Generating functional analysis of LDGM channel coding with many short loops

Kazushi Mimura
Department of Information Sciences,
Hiroshima City University,
Hiroshima 731-3194, Japan
Email: mimura@hiroshima-cu.ac.jp

A. C. C. Coolen
Department of Mathematics,
King's College London,
Strand, London WC2R 2LS, UK
Email: ton.coolen@kcl.ac.uk

***Abstract*— We study the dynamics of a simple message-passing decoder for LDGM channel coding by using the generating functional analysis (GFA). The decoder addressed here is one of the simplest examples, which is characterized by a sparse random graph with many short loops. The GFA allows us to study the dynamics of iterative systems in an exact way in the large codeword length limit.**

I. Introduction

In order to improve the performance of message-passing iterative decoders for channel coding, it is important to understand its dynamics. For low-density parity-check (LDPC) codes, Richardson and Urbanke have studied the dynamics of the message-passing decoder by means of their density evolution method [1], [2] in the large codeword length limit. In their case the bipartite graph, which defines the ensemble of LDPC codes, can be regarded as a loop-free graph in the large codeword length limit, because the length of the loop is evaluated as $O(\log N)$ [1]. Therefore, it becomes valid to assume that all messages are independent for finite iterative steps.

However, it has not been discussed enough about the effect of loops in sparse random graphs so far. So we here study a simple message-passing iterative decoder, which is defined on a sparse random graph with many short loops, for low-density generator-matrix (LDGM) channel coding as a typical example for this kind of problems. In more general settings we should treat loops in the sparse random graphs. The generating functional analysis (GFA) [3] used here can evaluate the influence of loops exactly in the large codeword length limit. When problems are defined on finite size graphs, we cannot ignore loops. It is hard to treat finite size graphs directly, however the asymptotic analysis discussed here might be useful to evaluate the influence of loops as one of alternative approaches.

Up to now, Hatchett *et al.* have studied the dynamics of parallel update iterative equations on finitely connected random graphs by the GFA in the context of statistical physics [4]. They have treated the Poissonian graph, which is a sparse Erdös-Rényi graph. We have extended Hatchett's analysis to the sparse random graph with an arbitrary degree distribution [5]. In this paper we analyze the dynamics of iterative decoder on the sparse random graph with an arbitrary degree distribution for LDGM channel coding.

II. Background

First of all, we briefly summarize the Sourlas codes [6], which is considered as a dense case of LDGM channel coding. In r-body Sourlas codes, an original message $\boldsymbol{\xi}=(\xi_i)\in\{-1,1\}^N$ is encoded into a length-$\binom{N}{r}$ codeword $y^0_{i_1,\cdots,i_r}=\xi_{i_1}\cdots\xi_{i_r}$ ($\forall\ i_1<\cdots<i_r$), which are given by the products of r bits. It's sent to the receiver through noisy channels. The receiver gets the received message $\boldsymbol{y}=(y_{i_1,\cdots,i_r})\in\{-1,1\}^{\binom{N}{r}}$ with noise and decodes the original message based on the Bayesian method. To simplify a problem, hereafter we assume that the products of two bits ($r=2$) are sent through a binary symmetric channel (BSC) with bit flip probability p_0,

$$p(y_{ij}|\xi_i\xi_j)=\begin{cases}1-p_0 & \text{if } y_{ij}\xi_i\xi_j=1\\ p_0 & \text{if } y_{ij}\xi_i\xi_j=-1\end{cases}=\frac{e^{F_0 y_{ij}\xi_i\xi_j}}{2\cosh F_0},\quad (1)$$

where $F_0=\frac{1}{2}\ln\frac{1-p_0}{p_0}$. Assuming a uniform prior for $\boldsymbol{\xi}$, one obtains the posterior probability by the Bayes formula as

$$p(\boldsymbol{\xi}|\boldsymbol{y})\propto p(\boldsymbol{y}|\boldsymbol{\xi})p(\boldsymbol{\xi}),\quad (2)$$

where

$$p(\boldsymbol{y}|\boldsymbol{\xi})=\prod_{i=1}^N\prod_{j>i}^N p(y_{ij}|\xi_i\xi_j)\quad (3)$$

and $p(\boldsymbol{\xi})=2^{-N}$. The decoding is based on the marginal-posterior-mode (MPM) estimator as

$$\hat{\xi}_i=\underset{\xi_i\in\{-1,1\}}{\mathrm{argmax}}\sum_{\boldsymbol{\xi}\backslash\xi_i\in\{-1,1\}^{N-1}}p(\boldsymbol{\xi}|\boldsymbol{y}).\quad (4)$$

We first derive a simple decoding algorithm for this decoder by means of an approximate belief-propagation (BP) [7], which is a kind of message-passing algorithm. Defining notations p_k^1 and p_k^{-1} to represent the prior probability, the BP algorithm is given by the following equations.

$$\rho^{\pm1}_{(ij)k}=\sum_{\boldsymbol{\xi}\backslash\xi_k}p(y_{ij}|\xi_k=\pm1,\{\xi_{k'\neq k}\})\prod_{k'\neq k}q^{\xi_{k'}}_{(ij)k'},\quad (5)$$

$$q^{\pm1}_{(ij)k}=d_{(ij)k}p_k^{\pm1}\prod_{(i'j')\in\mathcal{L}(k)\backslash\{(ij)\}}\rho^{\pm1}_{(i'j')k},\quad (6)$$

where $d^{\pm 1}_{(ij)k}$ denotes a normalization constant to hold $q^{1}_{(ij)k} + q^{-1}_{(ij)k} = 1$ and $\mathcal{L}(k) \equiv \{(ik),(kj) | 1 \le i < k < j \le N\}$. One need $p^{\pm 1}_k = q^{\pm 1}_{(ij)k} = 1/2$ ($\forall$ $(ij), k$) as an initial condition. The pseudo-posterior can be calculated as

$$q^{\pm 1}_k = \alpha_k p^{\pm 1}_k \prod_{(ij) \in \mathcal{L}(k)} \rho^{\pm 1}_{(ij)k}, \tag{7}$$

where α_k is a normalization constant. We now make the following two assumptions to derive a simple iterative decoding algorithm.

Assumption 1: Any single codeword bit has small effect, namely, $q^{\xi_k}_{(ij)k} \simeq q^{\xi_k}_k$ *for all* k. □

Assumption 2: When we replace $q^{\xi_k}_{(ij)k}$ *in (5) with* $\Theta(q^{\xi_k}_k - q^{-\xi_k}_k)$*, it doesn't cause a big influence in inferring the original message.* □

Under these assumptions, the difference of $q^{\xi_k}_{(ij)k}$ concerning (ij) is ignored. Consequently, the graph on which the decoder is defined becomes equivalent to a graph which has many short loops. We here use the following relationship

$$\Theta(q^{\xi_k}_k - q^{-\xi_k}_k) = \frac{1 + \xi_k \,\mathrm{sgn}\,(q^1_k - q^{-1}_k)}{2} = \delta_{\xi_k, \hat{\xi}_k}, \tag{8}$$

where Θ denotes a step function and $\delta_{m,n}$ denotes Kronecker's delta. Then, log-likelihood ratio of $\rho^1_{(ij)k}$ becomes

$$\ln \frac{\rho^1_{(ij)k}}{\rho^{-1}_{(ij)k}} = \left\{ \begin{array}{ll} 2F_0 y_{ij} \hat{\xi}_j, & i = k, j \neq k \\ 2F_0 y_{ij} \hat{\xi}_i, & i \neq k, j = k \end{array} \right. . \tag{9}$$

We therefore have

$$\ln \frac{q^1_k}{q^{-1}_k} = \sum_{(ij) \in \mathcal{L}(k)} \ln \frac{\rho^1_{(ij)k}}{\rho^{-1}_{(ij)k}} = 2F_0 \left(\sum_{k'=1}^{k-1} y_{k'k} \hat{\xi}_{k'} + \sum_{k'=k+1}^{N} y_{kk'} \hat{\xi}_{k'} \right). \tag{10}$$

If we put $y_{ji} = y_{ij}$ for all $i < j$, the tentative decision $\hat{\xi}_i$ for ξ_i can be written by the following simple form:

$$\hat{\xi}_i = \mathrm{sgn}\,(q^1_i - q^{-1}_i) = \mathrm{sgn}\left(2F \sum_{j \neq i}^{N} y_{ij} \hat{\xi}_j \right), \tag{11}$$

where $F = \frac{1}{2} \ln \frac{1-p}{p}$ and p is a control parameter representing the estimate of the true bit flip probability p_0. We regard $\hat{\xi}_i$ as a random variable like this:

$$\hat{\xi}_i = \left\{ \begin{array}{ll} -1, & \text{with prob. } (1 - x_i)/2 \\ 1, & \text{with prob. } (1 + x_i)/2 \end{array} \right. , \tag{12}$$

thus the log-likelihood ratio becomes $\ln \frac{q^1_i}{q^{-1}_i} = 2 \tanh^{-1} x_i$. Replacing $\hat{x}_i$ into x_i and introducing the iterative step t, we have

$$x_i(t+1) = \tanh\left(F \sum_{j \neq i}^{N} y_{ij} x_j(t) \right). \tag{13}$$

from (11). The initial condition $x_i(0)$ is randomly and independently generated from the distribution $\frac{1}{2}\delta(x_i(0)+1) + \frac{1}{2}\delta(x_i(0)-1)$ for all i. It should be noted that if an equilibrium state of the decoder (13) is $\boldsymbol{x}$, then $-\boldsymbol{x}$ also becomes an equilibrium state. In this scheme, the code rate is $R = \binom{N}{2}^{-1} N \simeq 2/N$ for large N.

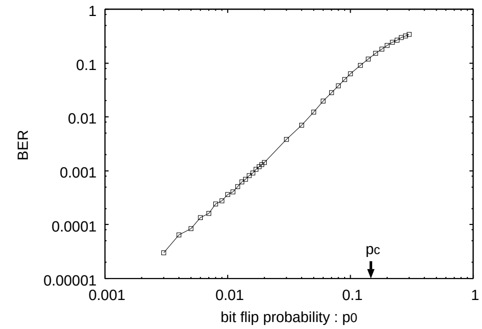

Fig. 1. Performance of the simple BP-based decoder with $p(k) = \delta_{k,3}$ and $d = 1$. The codeword length is $N = 10,000$. The code rate is $R = 2/5$. The critical noise level $p_c \simeq 0.146$ is the Shannon limit for $R = 2/5$.

III. LDGM channel coding

LDGM codes for channel coding are considered as the Sourlas codes whose codewords are thinned out sparsely [8], [9]. In this paper, to treat the effect of short loops we consider one of the simplest iterative decoder as follows.

Definition 1: The BP-based decoder duscussed here is defined as

$$x_i(t+1) = f(h_i(\boldsymbol{x}(t))), \tag{14}$$

by extending the decoder (13), where

$$h_i(\boldsymbol{x}(t)) = \sum_{j \neq i}^{N} c_{ij} y_{ij} x_j(t) + \theta_i(t) \tag{15}$$

and f *is an arbitrary function. The vector* $\boldsymbol{x}(t)$ *represents* $(x_1(t), \cdots, x_N(t))$*. We here introduce a connectivity parameter* $\boldsymbol{c} = (c_{ij}) \in \{0,1\}^{N^2}$ *which specify the realization of the graph. and* $\theta_i(t) \in \mathbb{R}$ *to represent more general settings.* □

It should be noted that Assumption 1 can be valid only in the large codeword length. Since we consider sparse codes in the decoder of (14), Assumption 1 is not justified. Due to this approximation, the encoder discussed here have not good error correcting performance. However, since this decoder has a very simple structure, we employ this decoder as the first step to treat loops.

We'll use $f(x) = \tanh(Fx)$, which may not be valid for sparse settings, as the function f. The connectivity parameter $\boldsymbol{c} = (c_{ij}) \in \{0,1\}^{N^2}$ are chosen randomly and independently according to

$$p_c(\boldsymbol{c}) = \frac{\left(\prod_{i=1}^{N} \prod_{j>i}^{N} p(c_{ij}) p(c_{ji}|c_{ij}) \right) \left(\prod_{i=1}^{N} \delta_{k_i, \sum_{j=1}^{N} c_{ij}} \right)}{\sum_{\boldsymbol{c}} \left(\prod_{i=1}^{N} \prod_{j>i}^{N} p(c_{ij}) p(c_{ji}|c_{ij}) \right) \left(\prod_{i=1}^{N} \delta_{k_i, \sum_{j=1}^{N} c_{ij}} \right)}, \tag{16}$$

with

$$p(c_{ij}) = \frac{c}{N}\delta_{c_{ij},1} + \left(1 - \frac{c}{N}\right)\delta_{c_{ij},0} \quad (17)$$

and

$$p(c_{ji}|c_{ij}) = \varepsilon\delta_{c_{ji},c_{ij}} + (1-\varepsilon)\left[\frac{c}{N}\delta_{c_{ji},1} + \left(1 - \frac{c}{N}\right)\delta_{c_{ji},0}\right]. \quad (18)$$

The degree of node i is defined as $k_i = \sum_{j=1}^{N} c_{ij}$ and is randomly and independently drawn from the given degree distribution $p(k)$. The average connectivity becomes $\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} = c$. The parameter ε, which denotes asymmetricity of c_{ij}, is introduced to evaluate bit error rate. We'll put $\varepsilon = 1$ later.

The received message can be represented by

$$y_{ij} = n_{ij}\xi_i\xi_j \quad (19)$$

for all $i < j$. Here, the random variable ξ_i denotes an element of the original message $\boldsymbol{\xi}$, where its distribution is $p(\xi_i) = \frac{1}{2}\delta(\xi_i + 1) + \frac{1}{2}\delta(\xi_i - 1)$. The random variable $n_{ij} \sim p(n_{ij})$ represents a channel noise with $p(n_{ij}) = p_0\delta(n_{ij} + 1) + (1 - p_0)\delta(n_{ij} - 1)$. We put $y_{ji} = y_{ij}$ and $n_{ji} = n_{ij}$ for all $i < j$. Taking the limit $p \to 0$, i.e., $F \to \infty$, this decoder with $f(x) = \tanh(Fx)$ becomes equivalent to Ising systems with the deterministic dynamics except for a normalization constant in $h_i(\boldsymbol{x}(t))$ [4], [5].

To avoid appearing two estimates $\boldsymbol{x}$ and $-\boldsymbol{x}$ with equal probability, we add the first dN bits of the original message ($0 \le d \le 1$) into the codeword. This information is used for the initial condition of (14). It's not necessary information essentially for the error correction. The length of the codeword is $M = \binom{N}{2}c/N + dN$, thus the code rate becomes $R = 2/(c + 2d)$ for large N.

The goal of the analysis is to obtain time evolution of bit error rate (BER) $P_b(t)$ as follows.

Definition 2: (bit error rate) The BER $P_b(t)$ of hard-decisions at the t^{th} iteration is defined as

$$P_b(t) = \frac{1 - m(t)}{2}, \quad m(t) = \frac{1}{N}\sum_{i=1}^{N}\xi_i \,\mathrm{sgn}\,[x_i(t)], \quad (20)$$

where the function $\mathrm{sgn}\,(x)$ *denotes the sign function taking 1 for $x \ge 0$ and -1 for $x < 0$.* □

We'll put $\theta_i(t) = 0$ later, therefore we can put $\xi_i = 1$ ($\forall i$) without loss of generality.

Figure 1 shows the performance of this decoder with $p(k) = \delta_{k,3}$ and $d = 1$. The BERs are $P_b(t)$ at $t = 10$. The simulations were carried out with $N = 10,000$, and all measurements were averaged over 20 runs. In this case the code rate becomes $R \simeq 2/5$. The critical noise level $p_c \simeq 0.146$ represents the Shannon limit $R = 1 - h_2(p_c) = 2/5$, where $h_2(x)$ denotes the binary entropy function defined as $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$.

Figures 2 and 3 shows schematic examples of graphs where this decoder is defined. These loops are originated from a peculiarity of the decoder. It should be noted that the updated quantities are attached to single bits and not to directed edges of the factor graph as the case of the usual message passing decoders. However it can be considered that this decoder is suitable as one of the simplest models in which we consider the effect of retarded self-interactions. The theoretical treatment addressed here makes clear how to treat loops and can be applied to other problems based on iterative algorithms.

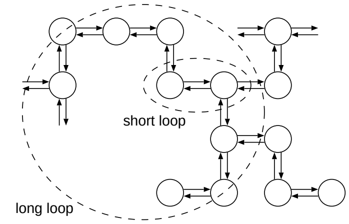

Fig. 2. Example of a graph where the iterative decoder is defined. Any state receives its own output two steps later.

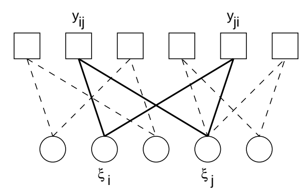

Fig. 3. Example of a tanner graph representation of the decoder. Since the decoder effectively has symmetry concerning the codeword, there are many length-4 loops (solid line).

IV. Generating functional analysis

Following [3] we assume that the macroscopic behaviour of the decoder depends only on the statistical properties of the random variables, i.e., the received message $\boldsymbol{y}$ and the realization of the graph $\boldsymbol{c}$. The decoder (14) is a kind of a Markov chain, so the joint distribution of $\boldsymbol{x}(0), \cdots, \boldsymbol{x}(t_m)$ is simply given by products of the individual transition probabilities $w[\boldsymbol{x}(t+1); \boldsymbol{x}(t)]$ of the chain:

$$p[\boldsymbol{x}(0), \cdots, \boldsymbol{x}(t_m)] = p_0[\boldsymbol{x}(0)]\prod_{t=0}^{t_m-1} w[\boldsymbol{x}(t+1); \boldsymbol{x}(t)], \quad (21)$$

where the transition probability is given by

$$w[\boldsymbol{x}(t+1); \boldsymbol{x}(t)] = \prod_{i=1}^{N}\delta[x_i(t+1) - f(h_i(\boldsymbol{x}(t)))]. \quad (22)$$

Here, δ denotes Dirac delta funciton. The probability $p_0[\boldsymbol{x}(0)]$ denotes a initial probability. To analyze the decoding dynamics we first define the following functional.

Definition 3: (averaged generating functional) The generating functional $Z[\boldsymbol{\psi}]$, which is averaged over the random variables, is defined by

$$\overline{Z[\boldsymbol{\psi}]} = \overline{\left\langle \exp\left[-i\sum_{i=1}^{N}\sum_{t=0}^{t_m}\psi_i(t)x_i(t)\right]\right\rangle} \quad (23)$$

where $\overline{[\cdots]}$ denotes average over $\{\boldsymbol{\xi},\boldsymbol{c},\boldsymbol{n}\}$, $\langle\cdots\rangle=\int_{\mathbb{R}^{N(t_m+1)}} d\boldsymbol{x}(0)\cdots d\boldsymbol{x}(t_m)\ p[\boldsymbol{x}(0),\cdots,\boldsymbol{x}(t_m)]\ (\cdots)$ and $d\boldsymbol{x}(0)=\prod_{i=1}^N dx_i(t)$. □

We here change the notation from $\boldsymbol{x}(t)=(x_1(t),\cdots,x_N(t))$ to $\boldsymbol{x}_i=(x_i(0),\cdots,x_i(t_m))$. Introducing the definition of $h_i(\boldsymbol{x}(t))$ by using δ fucntion, we then have

$$\overline{Z[\boldsymbol{\psi}]}=\overline{\left(\prod_{i=1}^N\int_{\mathbb{R}^{2t_m}}\frac{d\boldsymbol{h}_i d\hat{\boldsymbol{h}}_i}{(2\pi)^{t_m}}\right)\left(\prod_{i=1}^N\int_{\mathbb{R}^{(t_m+1)}}d\boldsymbol{x}_i\right)}$$
$$\times\exp\left[-i\sum_{i=1}^N\sum_{t=0}^{t_m}\hat{h}_i(t)\sum_{j\neq i}^N c_{ij}n_{ij}x_j(t)\right]$$
$$\times\exp\left[\sum_{i=1}^N\sum_{t=0}^{t_m-1}\{i\hat{h}_i(t)[h_i(t)+\theta_i(t)]-i\psi_i(t)x_i(t)\}\right]$$
$$\times\prod_{i=1}^N\prod_{t=0}^{t_m-1}\delta[x_i(t+1)-f(h_i(t))], \tag{24}$$

with $d\boldsymbol{h}_i=\prod_{t=0}^{t_m}dh_i(t)$, $d\hat{\boldsymbol{h}}_i=\prod_{t=0}^{t_m}d\hat{h}_i(t)$ and $d\boldsymbol{x}_i=\prod_{t=0}^{t_m}dx_i(t)$. We here use $y_{ij}=n_{ij}$ because of $\xi_i=1$ $(\forall i)$. To perform the average of the random variables in (24), we introduce the following integral expressions of Kronecker's delta

$$\delta_{k_i,\sum_{j=1}^N c_{ij}}=\int_0^{2\pi}\frac{d\omega_i}{2\pi}\exp\left[i\omega_i\left(k_i-\sum_{j=1}^N c_{ij}\right)\right], \tag{25}$$

which represents the constraint of the degree. Introducing the following functions

$$P(\boldsymbol{x},\hat{\boldsymbol{h}})=\frac{1}{N}\sum_{i=1}^N\delta(\boldsymbol{x}-\boldsymbol{x}_i)\delta(\hat{\boldsymbol{h}}-\hat{\boldsymbol{h}}_i)e^{-i\omega_i}, \tag{26}$$
$$Q(\boldsymbol{x},\hat{\boldsymbol{h}})=\frac{1}{N}\sum_{i=1}^N\delta(\boldsymbol{x}-\boldsymbol{x}_i)\delta(\hat{\boldsymbol{h}}-\hat{\boldsymbol{h}}_i), \tag{27}$$

(24) can be factorized concerning the index i. The averaged generating functional becomes

$$\overline{Z[\boldsymbol{\psi}]}=\int\{dPd\hat{P}\}\{dQd\hat{Q}\}e^{N\Psi[\{P,\hat{P},Q,\hat{Q}\}]} \tag{28}$$

with

$$\Psi[\{P,\hat{P},Q,\hat{Q}\}]=\frac{c(\varepsilon-2)}{2}$$
$$+i\int d\boldsymbol{x}\int d\hat{\boldsymbol{h}}\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})P(\boldsymbol{x},\hat{\boldsymbol{h}})+i\int d\boldsymbol{x}\int d\hat{\boldsymbol{h}}\hat{Q}(\boldsymbol{x},\hat{\boldsymbol{h}})Q(\boldsymbol{x},\hat{\boldsymbol{h}})$$
$$+\frac{c}{2}\int d\boldsymbol{x}\int d\hat{\boldsymbol{h}}\int d\boldsymbol{x}'\int d\hat{\boldsymbol{h}}'\hat{A}(\boldsymbol{x},\hat{\boldsymbol{h}};\boldsymbol{x}',\hat{\boldsymbol{h}}')$$
$$+\sum_{k=0}^\infty p(k)\ln\int d\boldsymbol{x}p_0[x(0)]\int\left(\prod_{t=0}^{t_m-1}\frac{dh(t)d\hat{h}(t)}{2\pi}e^{i\hat{h}(t)[h(t)-\theta(t)]}\right.$$
$$\left.\times\,\delta[x(t+1)-f(h(t))]\right)\frac{1}{k!}[-i\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})]^k e^{-i\hat{Q}(\boldsymbol{x},\hat{\boldsymbol{h}})} \tag{29}$$

and

$$\hat{A}(\boldsymbol{x},\hat{\boldsymbol{h}};\boldsymbol{x}',\hat{\boldsymbol{h}}')=P(\boldsymbol{x},\hat{\boldsymbol{h}})P(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle\varepsilon e^{-iy(\boldsymbol{x}\cdot\hat{\boldsymbol{h}}'+\boldsymbol{x}'\cdot\hat{\boldsymbol{h}})}\rangle_y$$
$$+Q(\boldsymbol{x},\hat{\boldsymbol{h}})P(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle(1-\varepsilon)e^{-iy\boldsymbol{x}\cdot\hat{\boldsymbol{h}}'}\rangle_y$$
$$+P(\boldsymbol{x},\hat{\boldsymbol{h}})Q(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle(1-\varepsilon)e^{-iy\boldsymbol{x}'\cdot\hat{\boldsymbol{h}}}\rangle_y, \tag{30}$$

where $\{dP,d\hat{P}\}=\prod_{\boldsymbol{x},\hat{\boldsymbol{h}}}(N/\sqrt{2\pi})dP(\boldsymbol{x},\hat{\boldsymbol{h}})d\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})$ and $\{dQ,d\hat{Q}\}=\prod_{\boldsymbol{x},\hat{\boldsymbol{h}}}(N/\sqrt{2\pi})dQ(\boldsymbol{x},\hat{\boldsymbol{h}})d\hat{Q}(\boldsymbol{x},\hat{\boldsymbol{h}})$. Unspecified integrals are over the range $\mathbb{R}^{t_m+1}$ concerning $d\boldsymbol{x}$ and $d\boldsymbol{x}'$ and over the range $\mathbb{R}^{t_m}$ concerning $d\boldsymbol{h}$, $d\hat{\boldsymbol{h}}$, $d\boldsymbol{h}'$ and $d\hat{\boldsymbol{h}}'$. The bracket $\langle\cdots\rangle_n$ denotes the average over n. This can be evaluated by steepest descent. Functional variation of Ψ with respect to $P(\boldsymbol{x},\hat{\boldsymbol{h}})$, $\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})$, $Q(\boldsymbol{x},\hat{\boldsymbol{h}})$ and $\hat{Q}(\boldsymbol{x},\hat{\boldsymbol{h}})$ gives the following saddle-point equations:

$$\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})=ic\int d\boldsymbol{x}'\int d\hat{\boldsymbol{h}}'P(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle\varepsilon e^{-iy(\boldsymbol{x}\cdot\hat{\boldsymbol{h}}'+\boldsymbol{x}'\cdot\hat{\boldsymbol{h}})}\rangle_y$$
$$+ic\int d\boldsymbol{x}'\int d\hat{\boldsymbol{h}}'Q(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle(1-\varepsilon)e^{-iy\boldsymbol{x}'\cdot\hat{\boldsymbol{h}}}\rangle_y,$$
$$P(\boldsymbol{x}',\hat{\boldsymbol{h}}')=\sum_{k=1}^\infty kp(k)\left\langle\frac{\delta[\boldsymbol{x}-\boldsymbol{x}']\delta[\hat{\boldsymbol{h}}-\hat{\boldsymbol{h}}']}{-i\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})}\right\rangle^*_{\boldsymbol{\theta},k},$$
$$\hat{Q}(\boldsymbol{x},\hat{\boldsymbol{h}})=ic\int d\boldsymbol{x}'\int d\hat{\boldsymbol{h}}'P(\boldsymbol{x}',\hat{\boldsymbol{h}}')\langle(1-\varepsilon)e^{-iy\boldsymbol{x}\cdot\hat{\boldsymbol{h}}'}\rangle_y,$$
$$Q(\boldsymbol{x}',\hat{\boldsymbol{h}}')=\sum_{k=0}^\infty p(k)\langle\delta[\boldsymbol{x}-\boldsymbol{x}']\delta[\hat{\boldsymbol{h}}-\hat{\boldsymbol{h}}']\rangle^*_{\boldsymbol{\theta},k},$$

with a measure $\langle\cdots\rangle^*_{\boldsymbol{\theta},k}$ which is defined as

$$\langle f(\boldsymbol{x},\hat{\boldsymbol{h}})\rangle^*_{\boldsymbol{\theta},k}=\frac{\int d\boldsymbol{x}\int d\hat{\boldsymbol{h}}f(\boldsymbol{x},\hat{\boldsymbol{h}})\hat{M}_k(\boldsymbol{x},\hat{\boldsymbol{h}}|\boldsymbol{\theta})}{\int d\boldsymbol{x}\int d\hat{\boldsymbol{h}}\hat{M}_k(\boldsymbol{x},\hat{\boldsymbol{h}}|\boldsymbol{\theta})}, \tag{31}$$
$$\hat{M}_k(\boldsymbol{x},\hat{\boldsymbol{h}}|\boldsymbol{\theta})=[-iP(\boldsymbol{x},\hat{\boldsymbol{h}})]^k e^{-iQ(\boldsymbol{x},\hat{\boldsymbol{h}})}p_0[x(0)]$$
$$\times\prod_{t=0}^{t_m-1}\int\frac{dh(t)d\hat{h}(t)}{2\pi}e^{i\hat{h}(t)[h(t)-\theta(t)]}\delta[x(t+1)-f(h(t))]. \tag{32}$$

Performing an inverse Fourier transformation, i.e., $P(\boldsymbol{x}|\boldsymbol{\theta}')\equiv\int d\hat{\boldsymbol{h}}e^{-i\boldsymbol{\theta}'\cdot\hat{\boldsymbol{h}}}P(\boldsymbol{x}|\hat{\boldsymbol{h}})$ and $Q(\boldsymbol{x}|\boldsymbol{\theta}')\equiv\int d\hat{\boldsymbol{h}}e^{-i\boldsymbol{\theta}'\cdot\hat{\boldsymbol{h}}}Q(\boldsymbol{x}|\hat{\boldsymbol{h}})$, gives

$$P(\boldsymbol{x}|\boldsymbol{\theta}')=\sum_{k=1}^\infty kp(k)\left\langle\frac{\delta[\boldsymbol{x}-\boldsymbol{x}']}{-i\hat{P}(\boldsymbol{x},\hat{\boldsymbol{h}})}\right\rangle^*_{\boldsymbol{\theta}+\boldsymbol{\theta}',k}, \tag{33}$$
$$Q(\boldsymbol{x}|\boldsymbol{\theta}')=\sum_{k=0}^\infty p(k)\langle\delta[\boldsymbol{x}-\boldsymbol{x}']\rangle^*_{\boldsymbol{\theta}+\boldsymbol{\theta}',k}. \tag{34}$$

Thus, $P(\boldsymbol{x}|\boldsymbol{\theta}')$ and $Q(\boldsymbol{x}|\boldsymbol{\theta}')$ are the averaged probability of finding a trajectory $\boldsymbol{x}$ for the given actual parameter $\boldsymbol{\theta}$ complemented by an amount $\boldsymbol{\theta}'$. This situation is similar to the other corresponding cases [4], [5], [10]. Putting $\theta(t)=0$, we arrive at the following compact form:

$$P(\boldsymbol{x}|\boldsymbol{\theta}') = \sum_{k=0}^{\infty} \frac{k+1}{c} p(k+1)$$
$$\times \left\langle \left(\prod_{l=1}^{k} \int d\boldsymbol{x}_l \left[\varepsilon P(\boldsymbol{x}_l|n_l\boldsymbol{x}) + (1-\varepsilon) Q(\boldsymbol{x}_l|\boldsymbol{0}) \right] \right) p_0[x(0)] \right.$$
$$\left. \times \prod_{t=0}^{t_m-1} \delta \left[x(t+1) - f\left(\theta'(t) + \sum_{l=1}^{k} n_l x_l(t) \right) \right] \right\rangle_{\{n_l\}}, \quad (35)$$

$$Q(\boldsymbol{x}|\boldsymbol{\theta}') = \sum_{k=0}^{\infty} p(k)$$
$$\times \left\langle \left(\prod_{l=1}^{k} \int d\boldsymbol{x}_l \left[\varepsilon P(\boldsymbol{x}_l|n_l\boldsymbol{x}) + (1-\varepsilon) Q(\boldsymbol{x}_l|\boldsymbol{0}) \right] \right) p_0[x(0)] \right.$$
$$\left. \times \prod_{t=0}^{t_m-1} \delta \left[x(t+1) - f\left(\theta'(t) + \sum_{l=1}^{k} n_l x_l(t) \right) \right] \right\rangle_{\{n_l\}}. \quad (36)$$

These equations are closed and exact, however it is no longer possible to simplify. For the discrete state case, i.e., the estimated bit flip probability $p \to 0$, the required field are also discrete, therefore the space is finite dimensional. The symmetric parameter is set to $\varepsilon = 1$ due to the definition of the decoder. Defining the new parameters $U(\boldsymbol{x}|\boldsymbol{x}') = P(\boldsymbol{x}|\boldsymbol{x}')$ and $V(\boldsymbol{x}) = Q(\boldsymbol{x}|\boldsymbol{0})$, we then arrived at the following proposition.

Proposition 1: The bit error rate $P_b(t)$ at the t^{th} iteration of (14) with $p = 0$ is evaluated as

$$P_b(t) = \frac{1-m(t)}{2}, \quad m(t) = \sum_{\boldsymbol{x} \in \{-1,1\}^{t_m+1}} x(t) V(\boldsymbol{x}). \quad (37)$$

The function $V(\boldsymbol{x}|\boldsymbol{0})$ can be obtained by

$$V(\boldsymbol{x}) = \sum_{k=0}^{\infty} p(k)$$
$$\times \left\langle \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_k} U(\boldsymbol{x}_1|n_1\boldsymbol{x}) \cdots U(\boldsymbol{x}_k|n_k\boldsymbol{x}) p_0[x(0)] \right.$$
$$\left. \times \prod_{t=0}^{t_m-1} \delta \left[x(t+1);\, \mathrm{sgn}\left(\sum_{l=1}^{k} n_l x_l(t) \right) \right] \right\rangle_{\{n_l\}} \quad (38)$$

from the solution of

$$U(\boldsymbol{x}|\boldsymbol{x}') = \sum_{k=0}^{\infty} \frac{k+1}{c} p(k+1)$$
$$\times \left\langle \sum_{\boldsymbol{x}_1} \cdots \sum_{\boldsymbol{x}_k} U(\boldsymbol{x}_1|n_1\boldsymbol{x}) \cdots U(\boldsymbol{x}_k|n_k\boldsymbol{x}) p_0[x(0)] \right.$$
$$\left. \times \prod_{t=0}^{t_m-1} \delta \left[x(t+1);\, \mathrm{sgn}\left(x'(t) + \sum_{l=1}^{k} n_l x_l(t) \right) \right] \right\rangle_{\{n_l\}}. \quad (39)$$

The initial probability is given by

$$p_0[x(0)] = \frac{1 + x(0)m(0)}{2}, \quad (40)$$

where $m(0) = (1-2p_0)d$. Here $\delta[m;n]$ denotes Kronecker's delta $\delta_{m,n}$. □

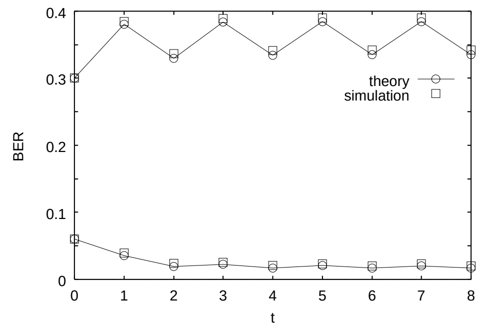

Fig. 4. Comparison between theoretical predictions and computer simulation results in the case of $p(k) = \delta_{k,3}$.

Equation (39) can be solved numerically by iteration. Figure 4 shows a comparison between theoretical predictions and computer simulation results in the case of $p(k) = \delta_{k,3}$. The GFA predictions exhibit good consistency with a computer simulation results.

V. Conclusion

We analyze the dynamics of a simple message-passing iterative decoder, which is defined on a sparse random graph with many short loops, for low-density generator-matrix (LDGM) channel coding as a typical example. We have derived an exact equation to represent time evolution of the bit error rate in the large codeword length limit. It can be considered that this kind of analysis contributes to providing a basis to evaluate effects of short loops. It is important to treat various length of loops. Thit is part of our future work.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments. This work was partially supported by a Grant-in-Aid for Encouragement of Young Scientists (B) No. 18700230 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] T. J. Richardson and R. L. Urbanke, *IEEE Trans. Info. Theory*, **47**, 2, 599 (2001).
[2] T. J. Richardson, M. A. Shokrollahi and R. L. Urbanke, *IEEE Trans. Info. Theory*, **47**, 2, 619 (2001).
[3] C. De Dominicis, *Phys. Rev. B*, **18**, 9, 4913 (1978).
[4] J. P. L. Hatchett, B. Wemmenhove, I. Pérez Castillo, T. Nikoletopoulos, N. S. Skantzos and A. C. C. Coolen, *J. Phys. A: Math. Gen.*, **37**, 6201, (2005).
[5] K. Mimura and A. C. C. Coolen, (*in preparation*).
[6] N. Sourlas, *Nature*, **339**, 693 (1989).
[7] Tanaka T and Okada M 2005 *IEEE Trans. Info. Theory* **51** 2 700.
[8] Y. Kabashima and D. Saad, *Europhys. Lett.*, **45**, 1, 97 (1999).
[9] R. Vicente, D. Saad and Y. Kabashima, *Phys. Rev. E*, **60**, 5, 5352 (1999).
[10] G. Semerjian and L. F. Cugliandolo, *Europhys. Lett.*, **61**, 2, 247 (2003).