

Evidence of chaos in the Belief Propagation for LDPC codes

Jean-Christophe Sibel¹, Sylvain Reynal², and David Declercq²

- ¹ INRIA / ETIS / ENSEA / Univ. of Cergy-Pontoise / CNRS UMR 8051, 95014 Cergy-Pontoise, France
(E-mail: jean-christophe.sibel@ensea.fr)
- ² ETIS / ENSEA / Univ. of Cergy-Pontoise / CNRS UMR 8051, 95014 Cergy-Pontoise, France
(E-mail: reynal@ensea.fr, declercq@ensea.fr)

Abstract. In this paper, we investigate the behaviors of the Belief Propagation algorithm considered as a dynamic system. In the context of LDPC (Low Density Parity-Check) codes, we use the noise power of the transmission channel as a potentiometer to evaluate the different motions that the BP can follow. The computations of dynamic quantifiers as the bifurcation diagram, the Lyapunov exponent and the reconstructed trajectory enable to bring out four main behaviors. In addition, we propose a novel measure that is the hyperspheres method, which provides the knowledge of the time evolution of the attractor size. The information collected from these different quantifiers helps to better understand the BP evolution and to focus on the noise power values for which the BP suffers from chaos.

Keywords: LDPC, iterative map, chaos, Lyapunov exponent, bifurcation diagram.

1 Introduction

The channel coding is a research field whose purpose is to protect an information to transmit from environmental disturbances. The first step is the encoding of the information, a procedure in which the information, modeled as a sequence of k bits u_1, \dots, u_k , is mapped to a larger sequence of N bits x_1, \dots, x_N . The map consists in artificial correlations called constraints or parity-check equations. In [1] are introduced the Low-Density Parity-Check (LDPC) codes which are a widespread technique to encode the information. Such a code can be represented by a Tanner graph [2], a graphical representation which turns out to be very useful in the second step, the decoding. In this part, the bits transmitted through a random noisy channel are iteratively handled by a decoding algorithm to create an associated output sequence of N bits that verify the whole set of parity-check equations and that must be as close as possible to the input sequence. One of the most famous decoding algorithm is the Belief Propagation (BP) [3] used to solve inference in graphical



models. Extensively studied in [5,6], it is deemed to be the optimal message-passing algorithm in the case the Tanner graph of the LDPC code is loopfree. However, in most cases the Tanner graph is not loopfree [7] that involves that the BP becomes suboptimal. Moreover, the BP presents some complex behaviors in terms of the noise power of the transmitted channel, as periodic and chaotic motions [9]. Along the whole paper, we present some measures to bring out these different behaviors. The paper is organized as follows: in the second section are presented preliminaries about the LDPC codes and the BP, in the third section we present the dynamic environment of the BP, the measures to identify the behaviors and the associated results.

2 Preliminaries

2.1 Graphical Model – LDPC codes

We consider a set of N hidden binary random variables $\mathbf{X} = \{X_1, \dots, X_N\}$ whose global state is denoted by $\mathbf{x} = [x_1, \dots, x_N]$. To each variable X_i is associated an observation y_i that provides a prior information on the state of X_i given that the a posteriori distribution on X_i is proportional to the likelihood:

$$p(x_i|y_i) \propto p(y_i|x_i)$$

In the digital communications area, the hidden variables play the role of bits to transmit through a noisy channel, the observations represent the data collected at the output at the channel. These data are used to compute $\hat{\mathbf{x}} = [\hat{x}_1, \dots, \hat{x}_N]$ the estimate of the input sequence \mathbf{x} , as it is shown on the figure 1.

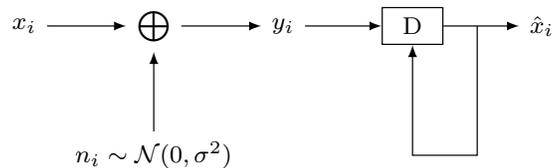


Fig. 1. Digital communication pattern: the channel is an additive Gaussian channel of power σ^2 , D is the estimation block that provides $\hat{\mathbf{x}}$

To ensure reliable communications is included the use of an LDPC code. An LDPC code is built by a set of M constraints $\mathbf{C} = \{C_1, \dots, C_M\}$ usually called parity-check equations. The value of the constraint C_j is computed by the set of variables mapped by C_j , namely its neighborhood \mathcal{N}_j such that:

$$c_j = \sum_{X_i \in \mathcal{N}_j} x_i$$

where the sum is computed over the Galois field $\text{GF}(2)$. The variables and the parity-check equations are respectively associated to the variable nodes and the check nodes of the graphical representation of the LDPC code, called the

Tanner graph $\mathbf{G} = (\mathbf{X} \cup \mathbf{C}, \{e_{ij}\})$. The check node C_j and the variable node X_i are linked by an edge e_{ij} if $X_i \in \mathcal{N}_j$. We define the neighborhood \mathcal{N}_i of the variable node X_i as the set of check nodes that map X_i . An example of a Tanner graph is displayed on the figure 2.

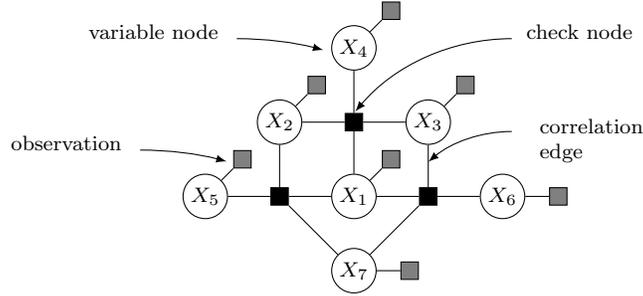


Fig. 2. Tanner graph of the Hamming code ($N = 7$)

2.2 Message-passing

The BP is an algorithm that helps to solve inference in graphical models. More accurately, it provides estimates $\{b_i(x_i)\}_i$ of the posterior marginal distributions of the variables, called beliefs. From these trial distributions can be extracted an estimate of \mathbf{x} such that:

$$\hat{\mathbf{x}} = \bigcup_{i=1}^N \arg \max_{x_i} b_i(x_i)$$

To obtain the beliefs, the BP passes messages iteratively between the variable nodes and the check nodes, according to their neighborhood dependence. An edge e_{ij} carries two different messages, each oriented in a specific way:

- the message from C_j to X_i is: $n_{ji}^{(k)}(x_i) = f_{ji}(\{m_{xy}^{(k-1)}\}_{(x,y)})$
- the message from C_i to C_j is: $m_{ij}^{(k)}(x_i) = g_{ij}(\{n_{yx}^{(k)}\}_{(x,y)}, l_i(x_i))$

where f_{ij} and g_{ji} are update functions whose expressions are detailed in [2], and $l_i(x_i)$ is the likelihood computed from the observation y_i . To give an idea of the analytic expressions of these functions, a message from a node A to a node B, whatever their nature, is somehow the geometric average of the messages incoming on A. The output of the BP is a set of beliefs that are computed from the same principle [2]: $b_i(x_i)$ is somehow the geometric average of the messages incoming on the variable node X_i in the state x_i .

2.3 Topological troubles

The BP has been introduced by Pearl [3] as an algorithm to solve inference on trees and polytrees. For such graphical model, this algorithm is surely optimal.

Though, the most of the LDPC codes have basically non tree-like topology, their Tanner graphs are full of loopy structures. This drawback is unavoidable because the check nodes need to be interwoven to make the LDPC code robust against the channel noise. Accordingly, the BP turns out to be suboptimal in most cases. In [7] and [8] the BP is investigated to bring out some convergence conditions depending on the topology of the Tanner graph, it was found that short loops are the most harmful and that the convergence of the BP must be unreachable if the LDPC code contains at least two loops. This conclusion brings the fact that most LDPC codes cannot be decoded perfectly by the BP. As a result are shown on the figure 3 two Bit Error Rates (BER) according to the Signal-to-Noise Ratio (SNR) on loopy codes of the same length N : one contains only large loops and the other only short loops.

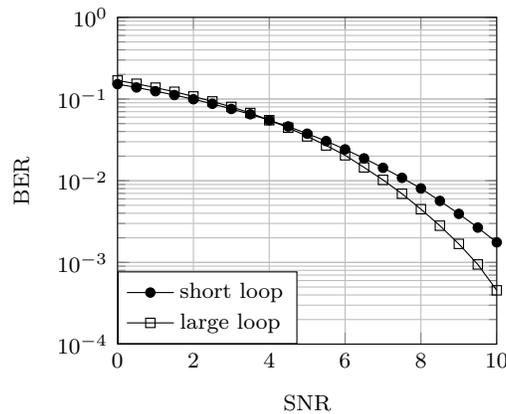


Fig. 3. BER of the BP – Difference between short loops and large loops

The BER of the code with large loops is less damaged than the one with short loops, that confirms the previous conclusion. This can be easily understood given that the BP is a message-passing algorithm: short loops have short term effects because only a small number of iterations is necessary to develop their harmful effect, contrary to the large loops.

Despite the practical interest of such an estimator, the BER does not bring the whole information about the behavior of the BP in case of loopy LDPC codes. On the figure 4 is displayed the evolution of the BER along the iterations according to four SNR values given a particular noise realization on the Tanner code [10] of length $N = 155$. It appears obvious that the BP suffers from great divergence depending on the SNR. The rationale behind these results is that the BP does not behave trivially as it could be wrongly thought given the BER.

In other terms, the SNR plays the role of a parameter that wields great influence on the behavior of the BP. Therefore, it appears necessary to investigate the BP as a dynamical system, a work presented in the next section.

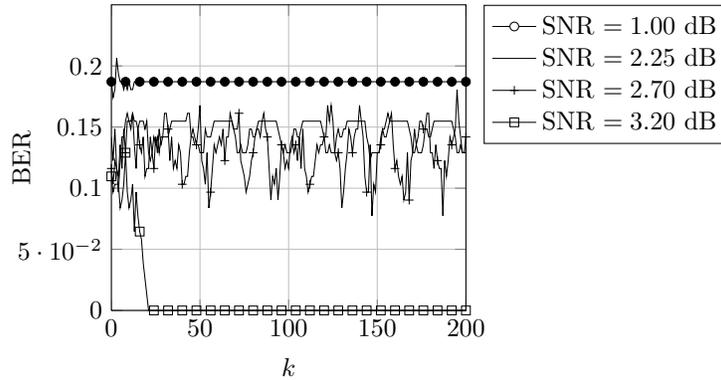


Fig. 4. BER of the BP on the Tanner code for four SNR values

3 Dynamics

In this section is presented an experimental study that brings out the dynamics of the BP algorithm according to the evolution of the SNR. To this end is introduced a toolbox including four estimators, each one carrying relevant information on the system. For each of them is presented the computation method, then a few results and finally the properties it reveals concerning the BP. The goal is to get information about:

- the SNR values that correspond to blatant changes in the behavior of the BP,
- the description of the different attractors the BP encounters,
- the size of these attractors.

Introducing the term attractor implies the definition of a state space. In the current study, such a space is built in such a way that each message n_{ji} defines a state variable. However, such a state space is of very high number of dimensions, given that a Tanner graph in practice could contain tens of thousands edges. Fortunately, experiments show that it is quite equivalent to consider the beliefs as pseudo-state variables, reducing dramatically the number of dimensions. Finally, in the following, all estimators are measured in the pseudo-state space such that each state variable is associated to a unique belief.

3.1 Bifurcation diagram

First of all, it appears necessary to go a little more deeply in the study of the figure 4. One would note that the BERs suffer from a threshold phenomenon, especially for SNR = 2.25 dB. This is actually an unfortunate consequence of the decoding process D, see figure 1, that thresholds the beliefs such that:

$$\forall X_i \in \mathbf{X}, \quad b_i(x_i) \in [0, 1] \mapsto \hat{x}_i \in \{0, 1\}$$

To extract relevant information concerning the BP, it is recommended to consider estimators that faithfully render the conduct of the BP. To this end, it

appears well suited to replace the BER by a smoother function, namely the mean square beliefs introduced in [9]:

$$\forall k \in \{1, \dots, K\}, \quad E(k) = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(b_i^{(k)}(x_i) \right)^2} \quad (1)$$

where the values $\{x_i\}_i$ are assumed to be the right ones and K is an arbitrary number of iterations. Its properties are partly equivalent to the BER ones in the sense that:

- $E(k) = 1$: the BP has perfectly decoded,
- $E(k) = 0.25$: the BP does not provide any relevant knowledge on the variables,
- $E(k) = 0$: the BP completely failed.

Experiments show that $E(k)$ lives between the two first situations, furthermore its evolution along the iterations is indeed softer than the BER, see figure 5.

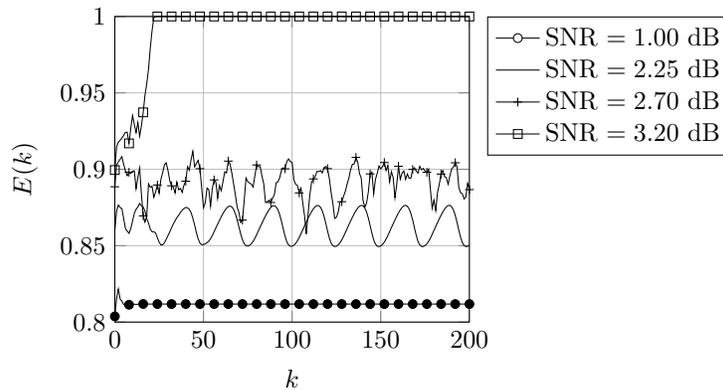


Fig. 5. Mean square beliefs of the BP on the Tanner code for four SNR values

On this figure appears the strong dependency of the BP dynamics on the SNR, the algorithm does not converge for all values. It is either stuck in steady states for $\text{SNR} \in \{1.00 \text{ dB}, 3.20 \text{ dB}\}$ or divergent for $\text{SNR} \in \{2.25 \text{ dB}, 2.92 \text{ dB}\}$. To draw an evolution of the mean square beliefs according to the SNR comes out the use of the bifurcation diagram. Instead of displaying the whole evolution of $E(k)$ along the iterations we only pick up its final value $E(K)$. Theoretically, in [9], at K the BP is expected to have reached a steady state. In practice, as shown by the figure 5, it is absolutely not systematic. For computation time's sake, the steady state is redefined as the permanent evolution after an arbitrary number of iterations. On the figure 6 are displayed the bifurcation diagrams of the BP for four noise realizations that we call Error Events (EE).

The bifurcation diagrams reveal critical SNR values that blatantly change the BP conduct. It appears five particular behaviors of $E(K)$:

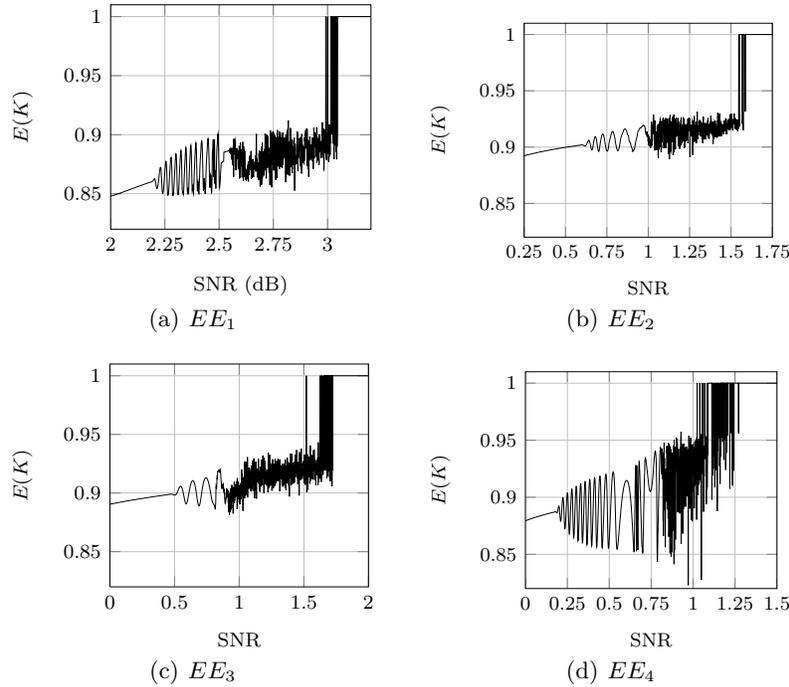


Fig. 6. Bifurcation diagrams of the BP

- (B1) smooth increasing,
- (B2) oscillations,
- (B3) erratic evolution,
- (B4) convergence jumps,
- (B5) convergence.

At this point appears a clue about the chaotic behavior of the BP (B3), even though are needed other observations to confirm it. Despite the order pattern is common to the error events, the critical SNR values are not the same. Actually it is strongly possible that most noise realizations lead to quite similar SNR critical values, the current difference we observe should correspond to the variance of the estimator, given that the number of simulations is quite small. A suited method to solve this problem would consist in average on a set of numerous noise realizations to obtain a mean bifurcation diagram with relevant variance. However, it appears quite impossible to conduct such a process. The reason comes out of the efficiency of the BP in terms of error correction, the error events that lead to non trivial behaviors of the BP along the SNR corresponds to rare events. Other regular error events imply very fast convergence of the BP to the perfect estimate. On the figure 7 is displayed the BER of the Tanner code decoded by the BP.

This figure makes appear that around $\text{SNR} = 2.50$ dB, for example, only one error event among a thousand will lead to wrong decoding by the BP, making the average computation of the bifurcation diagram quite untractable. Despite

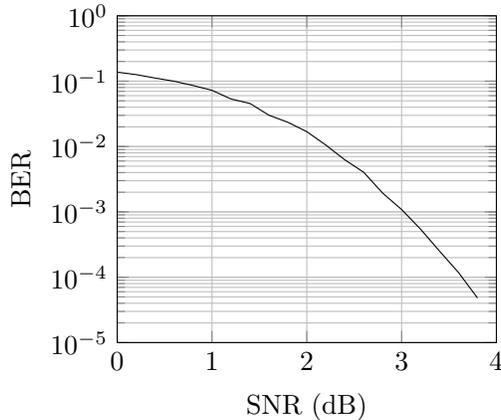


Fig. 7. BER of the Tanner code decoded by the BP

this drawback, a very important point raised up from extensive and numerous experiments is that the order of the five behaviors brought out previously is always the same, whatever the error event provided that it implies non trivial behaviors, and whatever the LDPC code. Therefore it is also always possible to extract four critical SNR values that share the whole range in five intervals corresponding with the behaviors B1, B2, B3, B4 and B5.

3.2 Lyapunov exponents

Given the critical SNR values, the next step is to find out the kind of the behaviors that were brought out, even though a few clues are given by the bifurcation diagram. To this end, we investigate the sensitivity of the BP to very small changes in the initial conditions, *i.e.* the likelihoods, by the use of the famous Lyapunov exponent. The computation of this estimator is made according to the method exposed in [12,11]. First of all we evaluate at each iteration $k \leq K$ the Euclidean distance d_k between two initially close trajectories. Then we estimate the Lyapunov exponent λ as the slope of the least square regression line of $\ln d_k$ along the iterations. Actually this method comes from the observation that for strongly divergent behavior, d_k follows an exponential law whose parameter is λ , as we can see on the figure 8.

On the figure 9 are displayed the Lyapunov exponents averages around the four error events introduced previously according to the Euclidean distance. The sign of λ reveals the behavior of the system around the corresponding initialization of the trajectories: $\lambda \geq 0$ means the trajectories have moved away one from the other, which is an evidence of a chaotic behavior, $\lambda \leq 0$ means the trajectories have got closer, which is an evidence of a convergent behavior to a small sized volume of the state space. This volume is reduced to a fixed point if and only if $\lambda \rightarrow -\infty$. When λ crosses the x-axis the system suffers from a bifurcation meaning that the algorithm has changed of conduct, as it was observed about the bifurcation diagram. To each SNR interval we obtain conclusions from λ :

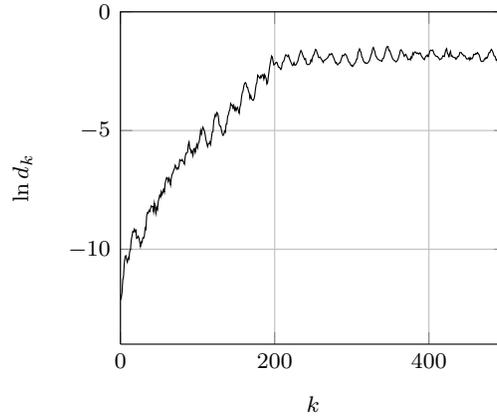


Fig. 8. Evolution of the log-distance between initially close trajectories for the BP, SNR = 2.90 dB, EE_1 . The evolution is exponential in $k \in [0, 200]$. For $k \geq 200$ appears a stair due to the compacity of the state space.

- (B1) $\lambda = 0$: the trajectories are quite close but never merge
- (B2) $\lambda = \text{cst} > 0$: the trajectories are moving away at a constant rate
- (B3) $\lambda > 0, \lambda \neq \text{cst}$: erratic evolutions of the trajectories, evidence of chaos
- (B4) λ getting lower: the trajectories begin to move closer
- (B5) $\lambda \rightarrow -\infty$: the trajectories merge

It is commonly accepted that the Lyapunov exponent provides a reliable signature of the behavior of any dynamical system. Therefore we can assert that the BP encounters chaos in the SNR interval B3 that is not of neglectible length. In addition, it appears that this chaos appears and disappears quite suddenly in terms of the SNR, looking at the slope of λ . It means that practically we can easily define a chaotic interval of the SNR for any LDPC code.

3.3 Reduced trajectory

The previous estimators revealed properties of the BP according to the SNR. A convenient approach to enforce these observations is to visualize the dynamical system in its state space.

However, our human skills prevent us from directly observing a system whose number of dimensions is several hundreds or even thousands, that is the case currently. To circumvent this undesired problem, we define a reduced 3-dimensional pseudo-state space and then a reduced pseudo-trajectory. To this end we make use of the state space reconstruction [11]. It consists in constructing a state space of arbitrary number of dimensions given a one dimensional map computed from the state variables. In this investigation, a known map of such property is the mean square beliefs. Firstly, the method aims to compute $E(k)$ at each iteration k to get a sequence $\mathbf{E} = [E(k)]_{0 \leq k \leq K}$. Secondly we map this

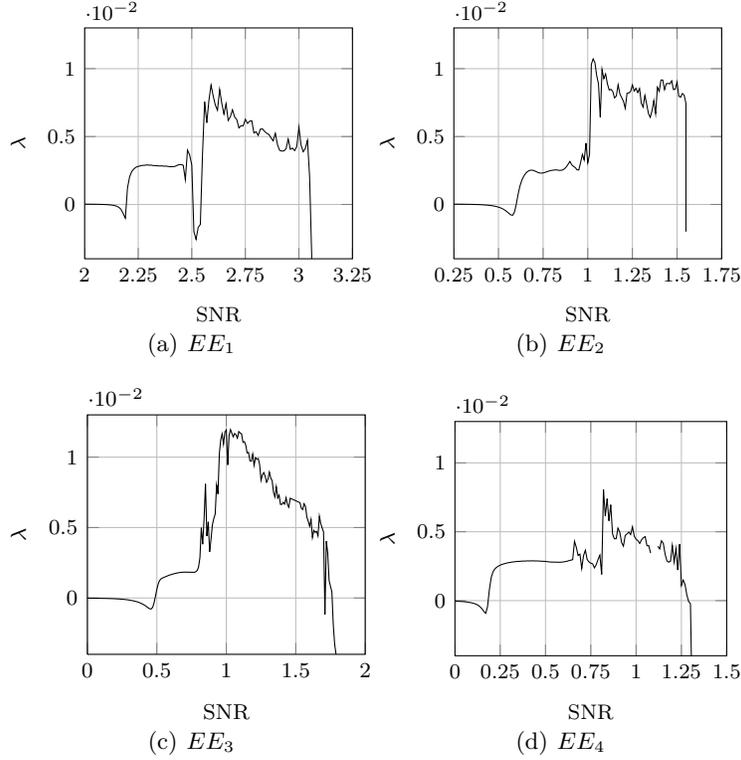


Fig. 9. Lyapunov exponents of the BP on the Tanner code

one dimensional sequence to a three dimensional sequence:

$$\mathbf{E} \mapsto \tilde{\mathbf{E}} = \begin{bmatrix} E(0) & E(1) & E(2) \\ \vdots & \vdots & \vdots \\ E(K-2) & E(K-1) & E(K) \end{bmatrix}$$

On the figure 10 are displayed a few reduced trajectories of the BP for typical values of the SNR deduced from the previous bifurcation diagram. On the first figure is exhibited at SNR = 2.10 dB a convergence of the trajectory toward a small size attractor, as it was expected according to the corresponding λ . By increasing the SNR between 2.19 dB and 2.49 dB the trajectory transforms to a limit cycle. The thickness of the trajectory along this limit cycle increases as the SNR is getting greater up to 2.50 dB. At the same time this limit cycle interleaves with other limit cycles, the BP encounters a sequence of period doubling bifurcations, displayed on the figure 11 with two interleaved cycles. Such a phenomenon is a typical route to chaos [11], a behavior observable from SNR = 2.51 dB. There is not any periodic evolution or fixed point convergence anymore, as it is displayed for 2.70 dB. When the SNR reaches 2.99 dB the trajectory collapses to a single point, meaning that the BP has correctly converged. Such behaviors are similar to the results of other experiments led on

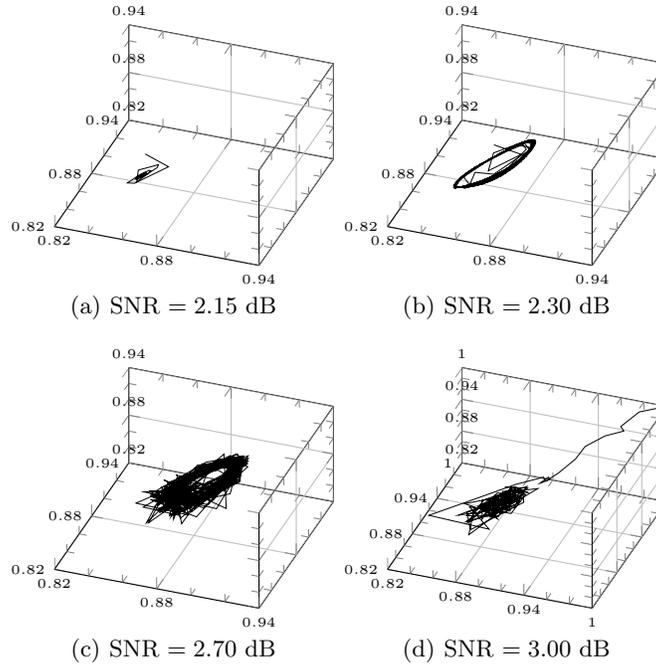


Fig. 10. Reduced trajectories for the BP on the Tanner code for the error event EE_1

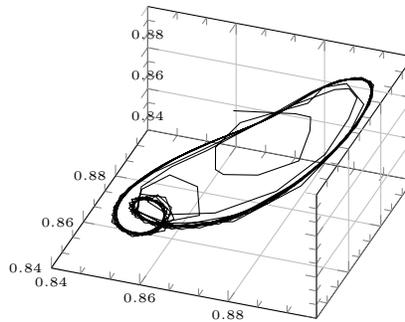


Fig. 11. Reduced trajectory for the BP on the Tanner code with SNR = 2.40 dB

other error events, that enforces the common dynamics between different noise realizations.

3.4 Hyperspheres method

Telling an attractor is chaotic is not enough to describe the situation of the BP. The assumed chaotic attractor can have different shapes and sizes. These properties are really important because they are the signaure of the practical unstability. A small chaotic attractor would be less troublesome than a large one, because the corresponding beliefs would be less eventful. In other terms,

we need to reveal the chaos intensity of the BP, somehow given by the size of an attractor in the pseudo-state space.

Computing such a quantity turns out to be a quite hard task because it depends on the shape of the attractor. Assumed that we find this shape, nothing ensures that it is part of our knowledge, contrary to the regular forms whose analytic expressions of the volumes are known, as the spheres, the ellipsoids, the hypercubes... To circumvent this problem, we establish a procedure that provides the hypersphere circumscribed to the pseudo-trajectory. Obviously the whole trajectory is not taken into account partly because it is important to get rid of the transient.

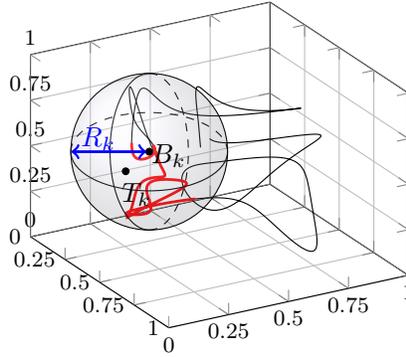


Fig. 12. Hyperspheres of radius R_k centered on B_k , the mean point of the pseudo-trajectory $\{T_{k-\frac{W}{2}}, \dots, T_{k+\frac{W}{2}}\}$. The red points are inside the hypersphere.

As shown on the figure 12, the method consists in:

- drag a temporal window $I_W(k)$ of arbitrary length W along the trajectory according to the iteration k ,
- for each k extracting B_k the mean point of the pseudo-trajectory in $I_W(k)$,
- searching for T_k the furthest point of B_k inside $I_W(k)$.

The vector $\overrightarrow{B_k T_k}$ is of length r_k the radius of the hypersphere circumscribed to the pseudo-trajectory inside $I_W(k)$. A part of the estimator is the evolution of r_k along the iterations. The difficulty lies in the experimental search for the length W such that the attractor is absorbed into the hypersphere. For the moment, only numerous experimental attempts help to find the well suited W . We present on figure 13 the evolution of r_k for the BP subjected to strong chaos according to the previous estimators.

We see that the values of the radii are of the same magnitude, even though they are not strictly equal. Nevertheless, in the current pseudo-state space of $N = 155$ dimensions, the hypervolume of the hypersphere is proportionnal to R_k^N . Then even a small difference between two radii involves a non neglectible difference, a fact that we will see later. Another observation is the fact that these radii are not constant, meaning that the hyperspheres shake. Due to this phenomenon the radius is often almost doubled as for EE_2 about

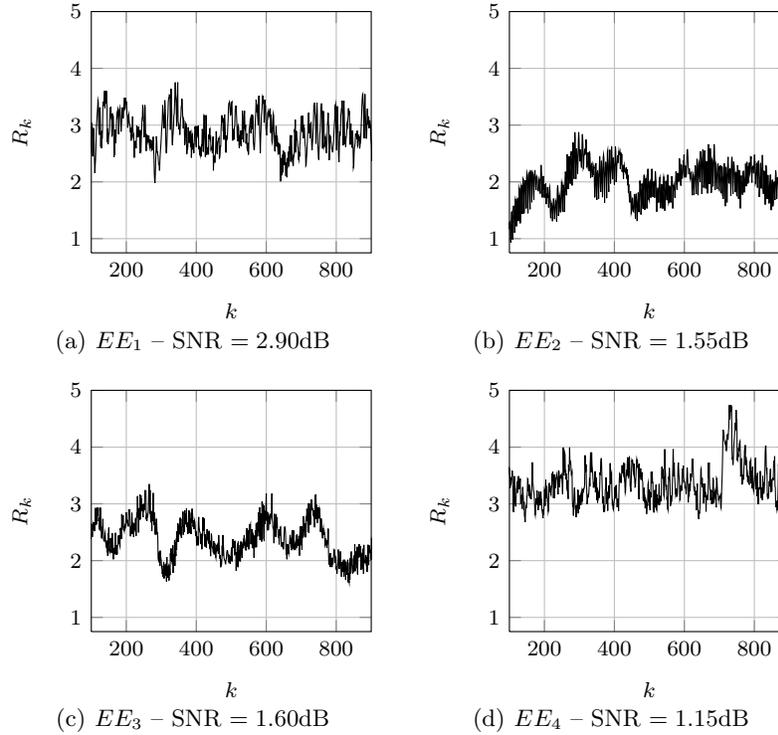


Fig. 13. Hypersphere radius R_k for chaotic attractors according to k

$k \in \{209, 298\}$. This is a consequence of the BP instability. To get a global overview of the attractor size inside the state space we display on figure 14 the average steady value \hat{R}_K for each error event according to the SNR. According to the previous estimators, it appears that the maxima of the radii are reached when the BP is trapped into chaotic attractors, and the minima are reached as soon as the algorithm left these attractors. In addition, a quite interesting observation is that \hat{R}_K is smoothly increased while the SNR lies within the limit cycle interval, meaning that these limit cycles are getting larger as the SNR is increased.

In the mean time, the Lyapunov exponent highlighted the information that initially close trajectories were moving away at a constant rate. Therefore as the limit cycle is growing, the stability of the BP does not really change provided the SNR is less than the critical values that leads to chaos. In other terms the divergence speed of two initially close trajectories is not changed even if the radius of the limit cycle increases, that is a quite surprising observation.

Finally it appear suitable to compare the values of the hyperspheres volumes so as to highlight their difference. In the table 3.4 are given the ratios between the maximum radii and the associated ratios between their corresponding hypervolumes, given that :

- $R_1 \triangleq \max \hat{R}_k(EE_1) = 2.6561$ reached at $\text{SNR} = 2.75$ dB,

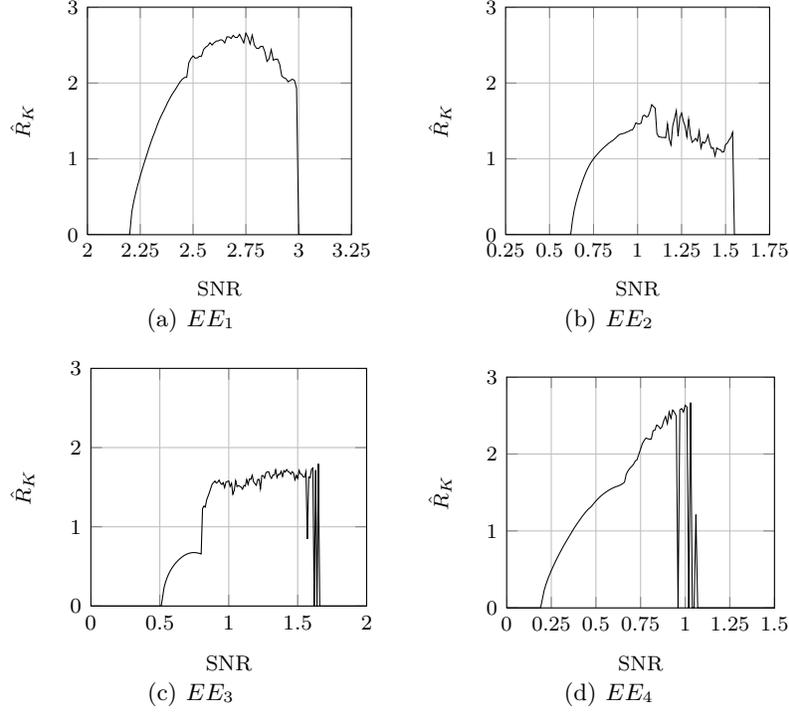


Fig. 14. Average hypersphere radius \hat{R}_K for chaotic attractors according to the SNR

- $R_2 \triangleq \max \hat{R}_k(EE_2) = 1.7139$ reached at SNR = 1.08 dB,
- $R_3 \triangleq \max \hat{R}_k(EE_3) = 1.7954$ reached at SNR = 1.65 dB,
- $R_4 \triangleq \max \hat{R}_k(EE_4) = 2.6684$ reached at SNR = 1.03 dB.

These tables raise the dramatical difference between the size of the chaotic attractors.

As an example, EE_4 involves a radius only 1.5569 times larger than the radius involved by EE_2 but $V_4 \approx 10^{29}V_2$ which is a very large difference. On the contrary error events whose radii are very close, as EE_1 and EE_4 do not differentiate much in terms of their corresponding hypervolumes. Finally, it appears a quite large diversity of chaotic attractors for the BP. This diversity lies within the intensity of the chaos, represented by the hypervolumes. This observation indicates that for example EE_2 entails a *less* chaotic attractor than EE_4 , and that EE_1 implies a much *more* chaotic attractor than EE_2 . Such comparisons provide somehow a reliability coefficient on the noise realizations, that is of very important practical interest.

4 Conclusion

In this paper, we address the dynamics issue of the BP by the use of known and new estimators from an experimental point of view. We brought out that the BP

(a) Ratios between hyperspheres radii

	EE_1	EE_2	EE_3	EE_4
EE_1	.	$\frac{R_1}{R_2} = 1.5497$	$\frac{R_1}{R_3} = 1.4794$	$\frac{R_1}{R_4} = 0.9954$
EE_2	$\frac{R_2}{R_1} = 0.6453$.	$\frac{R_2}{R_3} = 0.9546$	$\frac{R_2}{R_4} = 0.6423$
EE_3	$\frac{R_3}{R_1} = 0.6759$	$\frac{R_3}{R_2} = 1.0476$.	$\frac{R_3}{R_4} = 0.6728$
EE_4	$\frac{R_4}{R_1} = 1.0046$	$\frac{R_4}{R_2} = 1.5569$	$\frac{R_4}{R_3} = 1.4862$.

(b) Ratios between hyperspheres volumes

	EE_1	EE_2	EE_3	EE_4
EE_1	.	$\frac{V_1}{V_2} = 3.0788 \times 10^{29}$	$\frac{V_1}{V_3} = 2.3082 \times 10^{26}$	$\frac{V_1}{V_4} = 0.4894$
EE_2	$\frac{V_2}{V_1} = 3.2588 \times 10^{-30}$.	$\frac{V_2}{V_3} = 7.4528 \times 10^{-4}$	$\frac{V_2}{V_4} = 1.5826 \times 10^{-30}$
EE_3	$\frac{V_3}{V_1} = 4.2833 \times 10^{-27}$	$\frac{V_3}{V_2} = 1.3499 \times 10^3$.	$\frac{V_3}{V_4} = 2.1005 \times 10^{-27}$
EE_4	$\frac{V_4}{V_1} = 2.0368$	$\frac{V_4}{V_2} = 6.3156 \times 10^{29}$	$\frac{V_4}{V_3} = 4.6987 \times 10^{26}$.

follows a systematic pattern when the decoding is not trivial: convergence to a small-sized attractor, locking in a limit cycle, chaos and convergence to a fixed-point. Such a property turns out to be practically relevant because it is common to all LDPC codes. In addition it provides the critical values of the SNR for which the BP could present complex behaviors. We investigated the chaos by the use of new estimators to highlight the diversity of the chaotic attractors that the BP would encounter. By the use of the hyperspheres method we introduced the notion of chaos intensity that highlighted a novel notion of reliability on the channel noise realizations, and to some extent on the SNR values and the LDPC codes. Finally we have introduced a quite efficient toolbox for the study of the BP that can be adapted to any decoding algorithm provided its output can be computed as probability distributions.

References

- 1.R.G. Gallager. Low-Density Parity-Check Codes. *PhD Thesis, MIT*, 1963.
- 2.F.R. Kschischang, B. J. Frey eand H.A. Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Trans. on Inf. Theory*, 47.2, 2001.
- 3.J. Pearl. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan Kaufmann publisher*, 1988.
- 4.H.A. Bethe. Statistical Theory of Superlattices. *Proc. Roy. Soc. London*, 1935.
- 5.T.J. Richardson. Error floors of LDPC codes. *Proc. 41st Annual Allerton Conf. on Comm. Cont. and Comp.*, 2003.
- 6.S.Y Chung, T.J. Richardson and R.L. Urbanke. Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation. *IEEE Trans. on Inf. Theory*, 47.2, 2001.
- 7.K.P. Murphy, Y. Weiss and M.I. Jordan. Loopy belef propagation for approximate inference: an empirical study. *Proc. Uncertainty in AI*, 1999.

- 8.T. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 26:154–190, 2006.
- 9.X. Zheng, F.C.M. Lau, C.K. Tse and S.C. Wong. Study of bifurcation behavior of LDPC decoders. *Int. Journal of Bifurcation and Chaos*, 16:3435–3449, 2005.
- 10.R.M. Tanner, R. Michael, D. Sridhara and T. Fuja. A Class of Group-Structured LDPC Codes. 2001.
- 11.R. Hilborn. Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers. *Oxford University Press publisher*, 2000.
- 12.M.T. Rosenstein, J.J. Collins and C.J. De Luca. A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D*, 65:117–134, 1993.