



## STUDY OF BIFURCATION BEHAVIOR OF LDPC DECODERS

XIA ZHENG, FRANCIS C. M. LAU\*, CHI K. TSE and S. C. WONG

*Department of Electronic and Information Engineering,  
Hong Kong Polytechnic University, Hunghom, Hong Kong*

*\*encmlau@polyu.edu.hk*

Received October 11, 2005; Revised November 18, 2005

The use of low-density-parity-check (LDPC) codes in coding digital messages has aroused much research interest because of their excellent bit-error performance. The behavior of the iterative LDPC decoders of finite length, however, has not been fully evaluated under different signal-to-noise conditions. By considering the finite-length LDPC decoders as high-dimensional nonlinear dynamical systems, we attempt to investigate their dynamical behavior and bifurcation phenomena for a range of signal-to-noise ratios (SNRs). Extensive simulations have been performed on both regular and irregular LDPC codes. Moreover, we derive the Jacobian of the system and calculate the corresponding eigenvalues. Results show that bifurcations, including fold, flip and Neimark–Sacker bifurcations, are exhibited by the LDPC decoder. Results are useful for optimizing the choice of parameters that may enhance the effectiveness of the decoding algorithm and improve the convergence rates.

*Keywords:* Low-density-parity-check codes; iterative decoders; nonlinear dynamical systems; phase trajectories.

### 1. Introduction

Turbo codes [Berrou *et al.*, 1993; Agrawal & Vardy, 2001] and low-density-parity-check (LDPC) codes [Gallager, 1962; Richardson *et al.*, 2001] have been widely used in coding digital messages. Their ability to raise bit-error performance close to the Shannon limit has aroused much interest in the research community in the past decade. The LDPC codes were first proposed by Gallager [1962]. The idea was subsequently forgotten. Recently, with the rapid development of computational techniques as well as the popularization of personal computers, LDPC codes have been revisited and shown to outperform the popular turbo codes with lower error floor and shorter minimum distance between codewords.

LDPC codes can be decoded using various decoding methods, including hard-decision decoding and soft-decision decoding [Kou *et al.*, 2001;

Ryan, 2004; Shokrollahi, 2003]. Hard decision methods such as bit flipping and one-step majority-logic are simple for hardware implementation but are not good in error performance compared to soft-decision methods. In contrast, the optimal maximum likelihood decoder [Proakis, 1995] provides the lowest error rates among the various decoding algorithms, but it is too complex to be implemented in practice. A widely used algorithm is the *iterative decoding* based on belief propagation (BP) [Richardson *et al.*, 2001], which represents a good compromise between complexity and error performance.

In the BP algorithm, information is exchanged between two sets of intermediate nodes in the decoding graph, namely so-called variable nodes and check nodes, by passing messages along the edges connecting the nodes. The decoding algorithm works iteratively. The messages can be viewed as *a posteriori* probability values based on the

received information. The computation of *a posteriori* probabilities at each round requires that the incoming messages are statistically independent, and then the iterative equations correctly calculate the corresponding probabilities based on the observations. In fact, the independence assumption requires that the neighborhood of a node up to a certain depth is a tree, meaning that there is no cycle in the graph. Therefore, the BP algorithm assuming a cycle-free graph can achieve very good performance. As the block length tends to infinity, the behavior of the individual codes concentrates around its expected behavior of the ensemble and converges to the behavior of the cycle-free case. Many researchers have studied the iterative decoding of the infinite-length LDPC decoders and have proposed various kinds of methods to analyze the convergence behavior of the decoders [Richardson & Urbanke, 2001; Chung *et al.*, 2001; Franz *et al.*, 2002; Lehmann & Maggio, 2003]. As of today, the asymptotic behavior (as the block-length tends to infinity) of iterative decoding systems has been reasonably well understood.

One of the most popular analysis methods for studying LDPC decoders is *density evolution* (DE) [Richardson & Urbanke, 2001]. Given an initial probability density of the received signal, which depends on the channel parameters for a given channel, density evolution makes use of the initial density as the input and calculates the probability densities of the output messages iteratively according to the structure of the code ensemble. For the BP decoding algorithm, density evolution allows the algorithm to converge as the block length tends to infinity. For some classes of channels, there exists a maximum channel parameter  $\sigma^*$  (the threshold) such that the bit error rate (BER) tends to zero if  $\sigma < \sigma^*$  and converges to a nonzero fixed point if  $\sigma > \sigma^*$ . Note that bifurcation phenomena have been observed at the LDPC decoder as the SNR varies beyond the threshold.

Recently, many researchers have begun to pay attention to LDPC codes of finite length. Various construction methods such as the geometric approach [Kou *et al.*, 2001] and girth control algorithms [Mao & Banihashemi, 2001] have been proposed to produce good LDPC codes of finite length. Graph-based BP algorithms [Xiao & Banihashemi, 2004] have also been studied to improve the BER performance by removing loops during decoding. For easy hardware implementation, Chen and Fossorier [2002] have put forward

a series of simplified BP methods. While some researchers focus on the constructions of “good” codes and the design of low-complexity encoding/decoding techniques, others are interested in studying the behavior of the iterative decoding algorithms. Assuming a binary erasure channel, Di *et al.* [2002] have derived an analytical expression of the average bit error probability for regular LDPC ensembles with finite-length and have analyzed the convergence of the decoder as the channel parameter (block erasure probability) varies. The analysis is further generalized to include the study of irregular codes [Zhang & Orlitsky, 2002]. Unfortunately, both analyses are limited by the huge computational effort required, except for some simple codes. Under the same binary erasure channel, Richardson *et al.* [2002] and Amraoui *et al.* [2004] have investigated the behavior of the iterative decoding process. They have also found some approximated BER expressions for the average performance of an LDPC code ensemble of finite length. Much effort has been spent in studying the behavior of the LDPC decoder under a binary erasure channel, but the dynamical behavior of the decoder of finite length under a Gaussian noise channel has not been evaluated at all.

In their pioneering papers, Richardson [2000] and Agrawal and Vardy [2001] have studied in detail the performance and nature of the turbo decoding algorithm of finite length under a Gaussian noise channel by modeling the decoder as a discrete dynamical system. They have shown that in addition to fixed points, bifurcations leading to period doubling and oscillations may be produced by the decoding algorithm. Later, Kocarev *et al.* [2002] discovered that chaos exists in turbo decoding and a control method has been proposed to improve the convergence rate under such a scenario. Since both turbo decoders and LDPC decodes are iterative systems, it is highly probable that the decoding processes are rich in nonlinear phenomena.

In this paper, we attempt to study in depth the behavior of finite-length LDPC decoders. Assuming a Gaussian noise channel, we analyze the phase trajectories of *a posteriori* probabilities as the iterative process progresses. Extensive simulations have been performed for a wide range of SNRs. In Sec. 2, we will briefly review the representations of the LDPC codes and the iterative decoding process. The dynamical equations governing the decoding process are also given. When a fixed point exists, the corresponding Jacobian matrix is also derived. In

Sec. 3, we will present some results for both regular and irregular LDPC codes. Finally, we give our conclusions in Sec. 4.

## 2. Overview of LDPC Coding Systems

### 2.1. LDPC codes

#### 2.1.1. Graph representation

Low-density-parity-check codes are linear block codes which can be represented by bipartite graphs consisting of two sets of nodes, namely *variable nodes* and *check nodes*. The variable nodes correspond to the elements of the codeword and the check nodes correspond to the sets of parity-check constraints satisfied by the codewords of the code. The connections between the two different types of nodes are called *edges*. The number of edges emanated from a node is referred to as the *degree* of the node. The key property of LDPC codes is the sparsity of the graph. In other words, the degree of every node is low. Also, there are two kinds of LDPC codes: regular and irregular.

For regular LDPC codes, all nodes of the same type have the same degree. The degrees of the variable nodes and check nodes are usually called column weight and row weight, denoted by  $w_c$  and  $w_r$ , respectively. For irregular LDPC codes, the degree of each set of nodes is chosen according to some distributions. For a given distribution pair  $(\lambda, \rho)$  of an LDPC ensemble,

$$\lambda(x) := \sum_{i=2}^{d_v} \lambda_i x^{i-1} \left( \rho(x) := \sum_{j=2}^{d_c} \rho_j x^{j-1} \right) \quad (1)$$

specifies the variable (check) node degree distribution, where  $\lambda_i$  denotes the fraction of all edges connected to degree- $i$  variable nodes and  $\rho_j$  denotes the fraction of all edges connected to degree- $j$  check nodes. Moreover,  $d_v$  and  $d_c$  denote the maximum variable-node degree and maximum check-node degree, respectively. Note that regular code is just a special case of the irregular ones. For regular codes with column weight  $w_c$  and row weight  $w_r$ , all variable nodes are connected with  $w_c$  edges, i.e.

$$\lambda_i = \begin{cases} 1 & i = w_c \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

and consequently, we have  $\lambda(x) = x^{w_c-1}$ . For the same reason, we have  $\rho(x) = x^{w_r-1}$ .

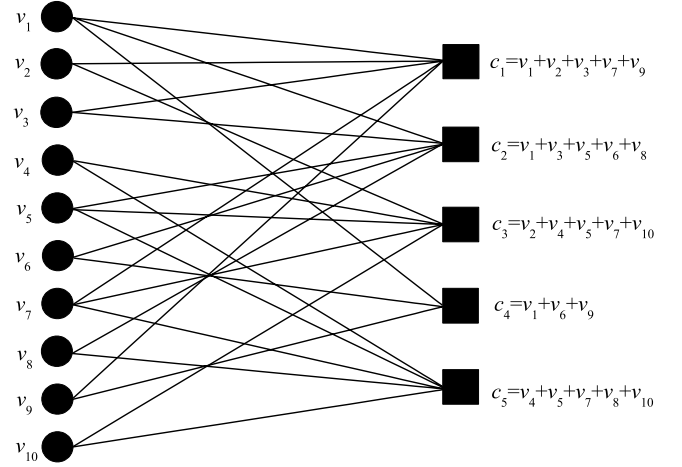


Fig. 1. A graph representation of (10, 5) irregular LDPC code.

In Fig. 1, an example of (10, 5) irregular LDPC code is shown. The (10, 5) code indicates that there are ten variable nodes (as shown on the left-hand side) and five check nodes (as shown on the right-hand side) in the bipartite graph. Each check node represents a check equation satisfied by the codeword. Referring to the figure, it can be observed that three edges emanate from  $v_1$  and hence the degree of node  $v_1$  equals 3.

For a given block length and a given degree distribution, we define an ensemble of codes by choosing the edges randomly. By definition, all elements from the same ensemble of infinite length have the same performance. But in practice, for finite-length codes, we need to choose the edges carefully to avoid the small loops, which will degrade the error performance of a code.

#### 2.1.2. Matrix representation

Consider a linear block code with  $k$  information bits and  $(n - k)$  check bits. The set of codewords  $\mathbf{C}$  can be described as a  $k$ -dimensional vector subspace of the space of all  $n$ -tuples over  $F_2^n$ , where  $F_2^n$  represents the binary field with  $n$  dimensions. In other words, the set of all  $n$ -bit vectors in  $\mathbf{C}$  are formed by linear combinations of  $k$  linearly independent basis vectors  $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k\}$  over  $F_2^n$ , and the basis vectors can be arranged as rows of a  $k \times n$  generation matrix  $\mathbf{G}$  such that  $\mathbf{G} = \{\mathbf{g}_1^T \ \mathbf{g}_2^T \ \dots \ \mathbf{g}_k^T\}^T$ . The null space of  $\mathbf{G}$  is associated with a matrix  $\mathbf{H}$ , which is called the parity-check matrix. The  $(i, j)$ th element of the parity-check matrix  $\mathbf{H}$ , denoted by  $h_{ij}$ , is 1 if and only if the  $i$ th check node is connected to the  $j$ th variable node. As a consequence,

the LDPC code can be defined as the set of vectors  $\mathbf{C}$  such that all elements  $\mathbf{c} \in \mathbf{C}$  satisfy  $\mathbf{c}\mathbf{H}^T = \mathbf{0}$ . In the bipartite graph, such as the one shown in Fig. 1, the edges represent the connections between the variable nodes and the check nodes. Such connections are also indicated by the corresponding parity-check equations. Therefore, based on the bipartite graph, the corresponding matrix representation of the LDPC code can be derived. For example, it is readily shown that the parity-check matrix of the (10, 5) code in Fig. 1 is given by

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \quad (3)$$

### 2.1.3. Decoding algorithm

Without loss of generality, we consider a binary-input additive white Gaussian noise channel and a transmitted codeword with a block length  $n$ . We denote the  $i$ th code bit ( $i = 1, 2, \dots, n$ ) by  $c_i \in \{0, 1\}$ . The transmitted signal corresponding to this code bit is denoted by  $s_i \in \{-1, 1\}$  and equals  $(-1)^{c_i}$ . The received signal, denoted by  $y_i$ , is given by  $y_i = s_i + z_i$ , where the variables  $z_i$  are independent and identically distributed zero-mean Gaussian random variables with variance (noise power)  $\sigma^2$ .

We assume that the received signal is  $y_i$ , and the message  $r_{j'i}(b)$  ( $b = 0, 1$ ) is passed from the neighboring check-node set  $C_i$  excluding the check node  $j$ . We define  $q_{ij}(b)$  as the conditional posterior probability of the bit  $c_i$  being equal to  $b$ . We further assume that the message  $q_{i'j}(b)$  is passed from the neighboring variable-node set  $V_j$  excluding the variable node  $i$ . We may define  $r_{ji}(b)$  as the conditional posterior probability of the event that the  $j$ th check equation is satisfied. Assuming that the passed messages in the iterative process are independent random variables, the BP algorithm in the probability domain proceeds as follows.

1. Estimate the noise power  $\sigma^2$ . For  $i = 1, 2, \dots, n$ , initialize  $P_i(b) := \Pr(c_i = b|y_i)$ , where  $\Pr(c_i = b|y_i)$  denotes *a posteriori* probability that bit  $c_i$  equals  $b$  given the received signal  $y_i$ . Set  $q_{ij}(b) = P_i(b)$  if the variable node  $i$  and the check node  $j$  are connected.

2. Update  $\{r_{ji}(b) : i, j = 1, 2, \dots, n; b = 0, 1\}$ , using

$$\begin{cases} r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in V_j/i} (1 - 2q_{i'j}(1)) \\ r_{ji}(1) = 1 - r_{ji}(0). \end{cases} \quad (4)$$

3. Update  $\{q_{ij}(b) : i, j = 1, 2, \dots, n; b = 0, 1\}$ , using

$$q_{ij}(b) = K_{ij} P_i(b) \prod_{j' \in C_i/j} r_{j'i}(b) \quad (5)$$

where  $K_{ij}$  is chosen to ensure that  $q_{ij}(0) + q_{ij}(1) = 1$ .

4. Compute *a posteriori* probability of bit  $c_i$  using

$$Q_i(b) = K_i P_i(b) \prod_{j \in C_i} r_{ji}(b) \quad (6)$$

where  $K_i$  is chosen to ensure that  $Q_i(0) + Q_i(1) = 1$ .

5. For  $i = 1, 2, \dots, n$ , set

$$Q_i(1) \stackrel{\hat{c}_i=1}{\underset{\hat{c}_i=0}{\geq}} Q_i(0). \quad (7)$$

If  $\hat{\mathbf{c}}\mathbf{H}^T = \mathbf{0}$  or the number of iterations equals the maximum limit, stop; else, go to Step 2.

## 2.2. Nonlinear dynamics of the finite-length LDPC decoder

For infinite-length LDPC codes, careful construction according to the density evolution produces very good LDPC codes with no loops in the graph, which means that the messages passed are independent of one another. Then, the decoders will always converge either to the indecisive fixed points or the unequivocal fixed point. However, finite-length LDPC codes with good minimum distance should contain some loops. Hence, in the iterative decoding process, the message may be trapped in the loops and the output may oscillate.

To analyze the convergence behavior of the finite-length LDPC decoder, we can rewrite the whole iterative process as

$$\begin{cases} \mathbf{q}^l(0, \sigma) = f_1(\mathbf{r}^l(0, \sigma)) \\ \mathbf{r}^{l+1}(0, \sigma) = f_2(\mathbf{q}^l(0, \sigma)) \end{cases} \quad (8)$$

where both  $\mathbf{r}^l(0, \sigma)$  and  $\mathbf{q}^l(0, \sigma)$  are vectors parameterized by  $\sigma$  and of length  $M = n/\int_0^1 \lambda(x)dx$ .



(Recall that  $\lambda(x)$  specifies the variable node degree distribution.)

It can be observed that the whole process is parameterized by the probabilities  $\Pr(c_i = 0|y_i)$ ,  $i = 1, 2, \dots, n$ , which are determined by the transmitted codeword and the noise values. Simulations have also shown that the iterative decoder is very sensitive to such parameters.

Suppose that a fixed point exists in the dynamical system. Linearizing (8) around the fixed point, we get

$$\begin{aligned} \begin{cases} \mathbf{q}^l(0, \sigma^*) = \mathbf{J}_1 \mathbf{r}^l(0, \sigma^*) \\ \mathbf{r}^{l+1}(0, \sigma^*) = \mathbf{J}_2 \mathbf{q}^l(0, \sigma^*) \end{cases} \\ \Rightarrow \mathbf{r}^{l+1}(0, \sigma^*) = \mathbf{J}_2 \mathbf{J}_1 \mathbf{r}^l(0, \sigma^*)|_{\mathbf{r}^{l+1}=\mathbf{r}^l} \\ = \mathbf{J} \mathbf{r}^l(0, \sigma^*)|_{\mathbf{r}^{l+1}=\mathbf{r}^l} \end{aligned} \quad (9)$$

where  $\mathbf{J}_1$  and  $\mathbf{J}_2$  are the Jacobian matrices of the functions  $f_1$  and  $f_2$ , respectively,  $\mathbf{J} = \mathbf{J}_2 \mathbf{J}_1$ , and  $\sigma^*$  is the parameter at the fixed point. For specific variable nodes  $i$  and  $i_1$  and check nodes  $j$  and  $j_1$ , the  $(\phi(i, j), \phi(i_1, j_1))$ th element of the matrix  $\mathbf{J}$  can be shown equal to

$$\begin{aligned} & J_{\phi(i, j)\phi(i_1, j_1)} \\ &= \frac{(2r_{j_i}^{l+1}(0) - 1) \cdot P_{i_1} \cdot \prod_{j' \in V_{i_1}/j} R_{j' i_1}^l}{\left[1 - \left(P_{i_1} \cdot \prod_{j' \in V_{i_1}/j} R_{j' i_1}^l\right)^2\right]} \cdot r_{j_1 i_1}^l(0) \cdot r_{j_1 i_1}^l(1) \\ & \quad \cdot \pi(i, j, i_1, j_1) \end{aligned} \quad (10)$$

where  $P_i = P_i(1)/P_i(0)$  and  $R_{j_i}^l = r_{j_i}^l(1)/r_{j_i}^l(0)$ .  $\phi(i, j)$  is an index function defined as  $\phi(i, j) = \sum_{i'=1}^{i-1} \sum_{j'=1}^m h_{j' i'} + \sum_{j'=1}^j h_{j' i}$ , and the output range of which is from 1 to  $M$ . Also,  $m$  is defined as the check length of the code, i.e. the number of check equations. For the function  $\pi(i, j, i_1, j_1)$ , its value equals 1 if the variable nodes  $i$  and  $i_1$  are both connected to the check node  $j$ , with check node  $j_1$  connected to variable node  $i_1$ ; otherwise, it equals 0.

The stability of the fixed point can then be determined from the eigenvalues of the Jacobian of the iterative system evaluated at the fixed point. Perturbations grow exponentially if one of the absolute values of the eigenvalues is larger than 1 and decay if all the eigenvalues lie in the unit circle. A fixed point is said to be stable if all sufficiently small disturbances remote from it damp out in time. Conversely, unstable equilibria, in which disturbances grow in time, are represented by unstable fixed points. If an eigenvalue approaches  $-1$

and 1, respectively, flip bifurcation and fold bifurcation would occur. Also, Neimark–Sacker bifurcation occurs when a pair of complex conjugate eigenvalues move towards the unit circle from inside [Strogatz, 1994].

Although we can find the entire phase trajectories of  $\mathbf{r}^l(0, \sigma)$  and  $\mathbf{q}^l(0, \sigma)$ , it is impractical to plot and study them all because both  $\mathbf{r}^l(0, \sigma)$  and  $\mathbf{q}^l(0, \sigma)$  are very high dimensional variables in the order of tens of thousands. Instead, we make use of the measure  $E(l)$  to investigate the dynamical behavior of the decoder, and  $E(l)$  is defined as the mean-square value of *a posteriori* probabilities of the code bits being equal to 0 at the  $l$ th iteration, i.e.

$$E(l) := \frac{1}{n} \sum_{i=1}^n [Q_i^l(0)]^2. \quad (11)$$

In our study, codewords with all zeros are used because it is known that the all-zero codeword is adequate for assessing the performance of a linear code with a symmetrical channel and a symmetrical decoding algorithm. Therefore, if all code bits are detected correctly after some iteration number,  $Q_i^l(0) = 1$  for all  $i$  and consequently  $E(l) = 1$ .

### 3. Bifurcation Phenomena by Simulations

Suppose the noise samples are represented by  $\mathbf{z} = (z_1, z_2, \dots, z_n)$ . If the ratios between consecutive sample values, i.e.  $(z_1/z_2, z_2/z_3, \dots, z_{n-1}/z_n)$ , are fixed, we refer to such noise samples as one noise realization. Different noise realizations correspond to different noise-ratios vectors. For a given noise realization, the noise vector  $\mathbf{z}$  completely determines the SNR because  $1/(2R \times \text{SNR}) = \hat{\sigma}^2 = (1/n) \sum_{i=1}^n z_i^2$  where  $R$  is the code rate. Usually,  $n$  is a large integer and  $\hat{\sigma}^2$  will be a good approximation of the channel noise power  $\sigma^2$ .

Extensive simulations have been performed to identify the relevant dynamics. In particular, it is found that fold bifurcation, flip bifurcation and Neimark–Sacker bifurcation occur within a certain range of the SNR called the “waterfall region”. For regions with high and low SNRs, two kinds of fixed points in the decoding system are observed, namely the unequivocal one and the indecisive one. Note that the unequivocal fixed point is the desired fixed point which produces a definite decoding outcome. In Fig. 2, we plot the histograms of *a posteriori* probability (based on (6)) at the fixed points

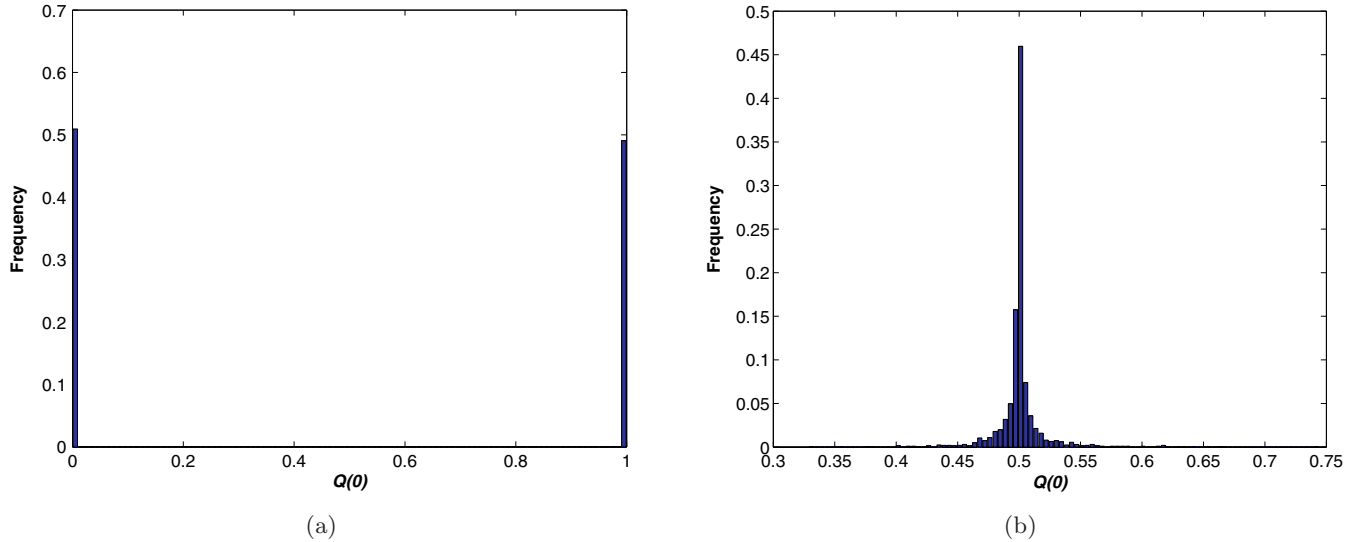


Fig. 2. Histogram of posterior probability corresponding to (a) an unequivocal fixed point, and (b) an indecisive fixed point.

for an arbitrary LDPC code. For an unequivocal fixed point, all the probability values converge to either 1 or 0, which is unequivocal for hard decision [see Fig. 2(a)]. Conversely, we refer to a fixed point as an indecisive fixed point when the LDPC decoding algorithm is relatively ambiguous regarding the values of the information bits, with posterior probability values heavily clustered around 0.5 [see Fig. 2(b)]. It is also interesting to note that the algorithm converges to the unequivocal fixed point if, and only if, the decoder finds a valid codeword.

### 3.1. Irregular LDPC codes

We first consider an irregular (1008, 504) LDPC code [MacKay, 2005] and study the trajectories of the iterative decoding algorithm.

#### 3.1.1. Fold bifurcation

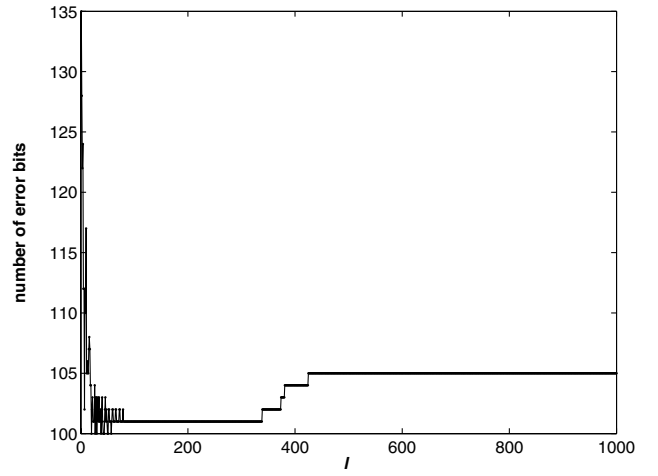
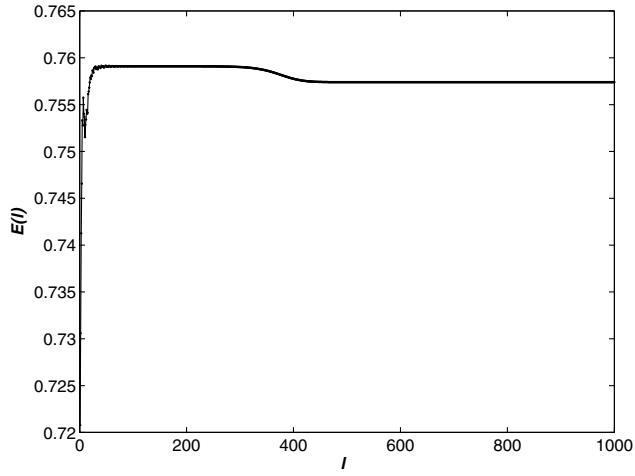
For some noise realizations, only fold bifurcations occur as we vary the SNR. Figure 3 shows the changes in the phase trajectories induced by a typical fold bifurcation for a particular noise realization. The figures on the left-hand side plot the value of  $E(l)$  against  $l$ , whereas those on the right plot the number of error bits against  $l$ . At an SNR value of 0.911785184 dB, the phase trajectory of the LDPC decoder converges to a stable indecisive fixed point, as indicated by the plots in [Fig. 3(a)]. As the SNR is increased to 0.911785185 dB, the indecisive fixed point disappears and the phase trajectory is able to move away from this neighborhood and converges to

an unequivocal fixed point. But there is a long time transient behavior of about 400 iterations before convergence finally takes place, as seen in Fig. 3(b). When the SNR is further increased to 0.92 dB, it takes less than 60 iterations for the trajectory to converge. In Fig. 4, the values of  $E(l)$  at the steady state are plotted against SNR. It can be seen that in the low SNR region, the iterative algorithm converges to an indecisive fixed point, whereas in the high-SNR region, the algorithm converges to an unequivocal fixed point. Also, the two SNR regions are separated by a fold bifurcation.

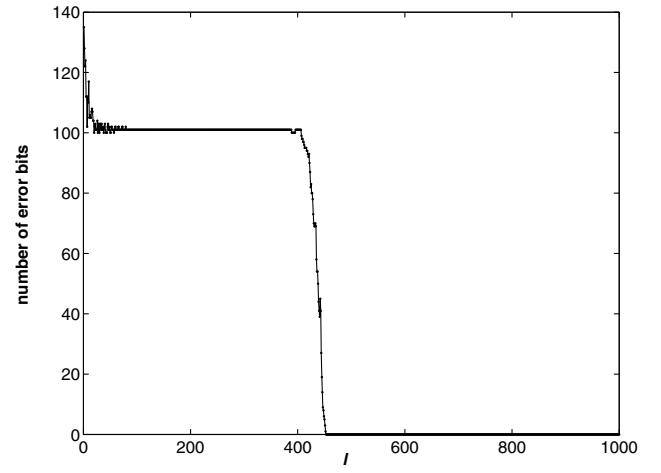
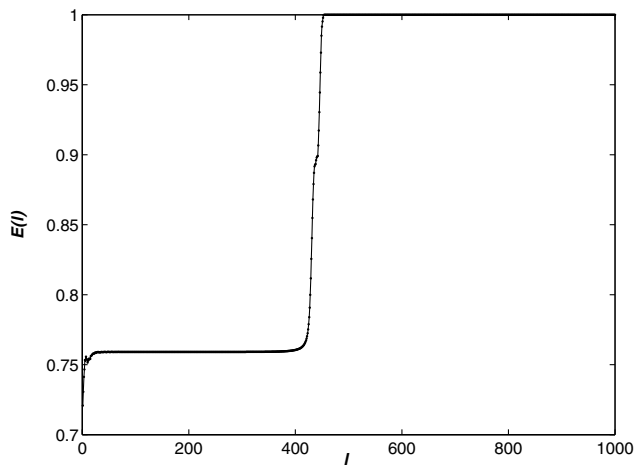
#### 3.1.2. Flip bifurcation and Neimark–Sacker bifurcation

For some other noise realizations, both flip bifurcations and Neimark–Sacker bifurcations occur. Figure 5 illustrates such bifurcations occurring at the LDPC decoder for another noise realization.

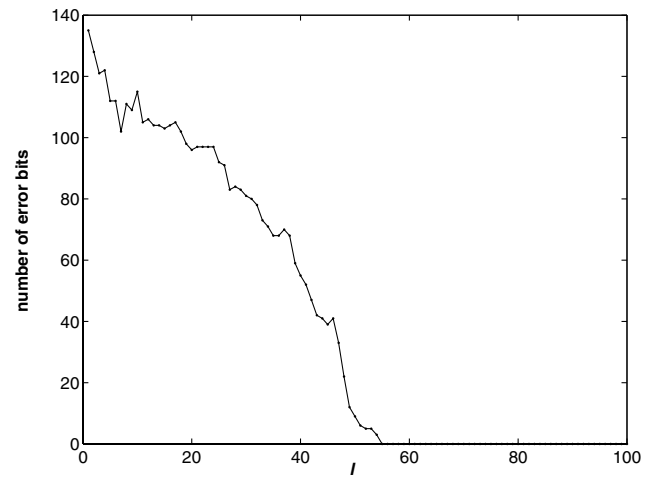
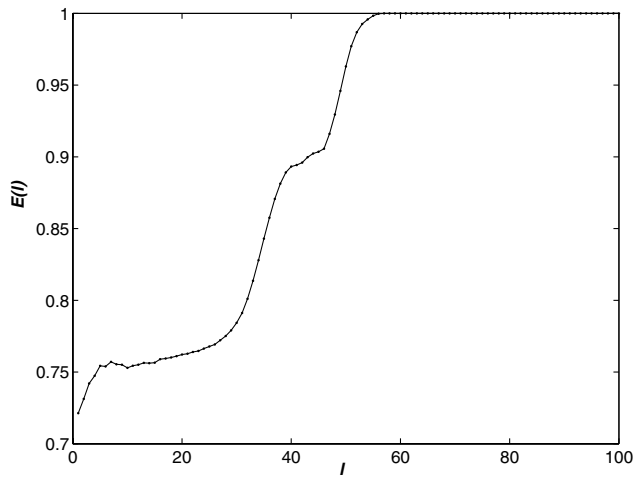
At an SNR value of 0.30 dB, the LDPC decoding algorithm converges to a stable indecisive fixed point [see Fig. 5(a)]. As the SNR increases to about 0.44 dB, flip bifurcation occurs. Figure 5(b) shows a stable period-two cycle at the steady state at SNR = 0.55 dB. The periodic points lose their stability and eventually bifurcate at around SNR = 0.595 dB. Figure 5(c) shows that the trajectory converges to an indecisive fixed point at SNR = 0.601 dB. When the SNR is further increased to 0.615 dB, the phase trajectory converges to the steady state much slowly, as can be seen in Fig. 5(d).



(a)



(b)



(c)

Fig. 3. A typical fold bifurcation. (Left)  $E(l)$  versus  $l$ . (Right) Number of error bits versus  $l$ . (a) SNR = 0.911785184 dB; (b) SNR = 0.911785185 dB; (c) SNR = 0.92 dB.

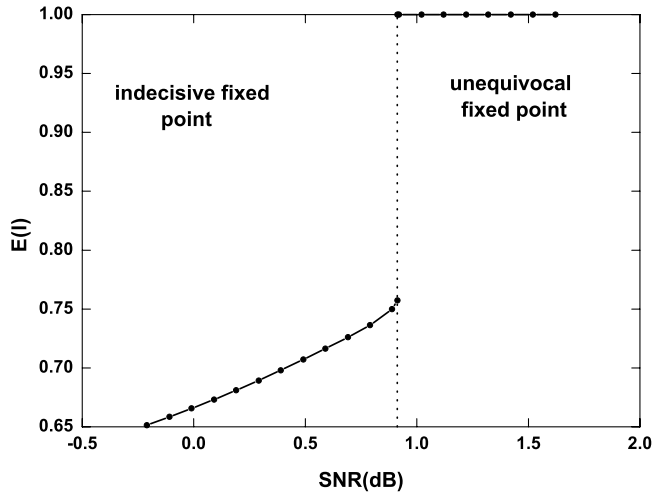


Fig. 4. Bifurcation diagram of  $E(l)$  for a particular noise realization. Dotted line corresponds to the SNR at which fold bifurcation occurs.

At around  $\text{SNR} = 0.62$  dB, the fixed point undergoes a Neimark–Sacker bifurcation and the phase trajectory goes into an invariant set. As a result, after a transient period, the phase trajectory converges to a quasi-periodic orbit. Figure 5(e) depicts the quasi-periodic orbit at the steady state at  $\text{SNR} = 0.65$  dB. As the SNR increases, the trajectory finally loses its stability and chaos emerges at  $\text{SNR} = 0.85$  dB. The chaotic trajectory at  $\text{SNR} = 0.94$  dB is shown in Fig. 5(f). Finally, when the SNR is large enough, the LDPC decoding algorithm is able to find an unequivocal fixed point after

a number of iterations. A trajectory corresponding to  $\text{SNR} = 1.004$  dB is shown in Fig. 5(g).

In Fig. 6, the values of  $E(l)$  at the steady state are plotted against SNR. It can be observed that bifurcations occur in the SNR range of around 0.45 dB to 1 dB. In the SNR range of 0.45 dB to 0.595 dB, the oscillations can hardly be displayed in the figure because the amplitude of the oscillation, typically in the order of  $10^{-4}$  as shown in Fig. 5(b), is too small.

Based on the values of  $E(l)$  as the iteration progresses, we compute the Lyapunov exponent. In Fig. 7, we plot the Lyapunov exponent value against SNR. It is found that the Lyapunov exponent value approaches zero at  $\text{SNR} = 0.45$  dB and at  $\text{SNR} = 0.62$  dB, where flip bifurcation and Neimark–Sacker bifurcation occur, respectively. At  $\text{SNR} = 0.86$  dB, the exponent turns positive, corresponding to the beginning of the chaotic region in Fig. 6. As the SNR is increased to around 1 dB, the Lyapunov exponent rapidly drops from a positive value to less than  $-30$  (not shown in the figure due to its very small value), indicating the super-stability of the unequivocal fixed point. Comparing Figs. 3 and 5, we also find that although different noise realizations produce different bifurcation diagrams, the whole SNR range can be roughly divided into three regions: (i) low-SNR region corresponding to indecisive fixed points; (ii) waterfall region where bifurcations occur; and (iii) high-SNR region corresponding to unequivocal fixed points.

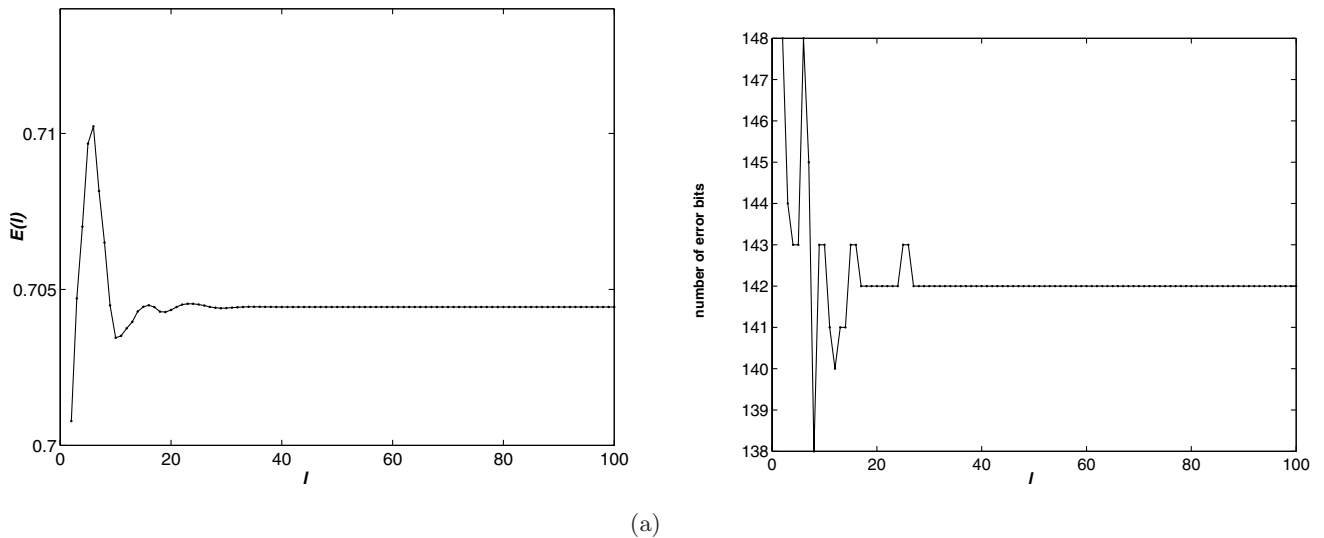
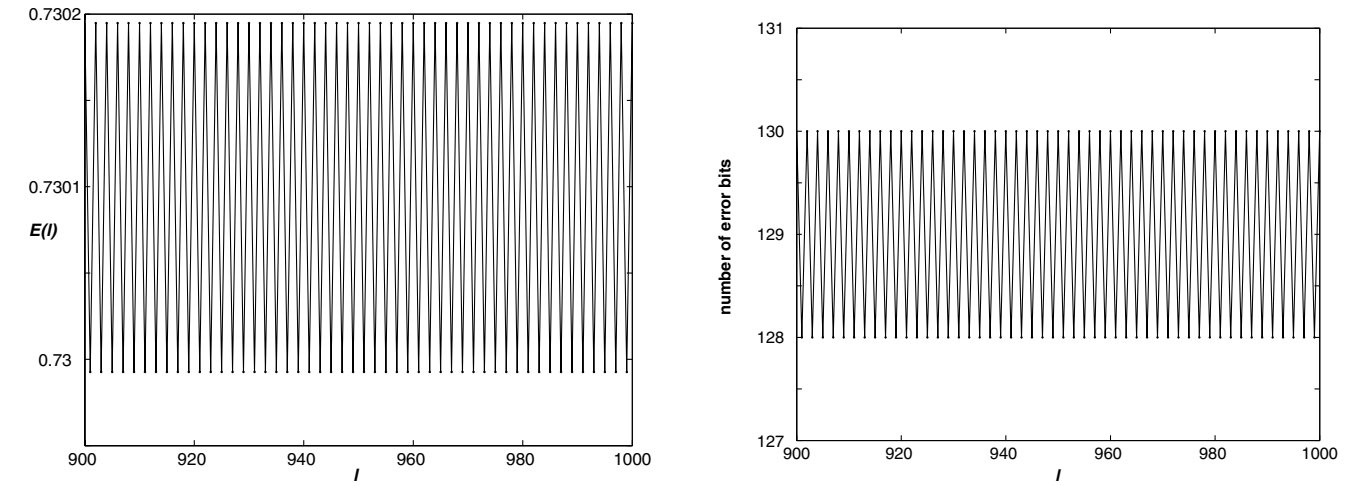
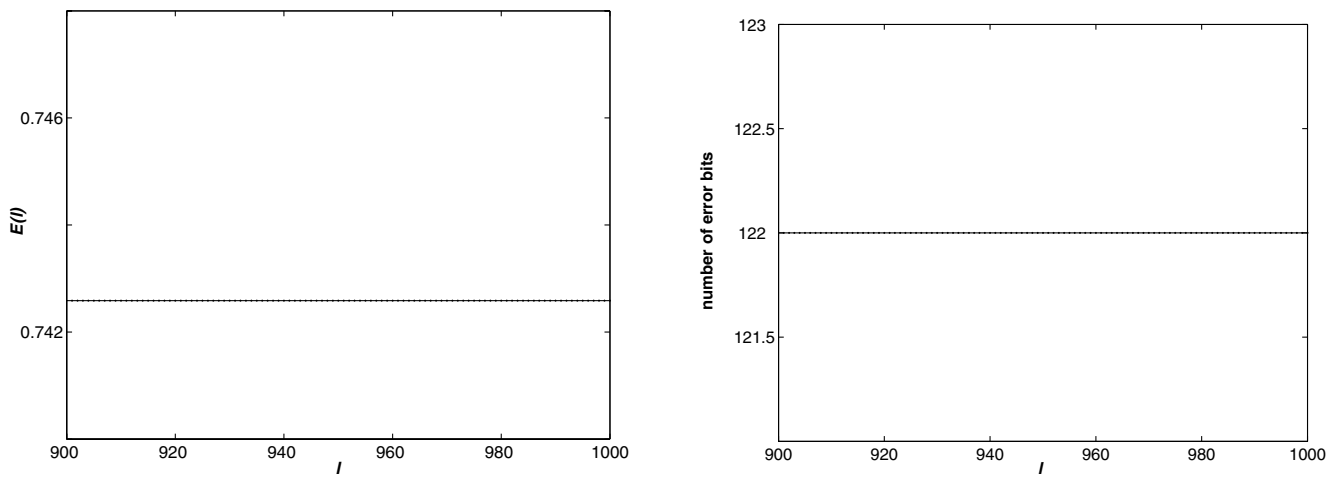


Fig. 5. Typical flip bifurcation and Neimark–Sacker bifurcation. (Left)  $E(l)$  versus  $l$ . (Right) Number of error bits versus  $l$ . (a)  $\text{SNR} = 0.30$  dB; (b)  $\text{SNR} = 0.55$  dB; (c)  $\text{SNR} = 0.601$  dB; (d)  $\text{SNR} = 0.615$  dB; (e)  $\text{SNR} = 0.65$  dB; (f)  $\text{SNR} = 0.94$  dB; (g)  $\text{SNR} = 1.004$  dB.

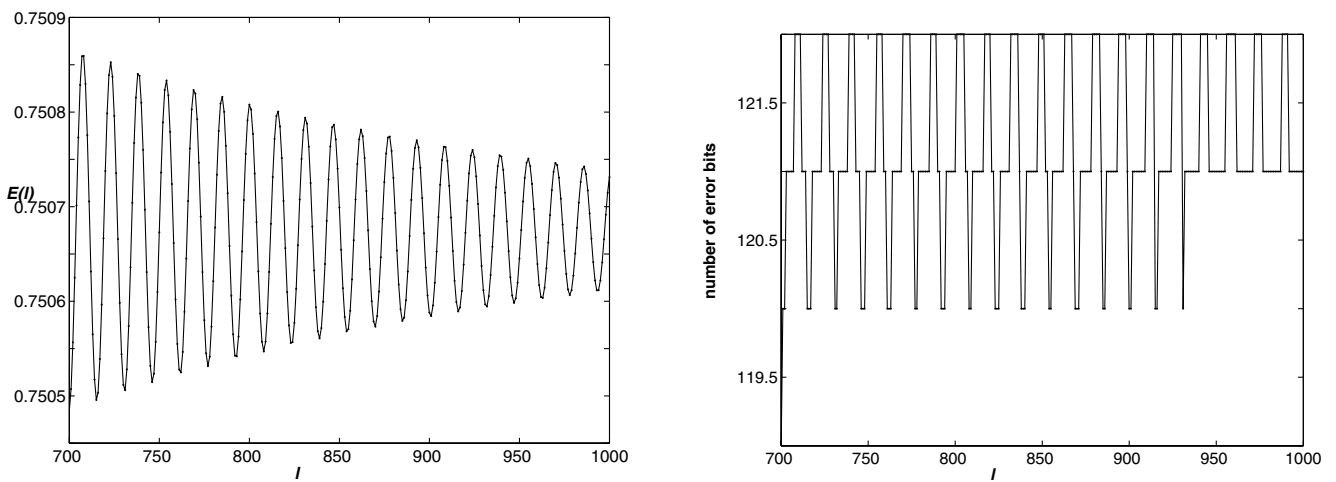




(b)

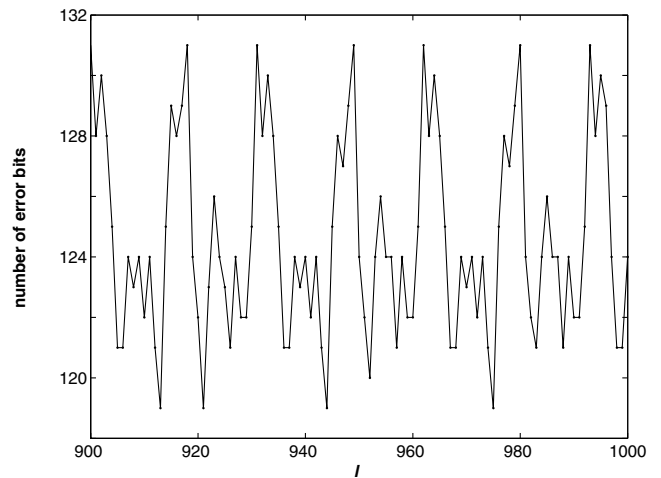
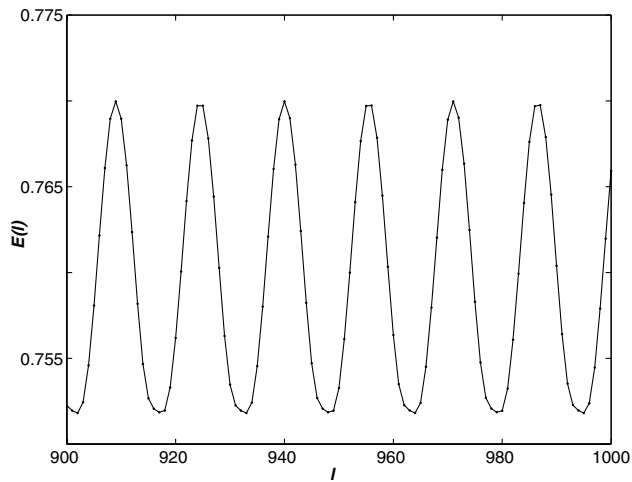


(c)

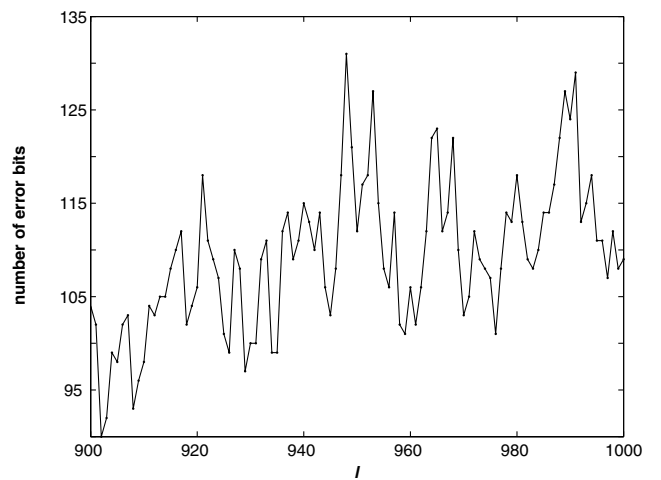
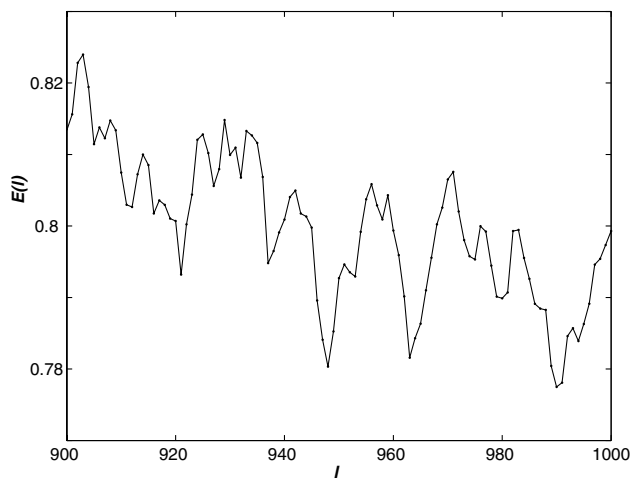


(d)

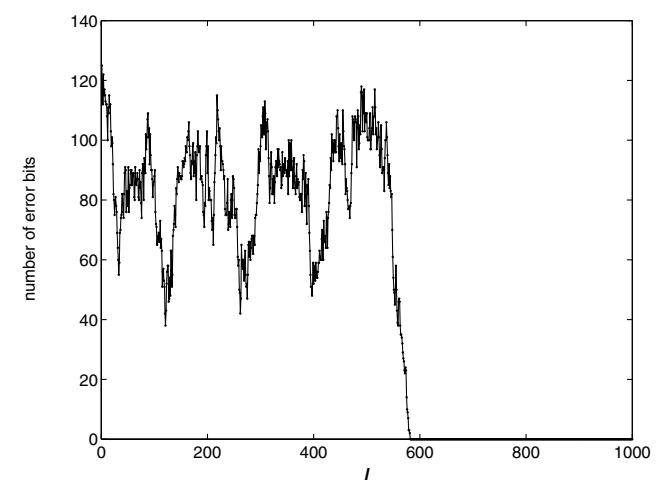
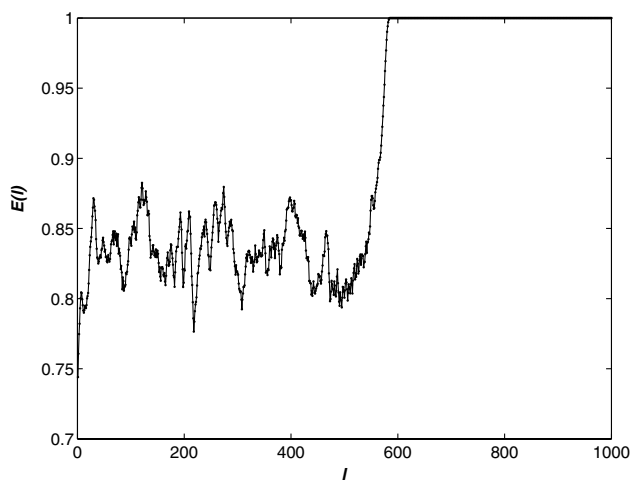
Fig. 5. (Continued)



(e)



(f)



(g)

Fig. 5. (Continued)

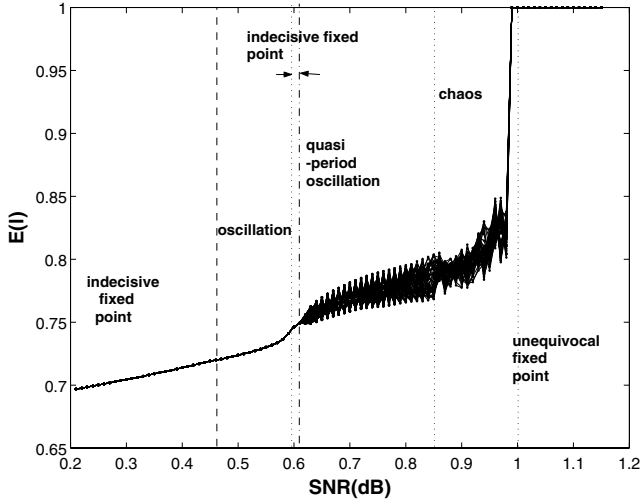


Fig. 6. Bifurcation diagram of  $E(l)$  for a particular noise realization. (Dash line) Flip bifurcation occurs; (dash-dotted line) Neimark–Sacker bifurcation occurs; (dotted line) other types of bifurcation occur.

### 3.2. Regular LDPC codes

In this section, we present some results for the regular LDPC codes. From our extensive simulation results, we observe that decoders for regular LDPC codes have similar behavior as those for irregular codes. Here, we choose the regular LDPC code (504,252) with variable degree being 3 and check

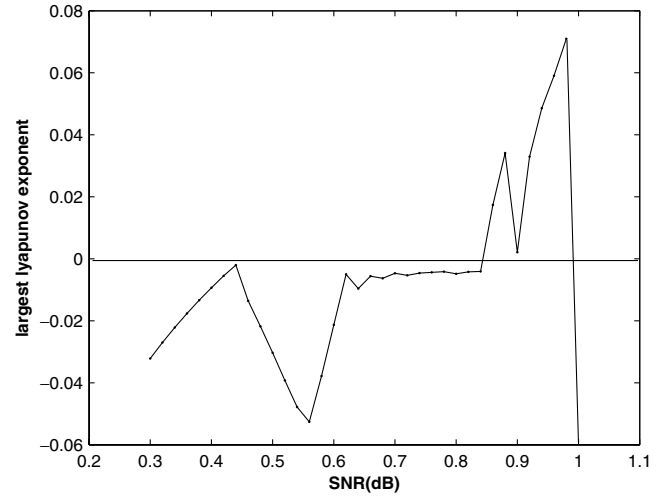
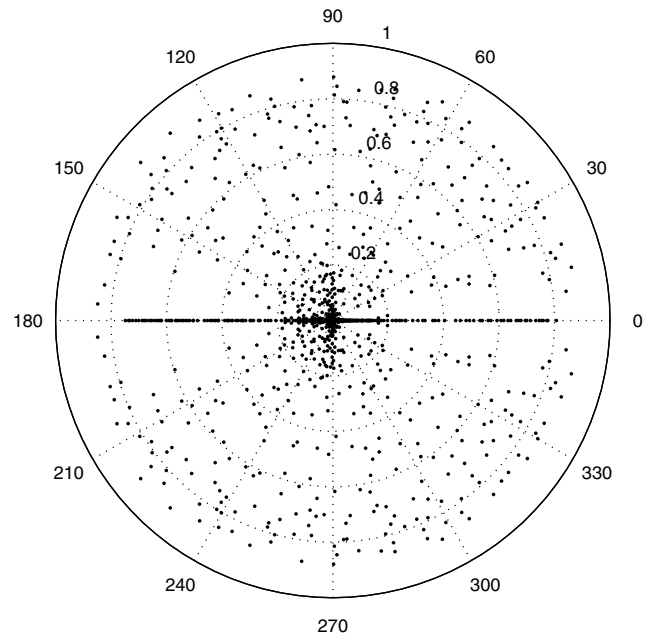
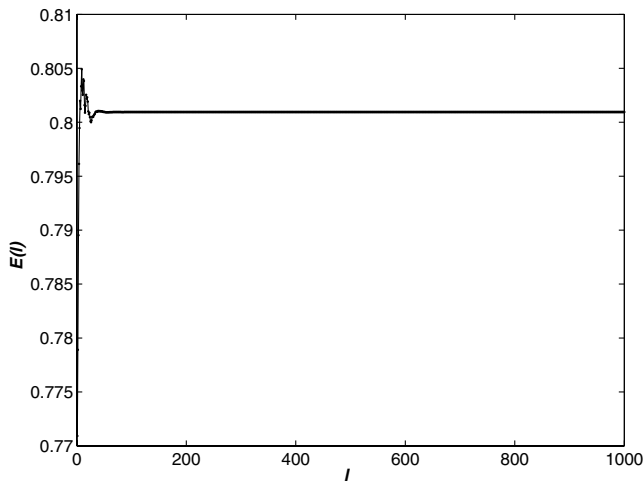


Fig. 7. A plot of the Lyapunov exponent value against SNR with a particular noise realization.

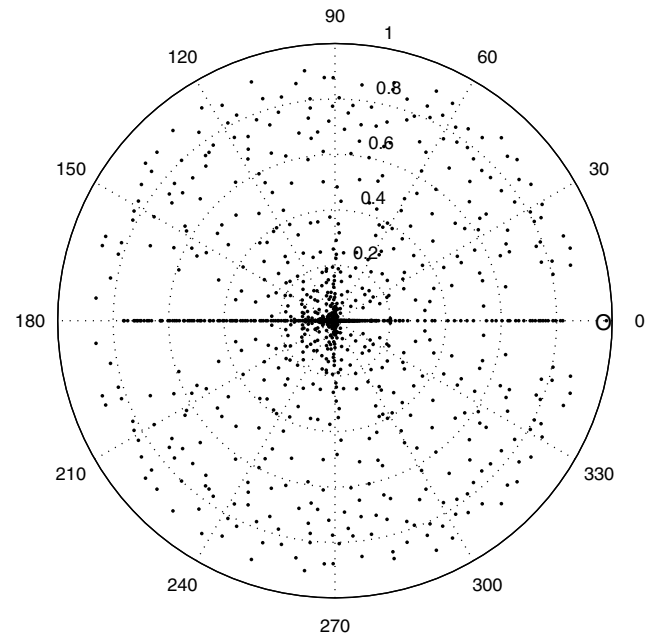
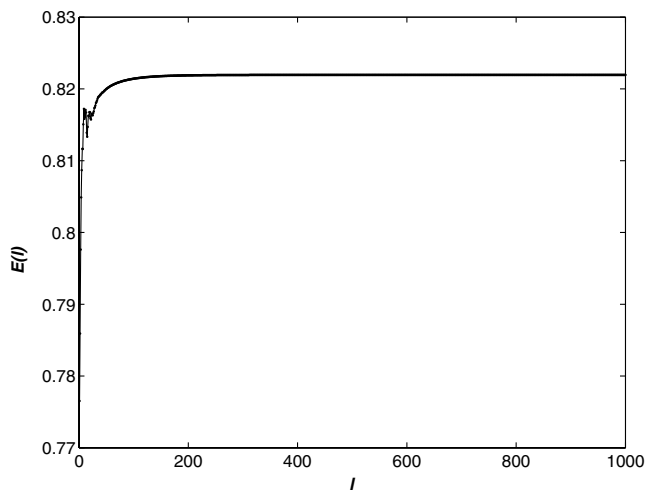
degree being 6. To describe the characteristics of bifurcations more clearly, we plot  $E(l)$  against  $l$  and also the eigenvalues of the Jacobian for the corresponding fixed points.

In Fig. 8, at SNR = 1.400 dB, all the eigenvalues for the indecisive fixed points fall within the unit circle, indicating the stability of the fixed points.

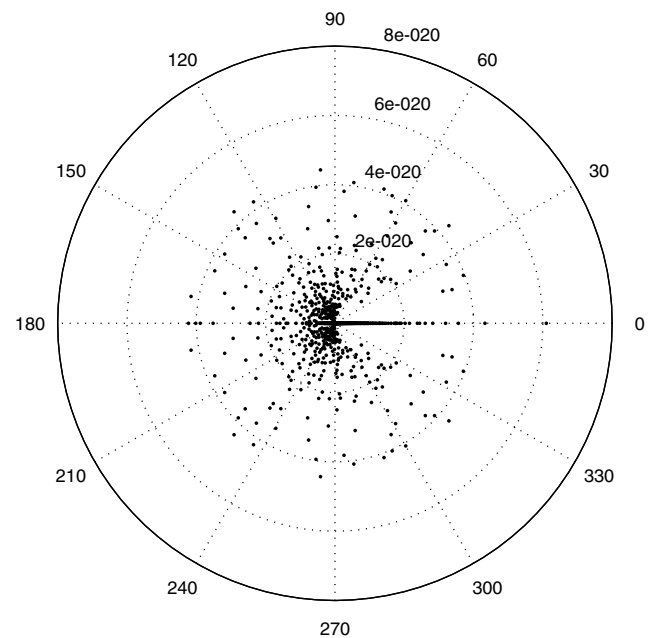
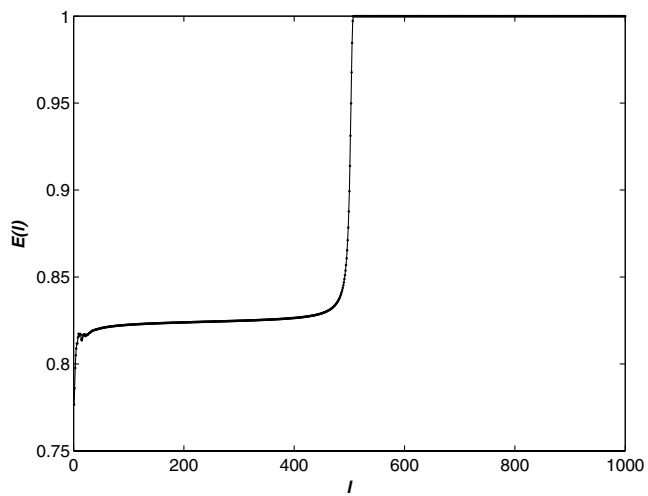


(a)

Fig. 8. A typical fold bifurcation. (Left)  $E(l)$  versus  $l$ . (Right) Polar plots of eigenvalues. (a) SNR = 1.400 dB; (b) SNR = 1.478 dB; (c) SNR = 1.480 dB.



(b)



(c)

Fig. 8. (Continued)

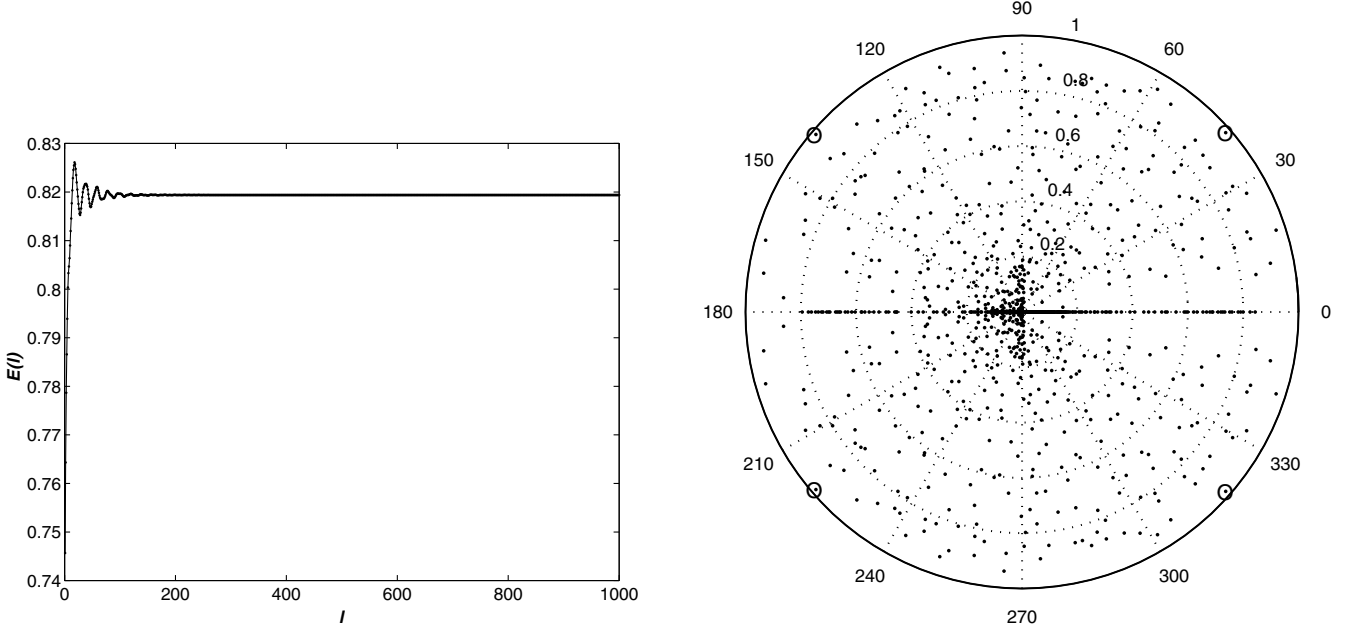


Fig. 9. Phase trajectory of  $E(l)$  before Neimark-Sacker bifurcation. (Left)  $E(l)$  versus  $l$ . (Right) Polar plots of eigenvalues. SNR = 0.80 dB. (Note that two pairs of complex eigenvalues approach the unit circle.)

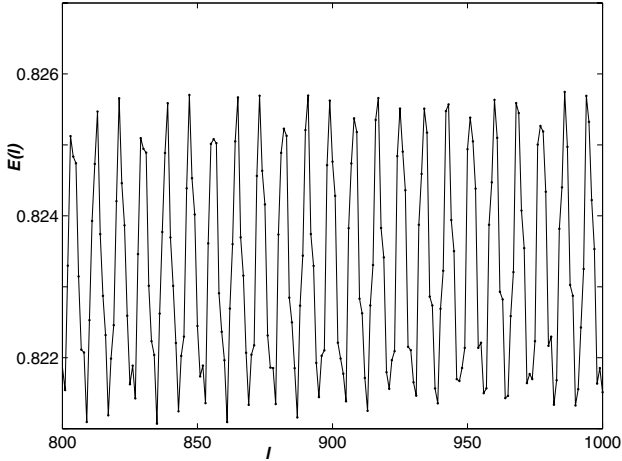


Fig. 10. Phase trajectory of  $E(l)$  after Neimark-Sacker bifurcation. SNR = 0.85 dB.

As SNR increases to 1.478 dB, one of the eigenvalues approaches 1, implying that fold bifurcation occurs. At SNR = 1.480 dB, the indecisive fixed point disappears, and an unequivocal fixed point appears.

To analyze the stability of the unequivocal fixed point, we derive the Jacobian as follows. As we know, when the algorithm converges to an unequivocal fixed point, all the messages passing between the variable nodes and check nodes converge to either 1 or 0. In our simulations, only the all-zero codes are transmitted, and hence the messages  $\mathbf{r}^l(0, \sigma)$  and  $\mathbf{r}^l(1, \sigma)$  converge to the all-one vector and all-zero vector, respectively. So, the  $(\phi(i, j), \phi(i_1, j_1))$ th element of the Jacobian  $\mathbf{J}$  where  $\pi(i, i, i_1, j_1) = 1$  can be written as

$$\begin{aligned}
 J_{\phi(i,j)\phi(i_1,j_1)} &= \lim_{\substack{\mathbf{r}^l(0,\sigma) \rightarrow \mathbf{1} \\ \mathbf{r}^l(1,\sigma) \rightarrow \mathbf{0}}} \frac{(2r_{ji}^{l+1}(0) - 1) \cdot P_{i_1} \cdot \prod_{j' \in V_{i_1}/j} R_{j'i_1}^l}{\left[ 1 - \left( P_{i_1} \cdot \prod_{j' \in V_{i_1}/j} R_{j'i_1}^l \right)^2 \right]} \cdot r_{j_1 i_1}^l(0) \cdot r_{j_1 i_1}^l(1) \\
 &= \lim_{x \rightarrow 0} \frac{P_{i_1} \cdot x^{(\lambda_{i_1}-1)}}{x} \\
 &= \begin{cases} P_{i_1} & \lambda_{i_1} = 2 \\ 0 & \text{otherwise.} \end{cases}
 \end{aligned} \tag{12}$$



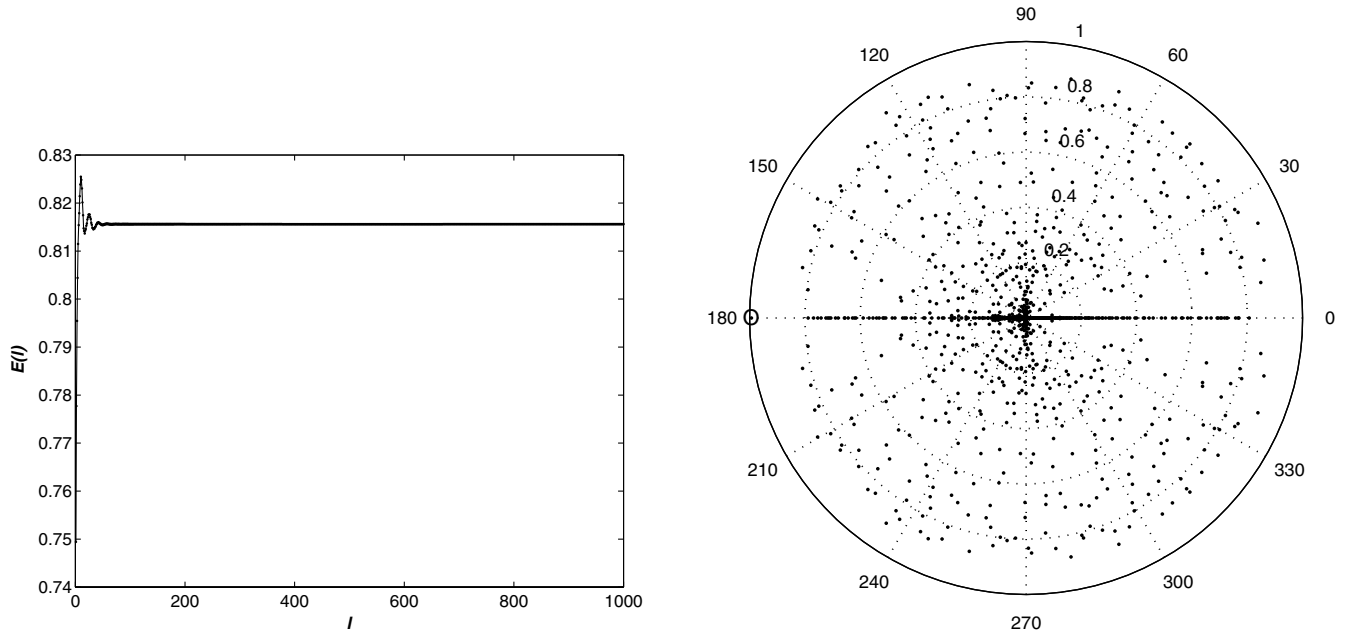


Fig. 11. Phase trajectory of  $E(l)$  before flip bifurcation. (Left)  $E(l)$  versus  $l$ . (Right) Polar plots of eigenvalues. SNR = 0.60 dB. Note that one eigenvalue approaches  $-1$ .

The above equation indicates that if the degree of the variable nodes is larger than 2, i.e.  $\lambda_{i_1} > 2$ ,  $i_1 = 1, 2, \dots, n$ , the eigenvalues at the unequivocal fixed point are all zeros and have nothing to do with the channel parameter. In Fig. 8(c), the eigenvalues all lie in the circle of diameter less than  $8 \times 10^{-20}$  (nonzero values caused by small computational inaccuracy).

With other noise realizations, the phenomena of Neimark–Sacker bifurcation (see Figs. 9 and 10) as well as flip bifurcation (see Figs. 11 and 12)

can be observed. Note that the noise realization used to observe a Neimark–Sacker bifurcation is different from that used to observe a flip bifurcation.

#### 4. Conclusions

In this paper, we have studied the dynamical behavior of iterative LDPC decoders under a Gaussian noise channel to some depth. Simulation results have shown that bifurcations, including fold, flip and Neimark–Sacker bifurcations, occur in the “waterfall SNR region”. Specifically in the waterfall region, oscillations and chaos are produced and the decoding algorithms do not converge. However, the exact bifurcation behavior in the waterfall region varies for different noise realizations. Based on the analysis of the eigenvalues of the iterative algorithm at the fixed points, we note that the indecisive fixed point loses stability as SNR increases, whereas the unequivocal fixed point remains stable in the whole SNR region. Since the attracting basin of the desired unequivocal fixed point shrinks as SNR decreases, it will become more difficult for the algorithm to converge to such a fixed point. For practical purposes, it will be interesting to investigate the effects of various decoding parameters on the bifurcation behavior as such knowledge will be directly relevant to the design of the decoders to achieve improved performance.

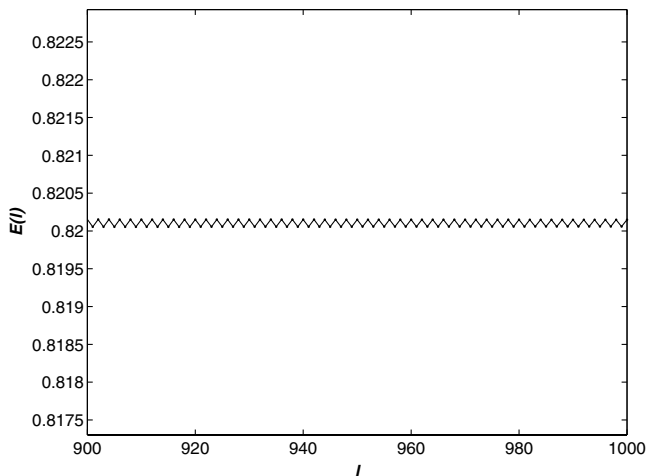


Fig. 12. Phase trajectory of  $E(l)$  after flip bifurcation. SNR = 0.65 dB.

## Acknowledgment

This work was supported in part by Hong Kong Polytechnic University under a special research grant awarded to Chair Professors.

## References

- Agarwal, D. & Vardy, A. [2001] "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inform. Th.* **47**, 699–722.
- Amraoui, A., Montanari, A., Richardson, T. J. & Urbanke, R. [2004] "Finite-length scaling for iteratively decoded LDPC ensembles," *CoRR cs.IT*, 0406050.
- Berrou, G., Glavieux, A. & Thitimajshima, P. [1993] "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. Int. Conf. Communication*, Geneva, Switzerland, pp. 1064–1070.
- Chen, J. H. & Fossorier, M. P. C. [2002] "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Inform. Th.* **50**, 406–414.
- Chung, S.-Y., Richardson, T. J. & Urbanke, R. [2001] "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Th.* **47**, 657–670.
- Di, C. Y., Proietti, D., Telatar, I. E., Richardson, T. J. & Urbanke, R. [2002] "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Th.* **48**, 1570–1579.
- Franz, S., Leone, M., Montanari, A. & Ricci-Tersenghi, F. [2002] "Dynamic phase transition for decoding algorithm," *Phys. Rev. E* **66**, 046120.
- Gallager, R. G. [1962] "Low-density parity-check codes," *IRE Trans. Inform. Th.* **8**, 21–28.
- Kocarev, L., Tasev, Z. & Vardy, A. [2002] "Improving turbo codes by control of transient chaos in turbo-decoding algorithms," *Electron. Lett.* **38**, 1184–1186.
- Kou, Y., Lin, S. & Fossorier, M. P. C. [2001] "Low-density parity-check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. Inform. Th.* **47**, 2711–2736.
- Lehmann, F. & Maggio, G. M. [2003] "Analysis of the iterative decoding of LDPC and product codes using the Gaussian approximation," *IEEE Trans. Inform. Th.* **49**, 2993–3000.
- MacKay, D. J. C. [2005] "Encyclopedia of sparse graph codes," available from: <http://wol.ra.phy.cam.ac.uk/mackay/codes/data.html>.
- Mao, Y. Y. & Banihashemi, A. H. [2001] "A heuristic search for good low-density parity-check codes at short block lengths," *Proc. Int. Conf. Communication*, Helsinki, Finland, pp. 41–44.
- Proakis, J. G. [1995] *Digital Communications*, 3rd edition (McGraw Hill, Singapore).
- Richardson, G. [2000] "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Th.* **46**, 9–23.
- Richardson, T. J. & Urbanke, R. [2001] "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Th.* **47**, 599–618.
- Richardson, T. J., Shokrollahi, M. A. & Urbanke, R. [2001] "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Th.* **47**, 619–637.
- Richardson, T., Shokrollahi, A. & Urbanke, R. [2002] "Finite-length analysis of various low-density parity-check ensembles for the binary erasure channel," *Proc. Int. Symp. Information Theory*, Lausanne, Switzerland, 1.
- Ryan, W. E. [2004] *An Introduction to LDPC Codes*, in CRC Handbook for Coding and Signal Processing for Recoding Systems, ed. Vasic, B. (CRC Press).
- Shokrollahi, A. [2003] "LDPC Codes: An introduction," available from: <http://www.ipm.ac.ir/IPM/homepage/Amin2.pdf>.
- Strogatz, S. H. [1994] *Nonlinear Dynamics and Chaos, with Applications to Physics, Biology, Chemistry, and Engineering* (Addison-Wesley).
- Xiao, H. & Banihashemi, A. H. [2004] "Graph-based message-passing schedules for decoding LDPC codes," *IEEE Trans. Commun.* **52**, 2098–2105.
- Zhang, J. & Orlitsky, A. [2002] "Finite-length analysis of LDPC codes with large left degree," *Proc. Int. Symp. Information Theory*, Lausanne, Switzerland, 3.