifold of S. However, $M_r \neq M_0$ and $M_r \neq M_{r'}$, $r \neq r'$, because $c_r(\boldsymbol{x})$ includes higher order correlations of $\{x_i\}$ and $c_r(\boldsymbol{x}) \neq c_{r'}(\boldsymbol{x})$. The expectation parameter for M_r is defined as

$$\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \sum_{\boldsymbol{x}} p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)\boldsymbol{x} = \partial_{\boldsymbol{\zeta}_r}\varphi_r(\boldsymbol{\zeta}_r). \tag{11}$$

We show that the soft decoding is the m–projection to M_0 of the posterior distribution. Let us consider the m–projection of $q(\boldsymbol{x})$ to M_0. The derivative of $D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})]$ with respect to $\boldsymbol{\theta}$ is

$$\begin{aligned}\partial_{\boldsymbol{\theta}} D[q(\boldsymbol{x}); p_0(\boldsymbol{x};\boldsymbol{\theta})] &= \partial_{\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}) - \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x} \\ &= \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}.\end{aligned}$$

By the definition of the m–projection, this vanishes at the projected point. Hence, the m–affine coordinate of the projected point $\boldsymbol{\theta}^*$ is given by $\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) = \sum_{\boldsymbol{x}} q(\boldsymbol{x})\boldsymbol{x}$,

$$\eta_{0,i}(\theta_i^*) = \sum_{\boldsymbol{x}} q(\boldsymbol{x}) x_i = \sum_{x_i} q(x_i) x_i,$$

which shows that the m–projection of $q(\boldsymbol{x})$ does not change the expectation of $\boldsymbol{x}$. This is equivalent to the soft decoding defined in (3) (Fig.3).

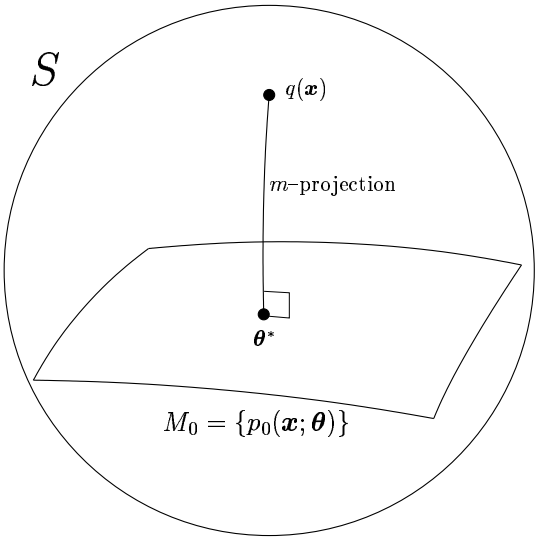

Fig. 3. Information geometry of MPM decoding.

V. Information Geometry of Turbo Decoding

The goal of the turbo decoding is to obtain a good approximation to the MPM decoding for $q(\boldsymbol{x}) = p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. Although obtaining the m–projection of $q(\boldsymbol{x})$ to M_0 is not tractable, evaluation of the m–projection of any distribution $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r) \in M_r$, $r = 1, 2$ to M_0 is tractable with BCJR algorithm. Since each $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, $r = 1, 2$, is derived from $p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_r|\boldsymbol{x})$ and a prior $\omega(\boldsymbol{x};\boldsymbol{\zeta}_r) \in M_D$, we can describe the turbo decoding as a method to approximate the m–projection of $q(\boldsymbol{x})$ to M_0 by evaluating the prior of $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$ iteratively and projecting $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$ to M_0.

A. Information Geometrical Definition of Turbo Decoding

We rewrite the turbo decoding algorithm in section II-A in the information geometrical framework. It is convenient to use an adequate e–affine coordinate system of M for the m–projection of $q(\boldsymbol{x})$ to M. Let $\pi_M \circ q(\boldsymbol{x})$ denote the coordinates $\boldsymbol{\theta}$ of M corresponding to the m–projected distribution:

$$\pi_M \circ q(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{\theta} \in \Re^N} D[q(\boldsymbol{x}); p(\boldsymbol{x};\boldsymbol{\theta})].$$

Turbo decoding (information geometrical view)

1) Let $\boldsymbol{\zeta}_2^t = \mathbf{o}$ for $t = 0$. For $t = 0, 1, 2, \cdots$, compose $p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^t) \in M_2$ with prior $\boldsymbol{\zeta}_2^t$.
2) Perform the m–projection of $p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^t)$ to M_0 as $\pi_{M_0} \circ p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^t)$, and update $\boldsymbol{\zeta}_1^{t+1}$ by using
$$\boldsymbol{\zeta}_1^{t+1} = \pi_{M_0} \circ p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^t) - \boldsymbol{\zeta}_2^t. \tag{12}$$
3) Compose $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^{t+1}) \in M_1$. Perform the m–projection of $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^{t+1})$ to M_0 as $\pi_{M_0} \circ p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^{t+1})$ and update $\boldsymbol{\zeta}_2^{t+1}$ by using
$$\boldsymbol{\zeta}_2^{t+1} = \pi_{M_0} \circ p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^{t+1}) - \boldsymbol{\zeta}_1^{t+1}. \tag{13}$$
4) If $\pi_{M_0} \circ p_1(\boldsymbol{x};\boldsymbol{\zeta}_1^{t+1}) \neq \pi_{M_0} \circ p_2(\boldsymbol{x};\boldsymbol{\zeta}_2^{t+1})$, go to step 1.

To clarify this procedure, we introduce three auxiliary parameters $\boldsymbol{\theta}$, $\boldsymbol{\xi}_1$, and $\boldsymbol{\xi}_2$:

$$\boldsymbol{\theta} = \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2, \quad \boldsymbol{\xi}_1 = \boldsymbol{\theta} - \boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_2, \quad \boldsymbol{\xi}_2 = \boldsymbol{\theta} - \boldsymbol{\zeta}_2 = \boldsymbol{\zeta}_1,$$

where $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ are equivalent to the extrinsic parameters in section II-A. The intuition behind this framework is as follows. Each of the higher order correlation terms, $c_1(\boldsymbol{x})$ or $c_2(\boldsymbol{x})$, is included only in Decoder 1 or Decoder 2, respectively. Decoders 1 and 2 calculate, using the m–projection, the linear approximations $\boldsymbol{\xi}_1 \cdot \boldsymbol{x}$ and $\boldsymbol{\xi}_2 \cdot \boldsymbol{x}$ of $c_1(\boldsymbol{x})$ and $c_2(\boldsymbol{x})$ and send messages $\boldsymbol{\xi}_1$ and $\boldsymbol{\xi}_2$ to the other decoders. In the interactive procedures, Decoder 1 forms the distribution $p_1(\boldsymbol{x};\boldsymbol{\zeta}_1)$, in which the nonlinear effect other than $c_1(\boldsymbol{x})$ (that is, $c_2(\boldsymbol{x})$ in the turbo decoding case of $K = 2$) is replaced by the estimate $\boldsymbol{\zeta}_1$, which is equal to the message $\boldsymbol{\xi}_2$ sent from Decoder 2. In the general case of $K > 2$, $\boldsymbol{\zeta}_1$ summarizes all the messages, $\boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_K$, from the other decoders. The same explanation holds for Decoder 2. The total linear estimate to the overall higher order term $c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})$ is given by $\boldsymbol{\theta} \cdot \boldsymbol{x} = \boldsymbol{\xi}_1 \cdot \boldsymbol{x} + \boldsymbol{\xi}_2 \cdot \boldsymbol{x}$.

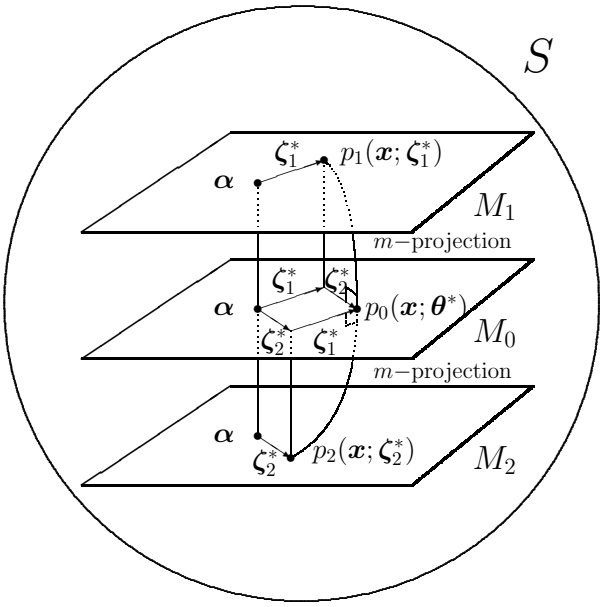

Fig. 4. Information geometrical view of turbo decoding.

The idea of the turbo decoding is schematically shown in Fig.4. The projected distribution is written as

$$\begin{aligned} p_0(\boldsymbol{x};\boldsymbol{\theta}) &= \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\theta})) \\ &= \exp(c_0(\boldsymbol{x}) + \boldsymbol{\xi}_1\cdot\boldsymbol{x} + \boldsymbol{\xi}_2\cdot\boldsymbol{x} - \varphi_0(\boldsymbol{\theta})). \end{aligned}$$

B. Equilibrium of Turbo Decoding

Assume that the decoding algorithm converges to a distribution $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, where $*$ is used to denote the equilibrium point. The distribution $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ is the approximation of the m–projection of $q(\boldsymbol{x})$ to M_0. The estimated parameter $\boldsymbol{\theta}^*$ satisfies $\boldsymbol{\theta}^* = \pi_{M_0}\circ p_1(\boldsymbol{x};\zeta_1^*) = \pi_{M_0}\circ p_2(\boldsymbol{x};\zeta_2^*)$ and $\boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^* = \zeta_1^* + \zeta_2^*$ from the definition of the algorithm.

The converged distributions $p_1(\boldsymbol{x};\zeta_1^*)$, $p_2(\boldsymbol{x};\zeta_2^*)$, and $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ satisfy the two conditions:

1) m–condition:

$$\pi_{M_0}\circ p_1(\boldsymbol{x};\zeta_1^*) = \pi_{M_0}\circ p_2(\boldsymbol{x};\zeta_2^*) = \boldsymbol{\theta}^*.$$

2) e–condition:

$$\boldsymbol{\theta}^* = \boldsymbol{\xi}_1^* + \boldsymbol{\xi}_2^* = \zeta_1^* + \zeta_2^*. \tag{14}$$

The m–condition can be rewritten with the expectation parameter defined in (10) and (11) as

$$\boldsymbol{\eta}_1(\zeta_1^*) = \boldsymbol{\eta}_2(\zeta_2^*) = \boldsymbol{\eta}_0(\boldsymbol{\theta}^*).$$

In order to give an information geometrical view of these conditions, we define two submanifolds in S. The first is the m–flat submanifold, $M(\boldsymbol{\theta})$, which we call the equimarginal submanifold, attached to each $p_0(\boldsymbol{x};\boldsymbol{\theta}) \in M_0$. It is defined by

$$M(\boldsymbol{\theta}) = \Big\{ p(\boldsymbol{x}) \;\Big|\; p(\boldsymbol{x}) \in S, \sum_{\boldsymbol{x}} p(\boldsymbol{x})\boldsymbol{x} = \sum_{\boldsymbol{x}} p_0(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{x} = \boldsymbol{\eta}_0(\boldsymbol{\theta}) \Big\}.$$

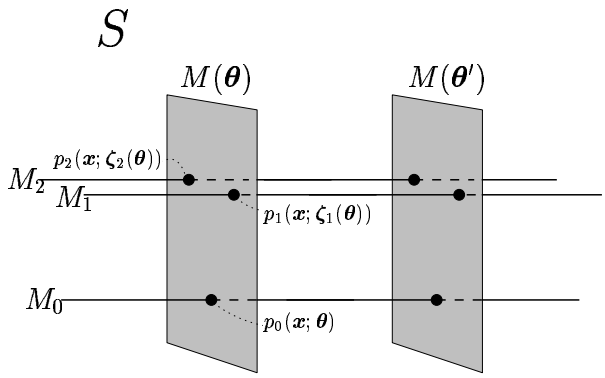

Fig. 5. Equimarginal submanifold $M(\boldsymbol{\theta})$.

The expectation of $\boldsymbol{x}$ is equal to $\boldsymbol{\eta}_0(\boldsymbol{\theta})$ for any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$. Hence, the m–projection of any $p(\boldsymbol{x}) \in M(\boldsymbol{\theta})$ to M_0 coincides with $p_0(\boldsymbol{x};\boldsymbol{\theta})$. In other words, $M(\boldsymbol{\theta})$ is the inverse image of $p_0(\boldsymbol{x};\boldsymbol{\theta})$ of the m–projection. For every $\boldsymbol{\theta}$, there exist unique $p_1(\boldsymbol{x};\zeta_1) \in M_1$ and $p_2(\boldsymbol{x};\zeta_2) \in M_2$ such that the expectations of $\boldsymbol{x}$ with respect to $p_r(\boldsymbol{x};\zeta_r)$ and $p_0(\boldsymbol{x};\boldsymbol{\theta})$ satisfy

$$\boldsymbol{\eta}_1(\zeta_1) = \boldsymbol{\eta}_2(\zeta_2) = \boldsymbol{\eta}_0(\boldsymbol{\theta}).$$

We denote the parameters that satisfy this equation by $\zeta_1(\boldsymbol{\theta})$ and $\zeta_2(\boldsymbol{\theta})$. In other words, we define $\zeta_1(\boldsymbol{\theta}) = \pi_{M_1}\circ p_0(\boldsymbol{x};\boldsymbol{\theta})$ and $\zeta_2(\boldsymbol{\theta}) = \pi_{M_2}\circ p_0(\boldsymbol{x};\boldsymbol{\theta})$. Obviously, $p_1(\boldsymbol{x};\zeta_1(\boldsymbol{\theta}))$, $p_2(\boldsymbol{x};\zeta_2(\boldsymbol{\theta})) \in M(\boldsymbol{\theta})$, and $\pi_{M_0}\circ p_1(\boldsymbol{x};\zeta_1(\boldsymbol{\theta})) = \pi_{M_0}\circ p_2(\boldsymbol{x};\zeta_2(\boldsymbol{\theta}))) = \boldsymbol{\theta}$; however generally, $\zeta_1(\boldsymbol{\theta}) + \zeta_2(\boldsymbol{\theta}) \neq \boldsymbol{\theta}$ except for the equilibrium point $\boldsymbol{\theta}^*$. The projection theorem shows that $M(\boldsymbol{\theta})$ is orthogonal to M_0, M_1, and M_2 (Fig.5), and that $p_r(\boldsymbol{x};\zeta_r(\boldsymbol{\theta}))$ is the intersection of M_r and $M(\boldsymbol{\theta})$.

In order to elucidate the e–condition, we next define an e–flat submanifold $E(\boldsymbol{\theta})$ connecting $p_0(\boldsymbol{x};\boldsymbol{\theta})$, $p_1(\boldsymbol{x};\zeta_1(\boldsymbol{\theta}))$, and $p_2(\boldsymbol{x};\zeta_2(\boldsymbol{\theta}))$ in a log-linear manner:

$$E(\boldsymbol{\theta}) = \Big\{ p(\boldsymbol{x}) = C p_0(\boldsymbol{x};\boldsymbol{\theta})^{t_0} p_1(\boldsymbol{x};\zeta_1(\boldsymbol{\theta}))^{t_1} p_2(\boldsymbol{x};\zeta_2(\boldsymbol{\theta}))^{t_2} \;\Big|\; t_r \in \Re, \sum_{r=0}^{2} t_r = 1 \Big\},\ C: \text{normalization factor.}$$

This manifold is a two-dimensional e–affine subspace of S. Apparently, $p_0(\boldsymbol{x};\boldsymbol{\theta})$, $p_1(\boldsymbol{x};\zeta_1(\boldsymbol{\theta}))$, and $p_2(\boldsymbol{x};\zeta_2(\boldsymbol{\theta}))$ belongs to $E(\boldsymbol{\theta})$. Moreover, at the equilibrium $\boldsymbol{\theta}^*$, $q(\boldsymbol{x})$ is included in $E(\boldsymbol{\theta}^*)$. This is easily proved by setting $t_0 = -1$, $t_1 = t_2 = 1$, and (14)

$$\begin{aligned} & C\frac{p_1(\boldsymbol{x};\zeta_1^*)p_2(\boldsymbol{x};\zeta_2^*)}{p_0(\boldsymbol{x};\boldsymbol{\theta}^*)} \\ &= C\exp(2c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x}) + (\zeta_1^* + \zeta_2^*)\cdot\boldsymbol{x} \\ &\qquad - (c_0(\boldsymbol{x}) + \boldsymbol{\theta}^*\cdot\boldsymbol{x})) \\ &= C\exp(c_0(\boldsymbol{x}) + c_1(\boldsymbol{x}) + c_2(\boldsymbol{x})) = q(\boldsymbol{x}). \end{aligned}$$

This discussion is summarized in the following theorem.

Theorem 3: At the equilibrium of the turbo decoding algorithm, $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\zeta_1^*)$, and $p_2(\boldsymbol{x};\zeta_2^*)$ belong to the equimarginal submanifold $M(\boldsymbol{\theta}^*)$, while its e–flat version, $E(\boldsymbol{\theta}^*)$, includes $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, $p_1(\boldsymbol{x};\zeta_1^*)$, $p_2(\boldsymbol{x};\zeta_2^*)$, and $q(\boldsymbol{x})$.

The theorem shows the information geometrical structure of the equilibrium point. If $M(\boldsymbol{\theta}^*)$ includes $q(\boldsymbol{x})$, $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ gives

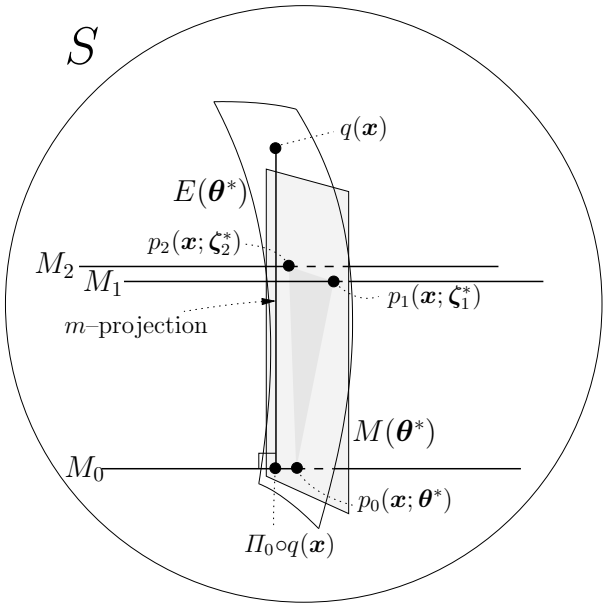

Fig. 6. $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$ of turbo decoding: $\Pi_0 \circ q(\boldsymbol{x})$ is the direct m–projection of $q(\boldsymbol{x})$ to M_0, which corresponds to the true "soft decoding" based on $q(\boldsymbol{x})$, while $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ is the equilibrium of the turbo decoding. The discrepancy between two submanifolds causes the decoding error.

the MPM decoding based on $q(\boldsymbol{x})$, since the soft decoding of $q(\boldsymbol{x})$ is equivalent to the m–projection of $q(\boldsymbol{x})$ to M_0, and $M(\boldsymbol{\theta}^*)$ is orthogonal to M_0 at $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$. However, since the m–flatness and the e–flatness do not coincide in general, $M(\boldsymbol{\theta}^*)$ does not necessarily include $q(\boldsymbol{x})$, while its e–flat version, $E(\boldsymbol{\theta}^*)$, includes $q(\boldsymbol{x})$ instead of $M(\boldsymbol{\theta}^*)$. This shows that the turbo decoding approximates the MPM decoding by replacing the m–flat manifold $M(\boldsymbol{\theta}^*)$ with the e–flat manifold $E(\boldsymbol{\theta}^*)$. It should be noted that $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$ is not the e–projection of $q(\boldsymbol{x})$ to M_0 either, because $E(\boldsymbol{\theta}^*)$ is not necessarily orthogonal to M_0. When it is orthogonal, it minimizes the KL–divergence $D[p_0(\boldsymbol{x};\boldsymbol{\theta}); q(\boldsymbol{x})]$, $\boldsymbol{\theta} \in \Re^N$, which gives the naive mean field approximation [23], [24]. The replacement of the m–projection by the e–projection shares the similar idea of the mean field approximation [10], [23]–[26]. Generally, there is a discrepancy between $M(\boldsymbol{\theta}^*)$ and $E(\boldsymbol{\theta}^*)$, which causes a decoding error (Fig.6). This suggests a possibility of a new method to improve the iterative decoding. We will study this in section VII.

C. Local Stability Analysis of Equilibrium Point

We discuss the local stability condition in this subsection. Let $I_0(\boldsymbol{\theta})$ be the Fisher information matrix of $p_0(\boldsymbol{x};\boldsymbol{\theta})$, and $I_r(\boldsymbol{\zeta}_r)$ be that of $p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)$, $r = 1, 2$. Since they belong to the exponential family, we have the following relations:

$$I_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}\boldsymbol{\theta}}\varphi_0(\boldsymbol{\theta}) = \partial_{\boldsymbol{\theta}}\boldsymbol{\eta}_0(\boldsymbol{\theta}),$$
$$I_r(\boldsymbol{\zeta}_r) = \partial_{\boldsymbol{\zeta}_r\boldsymbol{\zeta}_r}\varphi_r(\boldsymbol{\zeta}_r) = \partial_{\boldsymbol{\zeta}_r}\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r), \quad r = 1, 2.$$

Note that $I_0(\boldsymbol{\theta})$ is a diagonal matrix whose diagonal elements are

$$[I_0(\boldsymbol{\theta})]_{ii} = 1 - \eta_{0,i}^2.$$

In order to discuss the local stability, we give a sufficiently small perturbation, $\Delta\boldsymbol{\zeta}_2$, to $\boldsymbol{\zeta}_2^*$ and apply one step of the decoding procedure. Let $\boldsymbol{\zeta}_2' = \boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2'$ be the parameter after one step of the turbo decoding algorithm. From step 2, we have $\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta} = \pi_{M_0} \circ p_2(\boldsymbol{x}; \boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2)$, such that

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^* + \Delta\boldsymbol{\theta}) = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2^* + \Delta\boldsymbol{\zeta}_2).$$

By a simple expansion, we have

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^*) + I_0(\boldsymbol{\theta}^*)\Delta\boldsymbol{\theta} = \boldsymbol{\eta}_2(\boldsymbol{\zeta}_2^*) + I_2(\boldsymbol{\zeta}_2^*)\Delta\boldsymbol{\zeta}_2$$
$$\Delta\boldsymbol{\theta} = I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*)\Delta\boldsymbol{\zeta}_2.$$

Thus, $\boldsymbol{\zeta}_1$ in step 2 becomes

$$\boldsymbol{\zeta}_1 = \boldsymbol{\zeta}_1^* + (I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N)\Delta\boldsymbol{\zeta}_2.$$

Following the same line for step 3, $\Delta\boldsymbol{\zeta}_2'$ is given by

$$\begin{aligned}\Delta\boldsymbol{\zeta}_2' &= (I_0(\boldsymbol{\theta}^*)^{-1} I_1(\boldsymbol{\zeta}_1^*) - E_N)(I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N)\Delta\boldsymbol{\zeta}_2 \\ &= T_{turbo}\Delta\boldsymbol{\zeta}_2,\end{aligned}$$

where

$$T_{turbo} = (I_0(\boldsymbol{\theta}^*)^{-1} I_1(\boldsymbol{\zeta}_1^*) - E_N)(I_0(\boldsymbol{\theta}^*)^{-1} I_2(\boldsymbol{\zeta}_2^*) - E_N).$$

This shows that initial perturbation $\Delta\boldsymbol{\zeta}_2$ becomes $T_{turbo}\Delta\boldsymbol{\zeta}_2$ after one iteration.

Theorem 4: When $|\lambda_i| < 1$ for all i, where λ_i are the eigenvalues of the matrix T_{turbo}, the equilibrium point is locally stable.

This theorem coincides with the result of Richardson [12].

VI. Information Geometry of LDPC Decoding

A. Information Geometry of Decoding Process

The LDPC decoding algorithm in subsection II-B is rewritten in the information geometrical framework as follows.

LDPC decoding (information geometrical view)

Initialization:

For $t = 0$, set $\boldsymbol{\zeta}_r^0 = \mathbf{o}$, $r = 1, \cdots, K$. For $t = 0, 1, 2, \cdots$, compose $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t) \in M_r$.

Horizontal step:

Calculate the $m-$projection of $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t)$ to M_0 and define $\boldsymbol{\xi}_r^{t+1}$, $r = 1, \cdots, K$ as

$$\boldsymbol{\xi}_r^{t+1} = \pi_{M_0} \circ p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^t) - \boldsymbol{\zeta}_r^t. \tag{15}$$

Vertical step:

Update $\boldsymbol{\zeta}_r^{t+1}$, $r = 1, \cdots, K$ and $\boldsymbol{\theta}^{t+1}$:

$$\boldsymbol{\theta}^{t+1} = \sum_{r=1}^{K} \boldsymbol{\xi}_r^{t+1}, \quad \boldsymbol{\zeta}_r^{t+1} = \boldsymbol{\theta}^{t+1} - \boldsymbol{\xi}_r^{t+1}.$$

Convergence:

If $\boldsymbol{\theta}^t$ does not converge, repeat the process by incrementing t by 1.

Here, $\boldsymbol{\xi}_r$ is a message from decoder r that expresses the contribution of $c_r(\boldsymbol{x})$, and $\boldsymbol{\theta}$ integrates all the messages. Each decoder summarizes the information from all the other

decoders in the form of the prior $\omega(\boldsymbol{x};\boldsymbol{\zeta}_r)$. For turbo decoding, K is equal to 2, and $\boldsymbol{\xi}_1=\boldsymbol{\zeta}_2$ and $\boldsymbol{\xi}_2=\boldsymbol{\zeta}_1$. Therefore, (12) and (13) are both equivalent to (15). The main difference between the turbo and LDPC decodings is that the turbo decoding updates $\boldsymbol{\zeta}_r$ sequentially, while the LDPC decoding updates them simultaneously.

B. Equilibrium and Stability

The equilibrium of the LDPC decoding algorithm satisfies the two conditions:

1) m–condition:

$$\pi_{M_0}\circ p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^*)=\boldsymbol{\theta}^*,\quad r=1,\cdots,K.$$

which can be rewritten with the expectation parameters as,

$$\boldsymbol{\eta}_0(\boldsymbol{\theta}^*)=\boldsymbol{\eta}_1(\boldsymbol{\zeta}_1^*)=\cdots=\boldsymbol{\eta}_K(\boldsymbol{\zeta}_K^*).$$

2) e–condition:

$$\boldsymbol{\theta}^*=\sum_{r=1}^K\boldsymbol{\xi}_r^*=\frac{1}{K-1}\sum_{r=1}^K\boldsymbol{\zeta}_r^*.$$

Theorem 3 holds for the LDPC decoding, in which the definitions of submanifold $E(\boldsymbol{\theta})$ must be extended as follows:

$$E(\boldsymbol{\theta})=\Big\{p(\boldsymbol{x})\ \Big|\ p(\boldsymbol{x})=Cp_0(\boldsymbol{x};\boldsymbol{\theta})^{t_0}\prod_{r=1}^K p_r(\boldsymbol{x};\boldsymbol{\zeta}_r(\boldsymbol{\theta}))^{t_r},$$
$$t_r\in\Re,\sum_{r=0}^K t_r=1\Big\}\ C:\text{normalization factor},$$

where $\boldsymbol{\zeta}_r(\boldsymbol{\theta})$ is defined as

$$\boldsymbol{\zeta}_r(\boldsymbol{\theta})=\pi_{M_r}\circ p_0(\boldsymbol{x};\boldsymbol{\theta}),\quad r=1,\cdots,K.$$

At the converged point, $q(\boldsymbol{x})$ is included in $E(\boldsymbol{\theta}^*)$, which can be proved by setting $t_0=-(K-1),t_1=t_2=\cdots=1$:

$$\begin{aligned}&C\frac{\prod_{r=1}^K p_r(\boldsymbol{x};\boldsymbol{\zeta}_r^*)}{p_0(\boldsymbol{x};\boldsymbol{\theta}^*)^{K-1}}\\&=C\exp\Big(Kc_0(\boldsymbol{x})+\sum_{r=1}^K c_r(\boldsymbol{x})+\sum_{r=1}^K\boldsymbol{\zeta}_r^*\cdot\boldsymbol{x}\\&\qquad-(K-1)c_0(\boldsymbol{x})-(K-1)\boldsymbol{\theta}^*\cdot\boldsymbol{x}\Big)\\&=C\exp(c_0(\boldsymbol{x})+c_1(\boldsymbol{x})+\cdots+c_K(\boldsymbol{x}))=q(\boldsymbol{x}).\end{aligned}$$

The above equation proves that Theorem 3 holds for the LDPC decoding.

We next show the local stability condition for the LDPC decoding. Consider a case in which a sufficiently small perturbation is added to the equilibrium: $\boldsymbol{\zeta}_r=\boldsymbol{\zeta}_r^*+\Delta\boldsymbol{\zeta}_r$. The next state after a vertical step and a horizontal step is denoted by $\boldsymbol{\zeta}_r'=\boldsymbol{\zeta}_r^*+\Delta\boldsymbol{\zeta}_r'$. After the perturbation is added, the vertical step gives $\boldsymbol{\xi}_r=\boldsymbol{\xi}_r^*+\Delta\boldsymbol{\xi}_r$, where

$$\begin{aligned}\Delta\boldsymbol{\xi}_r&=I_0(\boldsymbol{\xi}^*)^{-1}I_r(\boldsymbol{\zeta}_r^*)\Delta\boldsymbol{\zeta}_r-\Delta\boldsymbol{\zeta}_r\\&=(I_0(\boldsymbol{\theta}^*)^{-1}I_r(\boldsymbol{\zeta}_r^*)-E_N)\Delta\boldsymbol{\zeta}_r.\end{aligned}$$

Following the horizontal step, we have

$$\Delta\boldsymbol{\zeta}_r'=\sum_{r\neq s}^K(I_0(\boldsymbol{\theta}^*)^{-1}I_s(\boldsymbol{\zeta}_s^*)-E_N)\Delta\boldsymbol{\zeta}_s.$$

The local stability condition of the LDPC decoding is summarized as follows.

Theorem 5: The linearization of the dynamics of the LDPC decoding around the equilibrium is

$$\begin{pmatrix}\Delta\boldsymbol{\zeta}_1'\\\vdots\\\Delta\boldsymbol{\zeta}_K'\end{pmatrix}=T_{LDPC}\begin{pmatrix}\Delta\boldsymbol{\zeta}_1\\\vdots\\\Delta\boldsymbol{\zeta}_K\end{pmatrix},$$

where

$$T_{LDPC}=\begin{pmatrix}O&I_0^{-1}I_2-E_N&\cdots&I_0^{-1}I_K-E_N\\I_0^{-1}I_1-E_N&O&&\vdots\\\vdots&&\ddots&\vdots\\I_0^{-1}I_1-E_N&\cdots&\cdots&O\end{pmatrix},$$

$I_0=I_0(\boldsymbol{\theta}^*)$, and $I_r=I_r(\boldsymbol{\zeta}_r^*)$. The equilibrium is locally stable when every eigenvalue, λ_i, $i=1,\cdots,NK$, of T_{LDPC} satisfies $|\lambda_i|<1$.

The local stability condition generally depends on the syndrome vector $\tilde{\boldsymbol{y}}$. However, intuitively speaking, if $I_r{\approx}I_0$, all the eigenvalues of T_{LDPC} are small, which leads to a stable and quick convergence. When the final guess by the decoder r is close to the integrated guess by $p_0(\boldsymbol{x};\boldsymbol{\theta}^*)$, it is expected that $I_0^{-1}I_r\approx E_N$. From simulations of LDPC codes, we observe good convergence in many cases which implies $I_r{\approx}I_0$. This property originates from the sparsity of the parity check matrix.

VII. Analysis of Decoding Errors

A. Framework of Error Analysis

We have described the information geometrical framework of the decoding algorithms and have shown how the MPM decoding is approximated by these decoding algorithms. In this section we analyze the error of the approximation and give a correction term for improving the approximation [27], [28]. We also provide an explanation why the sparsity, i.e., low density, of the parity check matrix has an advantage.

For the following discussion, we define an extended family of distributions,

$$M_S=\{p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})\},$$

by using two sets of parameters: $\boldsymbol{\theta}=(\theta_1,\cdots,\theta_N)^T\in\Re^N$ and $\boldsymbol{v}=(v_1,\cdots,v_K)^T\in\Re^K$.

$$\begin{aligned}p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})&=\exp\Big(c_0(\boldsymbol{x})+\boldsymbol{\theta}\cdot\boldsymbol{x}+\sum_{r=1}^K v_rc_r(\boldsymbol{x})-\varphi(\boldsymbol{\theta},\boldsymbol{v})\Big)\\&=\exp(c_0(\boldsymbol{x})+\boldsymbol{\theta}\cdot\boldsymbol{x}+\boldsymbol{v}\cdot\boldsymbol{c}(\boldsymbol{x})-\varphi(\boldsymbol{\theta},\boldsymbol{v})),\\\varphi(\boldsymbol{\theta},\boldsymbol{v})&=\ln\sum_{\boldsymbol{x}}\exp(c_0(\boldsymbol{x})+\boldsymbol{\theta}\cdot\boldsymbol{x}+\boldsymbol{v}\cdot\boldsymbol{c}(\boldsymbol{x})),\\\boldsymbol{c}(\boldsymbol{x})&=(c_1(\boldsymbol{x}),\cdots,c_K(\boldsymbol{x}))^T.\end{aligned}$$

The family M_S is a $(K+N)$–dimensional exponential family. The manifolds $M_0 = \{p_0(\boldsymbol{x};\boldsymbol{\theta})\}$ and $M_r = \{p_r(\boldsymbol{x};\boldsymbol{\zeta}_r)\}$ are submanifolds of M_S since $M_0 = \{p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})|\boldsymbol{v}=\mathbf{o}\}$ and $M_r = \{p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})|\boldsymbol{v}=\boldsymbol{e}_r\}$, where $\boldsymbol{e}_r$ is the unit vector

$$\boldsymbol{e}_r = (0,\cdots,0,\underset{\substack{\uparrow\\ r}}{1},0,\cdots,0)^T.$$

It also includes $q(\boldsymbol{x})$, when we set $\boldsymbol{\theta}=\mathbf{o}$ and $\boldsymbol{v}=\mathbf{1}_K$:

$$\mathbf{1}_K = (\underbrace{1,\cdots,1}_{K})^T = \sum_{r=1}^{K}\boldsymbol{e}_r.$$

We denote the expectation parameter of $p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v}) \in M_S$ by $\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = (\eta_1(\boldsymbol{\theta},\boldsymbol{v}),\cdots,\eta_N(\boldsymbol{\theta},\boldsymbol{v}))^T$, which is given by

$$\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = \partial_{\boldsymbol{\theta}}\varphi(\boldsymbol{\theta},\boldsymbol{v}) = \sum_{\boldsymbol{x}} p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})\boldsymbol{x}.$$

B. Analysis of Equimarginal Submanifold $M(\boldsymbol{\theta}^*)$

Let $p(\boldsymbol{x};\boldsymbol{\theta},\boldsymbol{v})$ be the distributions included in the equimarginal submanifold $M(\boldsymbol{\theta}^*)$, where

$$\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*,\mathbf{o}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

This constraint makes $\boldsymbol{\theta}$ an implicit function of $\boldsymbol{v}$, which is denoted by $\boldsymbol{\theta}(\boldsymbol{v})$. Note that $\boldsymbol{\theta}^* = \boldsymbol{\theta}(\mathbf{o})$. More precisely,

$$\boldsymbol{\eta}(\boldsymbol{\theta}(\boldsymbol{v}),\boldsymbol{v}) = \boldsymbol{\eta}(\boldsymbol{\theta}(\mathbf{o}),\mathbf{o}) = \boldsymbol{\eta}(\boldsymbol{\theta}^*),$$

for any $\boldsymbol{v}$. We analyze how $\boldsymbol{\theta}$ changes from $\boldsymbol{\theta}^*$ as $\boldsymbol{v}$ changes from $\mathbf{o}$ and finally becomes $\mathbf{1}_K$. In the following, we resort to the perturbation analysis and evaluate the derivatives of $\boldsymbol{\theta}(\boldsymbol{v})$ up to the second order. We start by introducing the derivative $D/\partial\boldsymbol{v}$ along $M(\boldsymbol{\theta})$:

$$\mathbf{o} = \frac{D}{\partial\boldsymbol{v}}\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}} + \frac{\partial\boldsymbol{\eta}}{\partial\boldsymbol{v}}. \tag{16}$$

The structural quantities $\partial\boldsymbol{\eta}/\partial\boldsymbol{\theta}$ and $\partial\boldsymbol{\eta}/\partial\boldsymbol{v}$ are the parts of the Fisher information matrix of M_S, because $\boldsymbol{\eta} = \partial\varphi(\boldsymbol{\theta},\boldsymbol{v})/\partial\boldsymbol{\theta}$. We use the index notation in which suffixes i,j, and k are for $\boldsymbol{\theta}$ and r,s, and t are for $\boldsymbol{v}$. In the component form, $G_{\boldsymbol{\theta\theta}} = (\partial\boldsymbol{\eta}/\partial\boldsymbol{\theta})$ and $G_{\boldsymbol{\theta v}} = (\partial\boldsymbol{\eta}/\partial\boldsymbol{v})$ are defined as

$$g_{ij}(\boldsymbol{\theta}) = \frac{\partial\eta_i}{\partial\theta_j} = G_{ij}(\boldsymbol{\theta}), \quad g_{ir}(\boldsymbol{\theta}) = \frac{\partial\eta_i}{\partial v_r}.$$

Note that $G_{\boldsymbol{\theta\theta}} = I_0(\boldsymbol{\theta}^*)$ at $\boldsymbol{\theta}=\boldsymbol{\theta}^*, \boldsymbol{v}=\mathbf{o}$. From (16), $G_{\boldsymbol{\theta\theta}}$, and $G_{\boldsymbol{\theta v}}$ we have

$$\begin{aligned}\mathbf{o} &= I_0(\boldsymbol{\theta})\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}} + G_{\boldsymbol{\theta v}}(\boldsymbol{\theta})\\ \frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}} &= -I_0^{-1}(\boldsymbol{\theta})G_{\boldsymbol{\theta v}}(\boldsymbol{\theta}), \quad \tilde{G}_{\boldsymbol{\theta v}} = -\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}},\end{aligned} \tag{17}$$

which gives the first-order derivative. We defined $\tilde{G}_{\boldsymbol{\theta v}}$ as the negative of it. Similarly, from

$$\frac{D^2}{\partial\boldsymbol{v}\partial\boldsymbol{v}}\boldsymbol{\eta}(\boldsymbol{\theta},\boldsymbol{v}) = 0,$$

we have

$$\begin{aligned}I_0(\boldsymbol{\theta})\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}'} = &-T_{\boldsymbol{\theta vv}'} - T_{\boldsymbol{\theta\theta\theta}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}'}\\ &- T_{\boldsymbol{\theta\theta v}'}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}} - T_{\boldsymbol{\theta\theta v}}\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}'},\end{aligned} \tag{18}$$

where

$$T_{\boldsymbol{\theta\theta\theta}} = \frac{\partial^3\varphi}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}}, \quad T_{\boldsymbol{\theta\theta v}} = \frac{\partial^3\varphi}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}\partial\boldsymbol{v}}, \quad T_{\boldsymbol{\theta vv}'} = \frac{\partial^3\varphi}{\partial\boldsymbol{\theta}\partial\boldsymbol{v}\partial\boldsymbol{v}'}.$$

More explicitly, by using the index notation, we have

$$\begin{aligned}\sum_j g_{ij}\frac{\partial^2\theta_j}{\partial v_r\partial v_s} = &-T_{irs} - \sum_{j,k}T_{ijk}\frac{\partial\theta_j}{\partial v_r}\frac{\partial\theta_k}{\partial v_s}\\ &- \sum_j T_{ijr}\frac{\partial\theta_j}{\partial v_s} - \sum_j T_{ijs}\frac{\partial\theta_j}{\partial v_r}.\end{aligned}$$

By replacing $\partial\boldsymbol{\theta}/\partial\boldsymbol{v}$ in (18) with the result of (17), we get

$$\begin{aligned}I_0(\boldsymbol{\theta})\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}'} = &-T_{\boldsymbol{\theta vv}'} - T_{\boldsymbol{\theta\theta\theta}}\tilde{G}_{\boldsymbol{\theta v}}\tilde{G}_{\boldsymbol{\theta v}'}\\ &+ T_{\boldsymbol{\theta\theta v}}\tilde{G}_{\boldsymbol{\theta v}'} + T_{\boldsymbol{\theta\theta v}'}\tilde{G}_{\boldsymbol{\theta v}}.\end{aligned}$$

We evaluate $\partial\boldsymbol{\theta}/\partial\boldsymbol{v} = -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta})$ and $\partial^2\boldsymbol{\theta}/\partial\boldsymbol{v}^2$ at $(\boldsymbol{\theta},\boldsymbol{v}) = (\boldsymbol{\theta}^*,\mathbf{o})$ and approximate $\boldsymbol{\theta}(\boldsymbol{v})$ with the second order Taylor series expansion with respect to $\boldsymbol{v}$ around the point. The differential operator $D/d\boldsymbol{v}$ at $(\boldsymbol{\theta},\boldsymbol{v}) = (\boldsymbol{\theta}^*,\mathbf{o})$ is written as

$$\frac{D}{d\boldsymbol{v}}\Big|_{(\boldsymbol{\theta}^*,\mathbf{o})} = B = \frac{\partial}{\partial\boldsymbol{v}}\Big|_{\boldsymbol{v}=\mathbf{o}} - \tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\frac{\partial}{\partial\boldsymbol{\theta}}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*},$$

In the component form, it is

$$B_r = \frac{\partial}{\partial v_r}\Big|_{\boldsymbol{v}=\mathbf{o}} - \sum_i \tilde{g}_{ir}(\boldsymbol{\theta}^*)\frac{\partial}{\partial\theta_i}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*}.$$

Following some calculations, we have

$$\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}}\Big|_{(\boldsymbol{\theta}^*,\mathbf{o})} = -I_0(\boldsymbol{\theta}^*)^{-1}B^2\boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

We denote the (r,s) component of B^2 by $B_{rs} = B_rB_s$. Note that $B^2\boldsymbol{\eta}(\boldsymbol{\theta}^*) \neq \mathbf{o}$ while $(D^2/\partial\boldsymbol{v}\partial\boldsymbol{v})\boldsymbol{\eta}(\boldsymbol{\theta}^*) = \mathbf{o}$.

The second-order approximation of $\boldsymbol{\theta}(\boldsymbol{v})$ around $(\boldsymbol{\theta}^*,\mathbf{o})$ is given by

$$\begin{aligned}\boldsymbol{\theta}(\boldsymbol{v}) &= \boldsymbol{\theta}^* + \frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{v}}\Big|_{(\boldsymbol{\theta}^*,\mathbf{o})}\boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^T\frac{\partial^2\boldsymbol{\theta}}{\partial\boldsymbol{v}\partial\boldsymbol{v}}\Big|_{(\boldsymbol{\theta}^*,\mathbf{o})}\boldsymbol{v}\\ &= \boldsymbol{\theta}^* - \tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*)\boldsymbol{v} - \frac{1}{2}\boldsymbol{v}^TI_0^{-1}(\boldsymbol{\theta}^*)(B^2\boldsymbol{\eta}(\boldsymbol{\theta}^*))\boldsymbol{v}.\end{aligned}$$

By plugging $\boldsymbol{v}=\mathbf{1}_K$ into the formula, we have

$$\theta_i(\mathbf{1}_K) = \theta_i^* - \sum_r\tilde{g}_{ir}(\boldsymbol{\theta}^*) - \frac{1}{2}(I_0^{-1}(\boldsymbol{\theta}^*))_{ii}\Big(\sum_{r,s}B_{rs}\Big)\eta_i(\boldsymbol{\theta}^*), \tag{19}$$

which shows the point at which $M(\boldsymbol{\theta}^*)$ intersects the submodels $\{p(\boldsymbol{x};\boldsymbol{\theta},\mathbf{1}_K)\}$. Since $q(\boldsymbol{x})$ is given by $p(\boldsymbol{x};\mathbf{o},\mathbf{1}_K)$, $\boldsymbol{\theta}(\mathbf{1}_K)$ is related to the discrepancy of $q(\boldsymbol{x})$ and the iterative decoding result.

This result is based on the perturbation analysis, of which justification is outlined below. When ϵ is small, the Taylor expansion for function $f(x)$ is

$$f(\epsilon) = f(0) + f'(0)\epsilon + \frac{1}{2}f''(0)\epsilon^2 + O(\epsilon^3).$$

When we rescale $v = x/\epsilon$,

$$f(v) = f(0) + \epsilon f'(0)v + \frac{1}{2}\epsilon^2 f''(0)v^2 + O(\epsilon^3).$$

In our analysis of iterative decoding, $x = \epsilon$ corresponds to $v_r = 1$, where the k-th derivative is of order ϵ^k. We have assumed that the effects of $\boldsymbol{v}$ are small, and we take the expansion with respect to $\boldsymbol{v}$ in terms of ϵ. We finally set $\epsilon = 1$, and the results are valid in the above sense.

So far, we have only considered the m–condition in order to obtain the perturbation expansion of $\boldsymbol{\theta}(\boldsymbol{v})$. However, in view of the small-ϵ expansion described above, as well as the e–condition, $\boldsymbol{\theta}^*$ itself is a small quantity. This is because the "true" posterior $p(\boldsymbol{x}; \mathbf{0}, \epsilon \mathbf{1}_K)$ tends to $p(\boldsymbol{x}; \mathbf{0}, \mathbf{0})$ as $\epsilon \to 0$, so that the iterative decoding result $\boldsymbol{\theta}^*$ should also tends to $\mathbf{0}$ in the limit $\epsilon \to 0$. In order to obtain the expansion which readily shows that $\boldsymbol{\theta}^*$ is a small quantity in the sense mentioned above, we invoke the e–condition, which is expressed as

$$\boldsymbol{\theta}^* = -\sum_{r=1}^{K} (\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^*). \tag{20}$$

In order to conclude our analysis of the decoding error based on perturbation analysis, we consider two distributions:

$$p(\boldsymbol{x}; \boldsymbol{\theta}^*, \mathbf{o}) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\theta}^* \cdot \boldsymbol{x} - \varphi(\boldsymbol{\theta}^*, \mathbf{o}))$$
$$p(\boldsymbol{x}; \boldsymbol{\zeta}_r, \epsilon \boldsymbol{e}_r) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{\zeta}_r \cdot \boldsymbol{x} + \epsilon c_r(\boldsymbol{x}) - \varphi(\boldsymbol{\zeta}_r, \epsilon \boldsymbol{e}_r)).$$

Note that $p(\boldsymbol{x}; \boldsymbol{\theta}^*, \mathbf{o}) \equiv p_0(\boldsymbol{x}; \boldsymbol{\theta}^*)$, and $p(\boldsymbol{x}; \boldsymbol{\zeta}_r, \epsilon \boldsymbol{e}_r)\big|_{\epsilon=1} = p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r^*)$. Let $p(\boldsymbol{x}; \boldsymbol{\zeta}_r, \epsilon \boldsymbol{e}_r)$, $r = 1, \cdots, K$, be included in $M(\boldsymbol{\theta}^*)$. From the result of (19), $\boldsymbol{\zeta}_r - \boldsymbol{\theta}^*$ is approximated in the power series of ϵ:

$$\boldsymbol{\zeta}_r - \boldsymbol{\theta}^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*) \boldsymbol{e}_r \epsilon - \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) B_{rr} \boldsymbol{\eta}(\boldsymbol{\theta}^*) \epsilon^2.$$

This gives the approximation of $\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^*$ as $\epsilon \to 1$:

$$\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*) \boldsymbol{e}_r - \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) B_{rr} \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

Hence, from (20), $\boldsymbol{\theta}^*$ satisfies

$$\begin{aligned} \boldsymbol{\theta}^* &= -\sum_{r=1}^{K} (\boldsymbol{\zeta}_r^* - \boldsymbol{\theta}^*) \\ &\simeq \tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*) \mathbf{1}_K + \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_r B_{rr} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \end{aligned} \tag{21}$$

Consider another distribution,

$$\begin{aligned} p(\boldsymbol{x}; \boldsymbol{u}, \epsilon \mathbf{1}_K) = \exp(c_0(\boldsymbol{x}) + \boldsymbol{u} \cdot \boldsymbol{x} \\ + \epsilon \mathbf{1}_K \cdot \boldsymbol{c}(\boldsymbol{x}) - \varphi(\boldsymbol{u}, \epsilon \mathbf{1}_K)). \end{aligned}$$

which is included in $M(\boldsymbol{\theta}^*)$. Note that $p(\boldsymbol{x}; \mathbf{0}, \epsilon \mathbf{1}_K)\big|_{\epsilon=1} = q(\boldsymbol{x})$ and that $p(\boldsymbol{x}; \boldsymbol{u}, \epsilon \mathbf{1}_K)$ is included in $M(\boldsymbol{\theta}^*)$. As ϵ increases from 0 to 1, $\boldsymbol{u}$ becomes $\boldsymbol{u}^*$, and generally $\boldsymbol{u}^* \neq \mathbf{0}$, which means $q(\boldsymbol{x})$ is generally not included in $M(\boldsymbol{\theta}^*)$.

From the result of (19), we have

$$\boldsymbol{u}^* - \boldsymbol{\theta}^* \simeq -\tilde{G}_{\boldsymbol{\theta v}}(\boldsymbol{\theta}^*) \mathbf{1}_K - \frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_{r,s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{22}$$

From (21) and (22), we have

$$\boldsymbol{u}^* \simeq -\frac{1}{2} I_0^{-1}(\boldsymbol{\theta}^*) \sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

From the Taylor expansion, we have

$$\begin{aligned} \boldsymbol{\eta}(\mathbf{o}, \mathbf{1}_K) &\simeq \boldsymbol{\eta}(\boldsymbol{u}^*, \mathbf{1}_K) - \nabla_{\boldsymbol{\theta}} \boldsymbol{\eta}(\boldsymbol{\theta}^*) \boldsymbol{u}^* \\ &= \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2} \sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \end{aligned} \tag{23}$$

Note that $\boldsymbol{\eta}(\mathbf{o}, \mathbf{1}_K)$ is the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$ which is equivalent to the soft decoding based on $q(\boldsymbol{x})$. Therefore, (23) shows the difference between the ultimate goal of the decoding and the result of the iterative decoding.

We summarize the above analysis:

Theorem 6: Let $\boldsymbol{\eta}_{MPM} = \boldsymbol{\eta}(\mathbf{o}, \mathbf{1}_K)$ be the expectation of $\boldsymbol{x}$ with respect to $q(\boldsymbol{x})$, and $\boldsymbol{\eta}(\boldsymbol{\theta}^*)$ be the expectation with respect to the distribution obtained by the iterative decoding. Then, $\boldsymbol{\eta}_{MPM}$ is approximated by the decoding result $\boldsymbol{\eta}(\boldsymbol{\theta}^*)$ as follows

$$\boldsymbol{\eta}_{MPM} \simeq \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2} \sum_{r \neq s} B_{rs} \boldsymbol{\eta}(\boldsymbol{\theta}^*). \tag{24}$$

C. Remark on $B_{rs}\eta_i$

We remark here that the error term is related to the curvature of $M(\boldsymbol{\theta}^*)$ without giving details about the definition of the e– and m–curvatures. See Amari and Nagaoka [15] for the mathematical details. We have shown that $M(\boldsymbol{\theta})$ is m–flat. This implies that the embedding m–curvature tensor vanishes; that is,

$$H_{rs}^{(m)i} = \frac{D^2}{\partial v_r \partial v_s} \eta_i(\boldsymbol{v}) = 0.$$

On the other hand, $M(\boldsymbol{\theta})$ is not e–flat, so the embedding e–curvature is given by

$$H_{rs}^{(e)i} = \frac{D^2}{\partial v_r \partial v_s} \theta_i(\boldsymbol{v}).$$

Its covariant version is given by

$$H_{rs}^{(e)i} = B_{rs} \eta_i,$$

which shows that the error term is directly related to the e–curvature of $M(\boldsymbol{\theta}^*)$.

VIII. Improving Decoding Errors for LDPC codes

A. Structural Terms

The terms $B_{rs}\eta_i$ are given by the structural tensors G and T at $p_0(\boldsymbol{x}; \boldsymbol{\theta}) \in M_0$. For LDPC codes, they are given by

$$\begin{aligned} g_{ir} &= E_{p_0}[(x_i - \eta_i)(c_r(\boldsymbol{x}) - \bar{c}_r)], \\ T_{ijr} &= E_{p_0}[(x_i - \eta_i)(x_j - \eta_j)(c_r(\boldsymbol{x}) - \bar{c}_r)], \end{aligned}$$

where E_{p_0} denotes the expectation with respect to $p_0(\boldsymbol{x}; \boldsymbol{\theta})$, and

$$\bar{c}_r = E_{p_0}[c_r(\boldsymbol{x})] = \rho \tilde{y}_r \prod_{j \in \mathcal{L}_r} \eta_j.$$

Because the x_i's are independent with respect to $p_0(\boldsymbol{x}; \boldsymbol{\theta})$, the following relations hold and are used for further calculation:

$$E_{p_0}[x_i c_r(\boldsymbol{x})] = \begin{cases} \eta_i \bar{c}_r, & \text{when } i \notin \mathcal{L}_r \\ \dfrac{1}{\eta_i} \bar{c}_r, & \text{when } i \in \mathcal{L}_r \end{cases}$$

$$E_{p_0}[c_r(\boldsymbol{x}) c_s(\boldsymbol{x})] = \frac{1}{P_{rs}} \bar{c}_r \bar{c}_s,$$

where

$$P_{rs} = \begin{cases} \prod_{j \in \mathcal{L}_r \cap \mathcal{L}_s} \eta_j^2, & \text{when } \mathcal{L}_r \cap \mathcal{L}_s \neq \emptyset \\ 1, & \text{when } \mathcal{L}_r \cap \mathcal{L}_s = \emptyset \end{cases}.$$

The explicit forms of G and T are given in Appendix III.

B. Algorithm to Calculate Correction Term

From the result of Theorem 6, the soft-decoded $\boldsymbol{\eta}^*$ is improved by

$$\boldsymbol{\eta}_{MPM} = \boldsymbol{\eta}(\boldsymbol{\theta}^*) + \frac{1}{2}\sum_{r \neq s} B_{rs}\boldsymbol{\eta}(\boldsymbol{\theta}^*).$$

By calculating $B_{rs}\eta_i$ for $(r \neq s)$, (see Appendix IV), we give the algorithm to calculate correction term $B_{rs}\eta_i$ as follows.

---

1) Calculate
$$\bar{c}_r = E_{p_0}[c_r(\boldsymbol{x})].$$
2) Given i, search for the pair (r, s) which includes i, that is, $i \in \mathcal{L}_r$ and $i \in \mathcal{L}_s$. Calculate
$$B_{rs}\eta_i = 2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s \sum_{j \neq i} \frac{1-\eta_j^2}{\eta_j^2} h_{jr}h_{js}. \tag{25}$$
3) Given i, search for the pair (r, s) such that $i \in \mathcal{L}_r$ and $i \notin \mathcal{L}_s$. Calculate
$$B_{rs}\eta_i = \bar{c}_r\bar{c}_s\frac{1-\eta_i^2}{\eta_i}\Big(-\frac{1-P_{rs}}{P_{rs}} + \sum_j \frac{1-\eta_j^2}{\eta_j^2}h_{jr}h_{js}\Big). \tag{26}$$
4) The correction term is given by summing up over all (r, s) in the above two cases.

The summation in (25) runs over $j \in \mathcal{L}_r \cap \mathcal{L}_s \setminus i$, and that in (26) runs over $j \in \mathcal{L}_r \cap \mathcal{L}_s$. Thus, when the parity-check matrix is designed such that, for any r and s,

$$h_{ir}h_{is} = 1$$

holds for at most one i, that is, any two columns of the parity-check matrix have at most one overlapping positions of 1, all the principal terms of the correction vanish [29], which leads to the following theorem for LDPC codes.

Theorem 7: The principal term of the decoding error vanishes when parity-check matrix H has no pair of columns with an overlap of 1 more than once.

It is believed [5] that the average probability of a decoding error is small, when any two columns of parity-check matrix H do not have an overlap of 1 more than once. Intuitively, this avoidance prevents loops with length 4 from appearing in the graphical representation. Results of many experiments indicate that short loops are harmful for iterative decodings; that is, they worsen the decoding errors. Our result in Theorem 7 analytically supports this indication: the principal term of the decoding error vanishes when the parity-check matrix is sparse and there are no two columns with an overlap of 1 more than once. Loops longer than 4 do not contribute to the decoding error at least via the principal term (although they may have effects via higher order terms). Many LDPC codes have been designed to satisfy this criterion [5]. The analysis presented here can be extended in a straightforward manner to higher order perturbation analysis in order to quantify these effects.

It should be noted that our approach is different from the approaches commonly used to analyze the properties of iterative decoders since we do not consider any *ensemble* of codes. A typical reasoning found in the literature (e.g., [4]) is first to consider an ensemble of random parity-check matrices and show that the probability (over the ensemble) of short loops in the associated graph decreases to zero as the codelength tends to infinity while the column and row weights are kept finite. This means that the behavior of iterative decoders for codes with longer loops is the same as that in the loop-free case. The statistical-mechanical approach to performance analysis of Gallager-type codes [30] also assumes random ensembles. Our analysis, on the other hand, does not assume ensembles but allows the evaluation of the performance of the iterative decoders with any *single instance* of a the parity-check matrix with a finite codelength.

IX. Discussion and Conclusion

We have discussed the mechanism of the iterative decoding algorithms from the information geometrical viewpoint. We built a framework for analyzing the algorithms and used it to reveal their basic properties.

The problem of the turbo and LDPC decodings is summarized as a unified problem of marginalizing the probability distribution $q(\boldsymbol{x})$ in (2). This problem is common to the belief propagation for the loopy belief diagram in artificial intelligence [7] and the Bethe approximation in statistical physics [9]–[11]. In all of them, the direct marginalization of $q(\boldsymbol{x})$ is intractable, and only the marginalization of partial distributions $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, $r = 1, \cdots, K$, in (4), is possible.

The marginalization of $q(\boldsymbol{x})$ is approximated through iterative processes of adjusting $\{\boldsymbol{\zeta}_r\}$, marginalizing $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$, and integrating them into the approximated parameter $\boldsymbol{\theta}$. Both of the decoding algorithms were redefined with the information geometrical terms, and the conditions of the equilibrium were derived. They revealed an intuitive information geometrical meaning of the equilibrium point, which is summarized in Theorem 3. In the information geometrical terms, the ideal goal is to have the cross section of M_0 and an m–flat submanifold $M(\boldsymbol{\theta})$ including $q(\boldsymbol{x})$: however, instead of $M(\boldsymbol{\theta})$, an e–flat manifold $E(\boldsymbol{\theta})$ is used to obtain the decoding result. A new prospect arose from the theorem: the discrepancy between $M(\boldsymbol{\theta})$ and $E(\boldsymbol{\theta})$ gives the decoding error.

The principal term of the discrepancy was obtained through perturbation analysis, which is summarized in Theorem 6. The decoding error was given in (24), and the correction term gives a method for improving the existing decoding algorithms. Moreover, since the correction term strongly depends on the encoders, it gives a new suggestion for designing the codes. We have done the perturbation analysis up to the second order, and it is possible to extend it to higher order analysis in a straightforward fashion.

We also derived the local stability conditions in Theorems 4 and 5. Although Theorem 4 coincides with the results of Richardson [12], Theorem 5 presents a new result for the local stability condition of LDPC codes. The global convergence property is another issue [31] which is one of our future works.

The belief propagation algorithm is not directly connected to the gradient method of minimizing a cost function. It has been pointed out that the final result is at the critical point of the Bethe free energy [10], [11].

For $\boldsymbol{\zeta}_1, \cdots, \boldsymbol{\zeta}_K$, and $\boldsymbol{\theta}$, we define the following function of $\{\boldsymbol{\zeta}_r\}$ and $\boldsymbol{\theta}$:

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) = D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); q(\boldsymbol{x})] - \sum_{r=1}^{K} D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)].$$

The first term is rewritten as

$$D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); q(\boldsymbol{x})] = E_{p_0}[c_0(\boldsymbol{x})] + \boldsymbol{\theta} \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_0(\boldsymbol{\theta}) - \Big(\sum_{r=0}^{K} E_{p_0}[c_r(\boldsymbol{x})] + \ln C\Big).$$

The second term is rewritten as

$$\begin{aligned} &\sum_{r=1}^{K} D[p_0(\boldsymbol{x}; \boldsymbol{\theta}); p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)] \\ &= K(E_{p_0}[c_0(\boldsymbol{x})] + \boldsymbol{\theta} \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_0(\boldsymbol{\theta})) \\ &\quad - \sum_{r=1}^{K} (E_{p_0}[c_0(\boldsymbol{x})] + E_{p_0}[c_r(\boldsymbol{x})] + \boldsymbol{\zeta}_r \cdot \boldsymbol{\eta}_0(\boldsymbol{\theta}) - \varphi_r(\boldsymbol{\zeta}_r)). \end{aligned}$$

These three equations give

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) = (K-1)\varphi_0(\boldsymbol{\theta}) - \sum_{r=1}^{K} \varphi_r(\boldsymbol{\zeta}_r) - \ln C + \sum_{r=1}^{K} \boldsymbol{\zeta}_r \cdot (\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)).$$

Since $\ln C$ is a constant, we neglect it and redefine $\mathcal{F}(\{\boldsymbol{\zeta}_r\}; \boldsymbol{\theta})$:

$$\mathcal{F}(\{\boldsymbol{\zeta}_r\}, \boldsymbol{\theta}) = (K-1)\varphi_0(\boldsymbol{\theta}) - \sum_{r=1}^{K} \varphi_r(\boldsymbol{\zeta}_r) + \sum_{r=1}^{K} \boldsymbol{\zeta}_r \cdot (\boldsymbol{\eta}_0(\boldsymbol{\theta}) - \boldsymbol{\eta}_r(\boldsymbol{\zeta}_r)).$$

When the $p_0(\boldsymbol{x}; \boldsymbol{\theta}), p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r) \in M(\boldsymbol{\theta})$, the last term vanishes, and this function with constraint $\boldsymbol{\zeta}_r = \boldsymbol{\zeta}_r(\boldsymbol{\theta})$ or $\boldsymbol{\eta}_r(\boldsymbol{\zeta}_r) = \boldsymbol{\eta}_0(\boldsymbol{\theta})$ coincides with the free energy introduced by Kabashima and Saad [10] from the statistical physical viewpoint.

The advantage of the information geometrical framework lies in its generality. The framework is common not only to turbo and LDPC codes, but is also generally valid for the Bethe approximation, the belief propagation applied to a loopy belief diagram, and its variants such as TRP [32]. We have used this framework to integrate the statistical-mechanical method and an interesting idea of the CCCP algorithm [33] in a separate paper [34]. Another important extension will be found when we use different models of channels. It is easy to extend the result for any memoryless channel (see Appendix I), and by employing more complicated channels, which is one of our future works, we can derive wide varieties of the turbo and the LDPC type decoding algorithms.

This study is a first step toward information geometrical understanding of turbo and LDPC codes. By using the framework presented in this paper, we expect that further understanding will appear and new improvements will emerge.

Appendix I
Extension to General Memoryless Channel

The information geometrical framework in this paper can be easily extended to the case where the channel is a general binary-input memoryless channel, which includes various important channels, such as AWGN and Laplace channels. We show that the Bayes posterior distribution is expressed in the form of (2) for turbo codes. Its extension to LDPC codes is also simple.

The information bits $\boldsymbol{x} = (x_1, \cdots, x_N)^T, x_i \in \{-1, +1\}$ and two sets of parity bits $\boldsymbol{y}_1 = (y_{11}, \cdots, y_{1L})^T$, $\boldsymbol{y}_2 = (y_{21}, \cdots, y_{2L})^T, y_{1j}, y_{2j} \in \{-1, +1\}$ are transmitted through a memoryless channel. The receiver observes their noisy version as $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2)$. Since the channel is memoryless the following relation holds

$$p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x}) = p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}). \tag{27}$$

The Bayes posterior with the uniform prior is

$$\begin{aligned} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) &= \frac{p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})}{\sum_{\boldsymbol{x}} p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x})} = C p(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2|\boldsymbol{x}) \\ &= C p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_1|\boldsymbol{x}) p(\tilde{\boldsymbol{y}}_2|\boldsymbol{x}). \end{aligned} \tag{28}$$

For memoryless channels, each conditional distribution on the right hand side of (27) is formulated as

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \prod_{i=1}^{N} p(\tilde{x}_i|x_i), \quad p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x}) = \prod_{j=1}^{L} p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x})), \tag{29}$$

for $r = 1, 2$. Let us view $p(\tilde{x}_i|x_i)$ as a function of x_i, where $\tilde{x}_i$ is fixed. By defining λ_i as

$$\lambda_i = \frac{1}{2} \ln \frac{p(\tilde{x}_i|x_i = +1)}{p(\tilde{x}_i|x_i = -1)},$$

$p(\tilde{x}_i|x_i)$ is rewritten as

$$p(\tilde{x}_i|x_i) \propto \exp(\lambda_i x_i). \tag{30}$$

Note that λ_i is a function of $\tilde{x}_i$. We can also rewrite $p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x}))$ as follows.

$$p(\tilde{y}_{rj}|y_{rj}(\boldsymbol{x})) \propto \exp(\mu_{rj} y_{rj}), \tag{31}$$

where

$$\mu_{rj} = \frac{1}{2} \ln \frac{p(\tilde{y}_{rj}|y_{rj} = +1)}{p(\tilde{y}_{rj}|y_{rj} = -1)}, \quad r = 1, 2.$$

From (29), (30), and (31), we can rewrite (28) as

$$\begin{aligned} p(\boldsymbol{x}|\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}_1, \tilde{\boldsymbol{y}}_2) &= C \exp(\boldsymbol{\lambda} \cdot \boldsymbol{x} + \boldsymbol{\mu}_1 \cdot \boldsymbol{y}_1(\boldsymbol{x}) + \boldsymbol{\mu}_2 \cdot \boldsymbol{y}_2(\boldsymbol{x})), \\ \boldsymbol{\lambda} &= (\lambda_1, \cdots, \lambda_N)^T, \quad \boldsymbol{\mu}_r = (\mu_{r1}, \cdots, \mu_{rL})^T, \end{aligned} \tag{32}$$

which has the identical form to (2), where $c_0(\boldsymbol{x}) = \boldsymbol{\lambda} \cdot \boldsymbol{x}$, and $c_r(\boldsymbol{x}) = \boldsymbol{\mu}_r \cdot \boldsymbol{y}_r(\boldsymbol{x})$. Other distributions $p_0(\boldsymbol{x}; \boldsymbol{\theta})$ and $p_r(\boldsymbol{x}; \boldsymbol{\zeta}_r)$ are also expressed with $c_0(\boldsymbol{x})$ and $c_r(\boldsymbol{x})$, which shows the

information geometrical framework is valid for general binary-input memoryless channels.

Finally, we give practical form of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}_r$ for an AWGN channel. Let the noise variance of an AWGN channel be ς^2 and $p(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ becomes

$$\begin{aligned} p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) &= (2\pi\varsigma^2)^{-N/2}\exp\Big(-\sum_{i=1}^{N}\frac{(\tilde{x}_i-x_i)^2}{2\varsigma^2}\Big) \\ &= (2\pi\varsigma^2)^{-N/2}\exp\Big(\frac{-1}{2\varsigma^2}\sum_{i=1}^{N}(x_i^2-2\tilde{x}_i x_i+\tilde{x}_i^2)\Big). \end{aligned}$$

Since $x_i^2=1$ holds, it becomes

$$p(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = (2\pi\varsigma^2)^{-N/2}\exp\Big(\frac{1}{2\varsigma^2}(2\tilde{\boldsymbol{x}}\cdot\boldsymbol{x}-N-|\tilde{\boldsymbol{x}}|^2)\Big).$$

Following the same line for $p(\tilde{\boldsymbol{y}}_r|\boldsymbol{x})$, the Bayes posterior with the uniform prior is

$$\begin{gathered} p(\boldsymbol{x}|\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}_1,\tilde{\boldsymbol{y}}_2) = C\exp(\boldsymbol{\lambda}\cdot\boldsymbol{x}+\boldsymbol{\mu}_1\cdot\boldsymbol{y}_1(\boldsymbol{x})+\boldsymbol{\mu}_2\cdot\boldsymbol{y}_2(\boldsymbol{x})), \\ \boldsymbol{\lambda}=\frac{1}{\varsigma^2}\tilde{\boldsymbol{x}},\quad \boldsymbol{\mu}_r=\frac{1}{\varsigma^2}\tilde{\boldsymbol{y}}_r, \end{gathered}$$

which is identical to (32).

Appendix II
Soft Constraint and Hard Constraint

The LDPC decoding was reformulated with a positive real number ρ in section III-C. Since the "soft constraint" defined with a finite ρ differs from the "hard constraint," it is important to discuss the influence of ρ on the hard decoding results. In this section, we show both constraints give the same hard decoding result for a sufficiently large but finite ρ.

The posterior probability of $\boldsymbol{x}$ conditional to $\tilde{\boldsymbol{y}}$ in (6) is rewritten as

$$p(\boldsymbol{x}|\tilde{\boldsymbol{y}}) = C\exp(\beta\mathbf{1}_N\cdot\boldsymbol{x}+\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})).$$

Let $\boldsymbol{\eta}_\rho$ be the expectation of $\boldsymbol{x}$ with respect to $p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$ and $\hat{\boldsymbol{x}}_\rho$ be the hard decoding result

$$\boldsymbol{\eta}_\rho = \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\tilde{\boldsymbol{y}})\boldsymbol{x},\quad \hat{\boldsymbol{x}}_\rho = \mathrm{sgn}(\boldsymbol{\eta}_\rho).$$

The ultimate goal of the LDPC decoding based on the "hard constraint" is to calculate $\hat{\boldsymbol{x}}_\infty$ defined as

$$\hat{\boldsymbol{x}}_\infty = \mathrm{sgn}(\boldsymbol{\eta}_\infty) = \mathrm{sgn}\big(\lim_{\rho\to\infty}\boldsymbol{\eta}_\rho\big).$$

If $\hat{\boldsymbol{x}}_\rho=\hat{\boldsymbol{x}}_\infty$ holds for a finite ρ, both constraints give the same hard decoding result.

Let us define $\mathcal{X}_{\mathcal{Y}}$ as a set of $\boldsymbol{x}$ which satisfy $\tilde{\boldsymbol{y}}=\boldsymbol{y}(\boldsymbol{x})$. As $\rho\to\infty$, $p(\boldsymbol{x}|\tilde{\boldsymbol{y}})$ concentrates on $\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}$, and $\boldsymbol{\eta}_\infty$ is redefined as

$$\boldsymbol{\eta}_\infty = \lim_{\rho\to\infty}\sum_{\boldsymbol{x}} p(\boldsymbol{x}|\tilde{\boldsymbol{y}})\boldsymbol{x} = \frac{\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}\boldsymbol{x}}{\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}}.$$

Now, $\boldsymbol{\eta}_\rho$ is rewritten as

$$\begin{aligned} \boldsymbol{\eta}_\rho &= \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\tilde{\boldsymbol{y}};\rho)\boldsymbol{x} \\ &= \sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} Ce^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}e^{\rho K}\boldsymbol{x} + \sum_{\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}}} Ce^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}\boldsymbol{x} \\ &= Ce^{\rho K}\boldsymbol{\eta}_\infty\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}} + C\sum_{\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}\boldsymbol{x}. \end{aligned} \tag{33}$$

A component of $\hat{\boldsymbol{x}}_\rho$ is different from that of $\hat{\boldsymbol{x}}_\infty$, when the second term in (33) dominates the first term with the opposite sign. Such a case cannot occur if

$$e^{\rho K}\Delta_\infty > \frac{\sum_{\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}}{\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}}, \tag{34}$$

where Δ_∞ is the smallest absolute value of the components of $\boldsymbol{\eta}_\infty$.

A. Strict Bound

Since $\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})\le K-2$ for $\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}}$ and from (34), $\hat{\boldsymbol{x}}_\rho=\hat{\boldsymbol{x}}_\infty$ is guaranteed for $\rho>\rho_0$, where ρ_0 is defined as

$$\begin{aligned} e^{\rho_0 K}\Delta_\infty &= \frac{e^{\rho_0(K-2)}\sum_{\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}}{\sum_{\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}} e^{\beta\mathbf{1}_N\cdot\boldsymbol{x}}} \\ &= \frac{e^{\rho_0(K-2)}\omega_0(\boldsymbol{x}\notin\mathcal{X}_{\mathcal{Y}})}{\omega_0(\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}})} \\ \rho_0 &= \frac{1}{2}\Big\{\ln\Big(\frac{(1-\omega_0(\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}))}{\omega_0(\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}})}\Big) - \ln\Delta_\infty\Big\}. \end{aligned}$$

Here, $\omega_0(\boldsymbol{x})$ is the prior of $\boldsymbol{x}$ defined in (7). Roughly speaking, as N increases, $\ln(1-\omega_0(\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}}))$ becomes negligible, and $\ln\omega_0(\boldsymbol{x}\in\mathcal{X}_{\mathcal{Y}})$ increases proportional to $-N$, and the positive number ρ_0 grows proportional to N.

B. Approximate stochastic bound of ρ for Large N and K

We show by probabilistic arguments that a finite ρ, not increasing in proportion to N, is sufficient to guarantee that a component of $\hat{\boldsymbol{x}}_\rho$ is equal to that of $\hat{\boldsymbol{x}}_\infty$, when N and K are large. Let $\mathcal{T}$ be the set of the typical sequences of $\boldsymbol{x}$,

$$\mathcal{T} = \Big\{\boldsymbol{x}\ \Big|\ \frac{1}{N}\sum_{i=1}^{N}x_i\simeq(1-2\sigma)\Big\},$$

of which cardinality is $|\mathcal{T}|=e^{NH(\sigma)}$, where σ is the probability of each bit to be flipped through the BSC and $H(\sigma)$ is the entropy. It is known [5] that, when N is large, with probability almost equal to 1, the vector satisfying the "hard constraint" exists uniquely in $\mathcal{T}$. Let $\boldsymbol{x}_0$ be the vector, and $\boldsymbol{\eta}_\infty=\hat{\boldsymbol{x}}_\infty=\boldsymbol{x}_0$. In the following, we neglect terms of relatively exponentially small order, e^{-CN}, by stating "except for small order terms." We can rewrite (34) as

$$e^{\rho K} > \frac{\sum_{\boldsymbol{x}\in\mathcal{T},\boldsymbol{x}\neq\boldsymbol{x}_0} e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}e^{\beta N(1-2\sigma)}}{e^{\beta N(1-2\sigma)}} = \sum_{\boldsymbol{x}\in\mathcal{T},\boldsymbol{x}\neq\boldsymbol{x}_0} e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}. \tag{35}$$

Here, the summation is taken only for $\boldsymbol{x}\in\mathcal{T}$ by neglecting exponentially small order terms, and $\Delta_\infty=1$ is used. Now,

we evaluate $e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})}$. We consider a regular LDPC codes, where h_{ri} of the parity check matrix H is randomly chosen and non-zero elements per row in H is fixed to Q. Since

$$\tilde{y}_r y_r(\boldsymbol{x}) = \prod_{i\in\mathcal{L}_r}(x_{0i}x_i),$$

it is 1 when the number of $x_{0i}x_i=-1$, $i\in\mathcal{L}_r$, is even, and is -1 when it is odd. Let R and $1-R$ be the probability of $\tilde{y}_r y_r(\boldsymbol{x})=-1$ and $\tilde{y}_r y_r(\boldsymbol{x})=+1$, respectively. Then, we can easily write down the probabilities which stay to be finite as $N\to\infty$. Since $\boldsymbol{x}_0$ and $\boldsymbol{x}$ are typical sequences, when σ is small, the probability that $\mathcal{L}_r$ does not include those i for which $x_{0i}x_i=-1$ is given by

$$((1-\sigma)^2+\sigma^2)^Q \simeq 1-2Q\sigma.$$

The probability that the number of $x_{0i}x_i=-1$ is two is much smaller. Hence, for a sufficiently small σ,

$$\begin{aligned}1-R &= \mathrm{Prob}[\tilde{y}_r y_r(\boldsymbol{x})=+1] \simeq 1-2Q\sigma,\\ R &= \mathrm{Prob}[\tilde{y}_r y_r(\boldsymbol{x})=-1] \simeq 2Q\sigma.\end{aligned}$$

Because of the law of large numbers,

$$\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x}) \simeq KE[\tilde{y}_r y_r(\boldsymbol{x})] = K(1-2R).$$

Now, we rewrite (35) as,

$$e^{\rho K} > \sum_{\boldsymbol{x}\in T,\boldsymbol{x}\neq\boldsymbol{x}_0} e^{\rho\tilde{\boldsymbol{y}}\cdot\boldsymbol{y}(\boldsymbol{x})} \simeq e^{\rho K(1-2R)}\cdot e^{NH(\sigma)},$$

where, $e^{NH(\sigma)}$ is the number of typical sequences. This shows when

$$\rho > \frac{NH(\sigma)}{2KR},$$

the probability that a component of $\hat{\boldsymbol{x}}_\infty$ is different from that of $\hat{\boldsymbol{x}}_\rho$ is negligibly small. When σ and Q are small, this reduces to

$$\rho > \frac{NH(\sigma)}{4KQ\sigma} \simeq \frac{N}{4KQ}.$$

Since N and K are of the same order, the right-hand side does not grow with N.

Appendix III
Explicit Forms of G and T

Metric tensor G :

for g_{ij} :

$$g_{ij} = E_{p_0}[(x_i-\eta_i)(x_j-\eta_j)] = (1-\eta_i^2)\delta_{ij},$$

which is the diagonal matrix $I_0(\boldsymbol{\theta}^*)$.

for g_{ir} :

$$\begin{aligned}g_{ir} &= \mathrm{Cov}[x_i, c_r(\boldsymbol{x})] = \frac{1-\eta_i^2}{\eta_i}\bar{c}_r h_{ir},\\ \tilde{g}_{ir} &= (I_0^{-1}(\boldsymbol{\theta}^*)G_{\boldsymbol{\theta}\boldsymbol{v}})_{ir} = \frac{1}{\eta_i}\bar{c}_r h_{ir}.\end{aligned}$$

Skewness tensor T :

for T_{ijk} :

$$\begin{aligned}T_{ijk} &= E_{p_0}[(x_i-\eta_i)(x_j-\eta_j)(x_k-\eta_k)]\\ &= -2\eta_i(1-\eta_i^2)\delta_{ijk},\end{aligned}$$

where δ_{ijk} is equal to 1 when $i=j=k$ and 0 otherwise. Hence, it is diagonal.

for T_{ijr} :

$$\begin{aligned}T_{iir} &= -2h_{ir}(1-\eta_i^2)\bar{c}_r,\\ T_{ijr} &= h_{ir}h_{jr}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j}\bar{c}_r.\end{aligned}$$

for T_{irs} $(r\neq s)$:

$$T_{irs} = E_{p_0}[(x_i-\eta_i)(c_r(\boldsymbol{x})-\bar{c}_r)(c_s(\boldsymbol{x})-\bar{c}_s)]. \quad (36)$$

When $\mathcal{L}_r\cap\mathcal{L}_s=\emptyset$, $T_{irs}=0$. For $\mathcal{L}_r\cap\mathcal{L}_s\neq\emptyset$, we consider three cases.

case 1) $i\notin\mathcal{L}_r,\mathcal{L}_s$: In this case, x_i and $(c_r(\boldsymbol{x}), c_s(\boldsymbol{x}))$ are independent:

$$T_{irs}=0.$$

case 2) $i\in\mathcal{L}_r$, $i\in\mathcal{L}_s$: Careful calculation of (36) gives

$$T_{irs} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s.$$

case 3) $i\in\mathcal{L}_r$, $i\notin\mathcal{L}_s$ or $i\notin\mathcal{L}_r$, $i\in\mathcal{L}_s$: Careful calculation gives

$$T_{irs} = \bar{c}_r\bar{c}_s\Big\{-\frac{1-\eta_i^2}{\eta_i}+\frac{1-\eta_i^2}{\eta_i}\frac{1}{P_{rs}}\Big\}.$$

Appendix IV
Explicit Form of $B_{rs}\eta_i$ for $r\neq s$

First, we give the form of $B_{rs}\eta_i$ as follows,

$$\begin{aligned}B_{rs}\eta_i = &-T_{irs}-\sum_{jk}T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks}\\ &+\sum_j\big(T_{ijr}\tilde{G}_{js}+T_{ijs}\tilde{G}_{jr}\big).\end{aligned}$$

for $i\notin\mathcal{L}_r$, $i\notin\mathcal{L}_s$:

$$B_{rs}\eta_i = 0.$$

for $i\in\mathcal{L}_r$, $i\in\mathcal{L}_s$:

$$T_{irs} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s,$$

$$\sum_{jk}T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks} = T_{iii}\tilde{G}_{ir}\tilde{G}_{is} = -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s,$$

$$\begin{aligned}\sum_j T_{ijr}\tilde{G}_{js} &= T_{iir}\tilde{G}_{is}+\sum_{j\neq i}T_{ijr}\tilde{G}_{js}\\ &= -2\frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s\\ &\quad+\sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s\setminus i}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j^2}\bar{c}_r\bar{c}_s.\end{aligned}$$

Hence

$$B_{rs}\eta_i = 2\sum_{j\in\mathcal{L}_r\cap\mathcal{L}_s\setminus i}\frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i\eta_j^2}\bar{c}_r\bar{c}_s,$$

which vanishes when $\mathcal{L}_r \cap \mathcal{L}_s$ does not include any j other than i.

for $i \in \mathcal{L}_r$, $i \notin \mathcal{L}_s$ (or $i \in \mathcal{L}_s, i \notin \mathcal{L}_r$):

$$T_{irs} = \bar{c}_r \bar{c}_s \frac{1-\eta_i^2}{\eta_i}\Big(\frac{1}{P_{rs}} - 1\Big),$$
$$T_{ijk}\tilde{G}_{jr}\tilde{G}_{ks} = 0, \quad T_{ijs}\tilde{G}_{jr} = 0,$$
$$\sum_j T_{ijr}\tilde{G}_{js} = \sum_{j \in \mathcal{L}_r \cap \mathcal{L}_s} \frac{(1-\eta_i^2)(1-\eta_j^2)}{\eta_i \eta_j^2} \bar{c}_r \bar{c}_s.$$

Hence,

$$B_{rs}\eta_i = \frac{1-\eta_i^2}{\eta_i}\bar{c}_r\bar{c}_s\Big(-\frac{1-P_{rs}}{P_{rs}} + \sum_{j \in \mathcal{L}_r \cap \mathcal{L}_s} \frac{1-\eta_j^2}{\eta_j^2}\Big).$$

When $\mathcal{L}_r \cap \mathcal{L}_s = \{j\}$, $P_{rs} = \eta_j^2$, which reduces to

$$B_{rs}\eta_i = 0.$$

ACKNOWLEDGMENT

The authors gratefully acknowledge Chiranjib Bhattacharyya, Yoshiyuki Kabashima, and Motohiko Isaka for helpful discussions. The authors also wish to thank the Editor and the anonymous reviewers for their valuable comments.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. on Communications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[3] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[4] ——, *Low density parity check codes*, ser. Research Monograph series. Cambridge, MA: MIT Press, 1963.

[5] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo. CA: Morgan Kaufmann, 1988.

[7] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's "belief propagation" algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 140–152, Feb. 1998.

[8] Y. Kabashima and D. Saad, "Belief propagation vs. TAP for decoding corrupted messages," *Europhysics Lett.*, vol. 44, no. 5, pp. 668–674, Dec. 1998.

[9] ——, "Statistical mechanics of error-correcting codes," *Europhysics Lett.*, vol. 45, no. 1, pp. 97–103, Jan. 1999.

[10] ——, "The TAP approach to intensive and extensive connectivity systems," in *Advanced Mean Field Methods – Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2001, ch. 6, pp. 65–84.

[11] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Bethe free energy, Kikuchi approximations, and belief propagation algorithms," Mitsubishi Electric Research Laboratories, Tech. Rep. TR2001–16, May 2001.

[12] T. J. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, no. 1, pp. 9–23, Jan. 2000.

[13] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.

[14] S. Amari, *Differential-Geometrical Methods in Statistics*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer-Verlag, 1985, vol. 28.

[15] S. Amari and H. Nagaoka, *Methods of Information Geometry*. Providence, Rhode Island: AMS and Oxford University Press, 2000.

[16] A. Amraoui, S. Dusad, and R. Urbanke, "Achieving general points in the 2-user Gaussian MAC without time-sharing or rate-splitting by means of iterative coding," in *Proc. of 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 334.

[17] A. de Baynast and D. Declercq, "Gallager codes for multiple user applications," in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 335.

[18] R. Müller, G. Caire, and T. Tanaka, "Density evolution and power profile optimization for iterative multiuser decoders based on individually optimum multiuser detectors," in *Proc. 40th Allerton Conf. Commun., Contr., Comput.*, Monticello, IL, Oct. 2002.

[19] G. Caire, R. Müller, and T. Tanaka, "Iterative multiuser joint decoding: optimal power allocation and low-complexity implementation," 2003, submitted to IEEE Trans. Inform. Theory.

[20] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

[21] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[22] S. Amari, "Information geometry on hierarchy of probability distributions," *IEEE Trans. Inform. Theory*, vol. 47, no. 5, pp. 1701–1711, Jul. 2001.

[23] T. Tanaka, "Information geometry of mean-field approximation," *Neural Computation*, vol. 12, no. 8, pp. 1951–1968, Aug. 2000.

[24] ——, "Information geometry of mean-field approximation," in *Advanced Mean Field Methods – Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2001, ch. 17, pp. 259–273.

[25] S. Amari, S. Ikeda, and H. Shimokawa, "Information geometry and mean field approximation: The α-projection approach," in *Advanced Mean Field Methods – Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2001, ch. 16, pp. 241–257.

[26] H. J. Kappen and W. J. Wiegerinck, "Mean field theory for graphical models," in *Advanced Mean Field Methods – Theory and Practice*, M. Opper and D. Saad, Eds. Cambridge, MA: MIT Press, 2001, ch. 4, pp. 37–49.

[27] S. Ikeda, T. Tanaka, and S. Amari, "Information geometrical framework for analyzing belief propagation decoder," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 407–414.

[28] ——, "Information geometry of turbo codes," in *Proc. 2002 IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 114.

[29] T. Tanaka, S. Ikeda, and S. Amari, "Information-geometrical significance of sparsity in Gallager codes," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 527–534.

[30] T. Murayama, Y. Kabashima, D. Saad, and R. Vicente, "Statistical physics of regular low-density parity-check error-correcting codes," *Physical Review E*, vol. 62, no. 2, pp. 1577–1591, Aug. 2000.

[31] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 699–722, Feb. 2001.

[32] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-based reparameterization for approximate inference on loopy graphs," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 1001–1008.

[33] A. L. Yuille, "CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation," *Neural Computation*, vol. 14, no. 7, pp. 1691–1722, Jul. 2002.

[34] S. Ikeda, T. Tanaka, and S. Amari, "Stochastic reasoning, free energy and information geometry," to appear in Neural Computation.

Shiro Ikeda (M'00) received the B. Eng., M. Eng., and Dr. Eng. degrees in information physics from the University of Tokyo, Tokyo, Japan, in 1991, 1993, and 1996, respectively.

From 1996 to 2001, he was with the Brain-Style Information Systems Research Group, RIKEN Brain Science Institute, Saitama, Japan, former half as a Special Postdoctoral Researcher of RIKEN, and the latter half as a Reseacher of Japan Science and Technology Agency. He was an Associate Professor at Kyushu Institute of Technology, Fukuoka, Japan from 2001 to 2003 and since February 2003, he has been an Associate Professor at the Institute of Statistical Mathematics, Tokyo. He is now visiting the Gatsby Computational Neuroscience Unit, University College London, London, United Kingdom, under the fellowship between the Royal Society and the Japan Society for the Promotion of Science. His research interests are in the areas of statistical signal processing, learning theory, and information geometry.

Dr. Ikeda received the Best Research Award and Best Paper Award from Japan Neural Network Society, in 1999 and 2001, respectively.

Toshiyuki Tanaka (S'90-M'93) received the B. Eng., M. Eng., and Dr. Eng. degrees in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1988, 1990, and 1993, respectively.

In 1993, he joined the Department of Electronics and Information Engineering, Tokyo Metropolitan University, Tokyo, where he is currently an Associate Professor. His research interests are in the interdisciplinary areas of information and communication theory, learning theory, information geometry, and statistical mechanics.

Dr. Tanaka received the DoCoMo Mobile Science Award in 2002.

Shun-ichi Amari (M'71-M'88-F'94) graduated from the University of Tokyo, Tokyo, Japan, in 1958, majoring in mathematical engineering, and received the Dr. Eng. degree from the University of Tokyo in 1963.

He was an Associate Professor at Kyushu University, Fukuoka, Japan, an Associate and then Full Professor at the Department of Mathematical Engineering and Information Physics, University of Tokyo, and is now Professor-Emeritus at the University of Tokyo. He is the Director of RIKEN Brain Science Intsitute, Saitama, Japan. He has been engaged in research in wide are as of mathematical engineering and applied mathematics, such as topological network theory, differential geometry of continuum mechanics, pattern recognition, mathematical foundations of neural networks, and information geometry.

Dr. Amari served as President of the International Neural Network Society, Council member of Bernoulli Society for Mathematical Statistics and Probability Theory, and is President-Elect of the Institute of Electrical, Information and Communication Engineers, Japan. He was founding Co-Editor-in-Chief of Neural Networks. He has been awarded the Japan Academy Award, IEEE Neural Networks Pioneer Award, IEEE Emanuel R. Piore Award, Neurocomputing best paper award, IEEE Signal Processing Society best paper award, and NEC C&C Prize, among many others.