Experimental results about the dynamics of the Generalized Belief Propagation used on LDPC codes

Jean-Christophe Sibel, Sylvain Reynal, David Declercq

Experimental results about the dynamics of the Generalized Belief Propagation used on LDPC codes

Jean-Christophe Sibel, Sylvain Reynal and David Declercq
ETIS / ENSEA / Univ. Cergy-Pontoise / CNRS UMR 8051
F-95000, Cergy-Pontoise, France
{jean-christophe.sibel,reynal,declercq}@ensea.fr

Abstract—In the context of channel coding, the Generalized Belief Propagation (GBP) is an iterative algorithm used to recover the transmission bits sent through a noisy channel. To ensure a reliable transmission, we apply a map on the bits, that is called a code. This code induces artificial correlations between the bits to send, and it can be modeled by a graph whose nodes are the bits and the edges are the correlations. This graph, called Tanner graph, is used for most of the decoding algorithms like Belief Propagation or Gallager-B. The GBP is based on a non unic transformation of the Tanner graph into a so called region-graph. A clear advantage of the GBP over the other algorithms is the freedom in the construction of this graph. In this article, we explain a particular construction for specific graph topologies that involves relevant performance of the GBP. Moreover, we investigate the behavior of the GBP considered as a dynamic system in order to understand the way it evolves in terms of the time and in terms of the noise power of the channel. To this end we make use of classical measures and we introduce a new measure called the hyperspheres method that enables to know the size of the attractors.

Keywords—iterative decoder, LDPC, region-graph, chaos.

I. 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 $u_1, \ldots, u_k$, is mapped to a larger sequence of N bits $x_1, \ldots, x_N$. 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 though 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) introduced in [3], extensively studied in [4], [5], [6], which 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. To circumvent this problem has been proposed the Generalized Belief Propagation (GBP) [8], [9] which is an adaptation of the Kikuchi approximation [10], [11] used in statistical physics. This algorithm is a message-passing algorithm which maps the Tanner graph to a non unic region-graph whose edges are media messages. The non uniqueness of the region-graph involves a large degree of freedom, thus it implies a possible way to go beyond the BP.

Along the whole paper, we focus on a particular LDPC code, the Tanner code [12] whose main property in our study is the fact that it can be entirely described by a set of particular combinations of loops called the Trapping-Sets [4], [13] that we use for the region-graph construction. The second section first deals with the preliminaries about the Tanner graph and the Belief Propagation. In the third section is explained the region-graph construction rules and the equations of the GBP with a novel construction of the region-graph for the Tanner code. The last section is dedicated to the exposure of some measures of dynamics to better understand the GBP and to highlight its relevant properties. Also we introduce a new measure called the hyperspheres method that enables to get the size of the attractors, that helps to evaluate the divergence of the algorithm.

II. Preliminaries

A. The Tanner graph

Let us consider a set of N binary random variables $\mathbf{X} = \{X_1, \ldots, X_N\}$ whose global state is denoted by $\mathbf{x} = [x_1, \ldots, x_N]$. In the following of the article, we use the notation $\mathbf{x} = [x_1, \ldots, x_N]$ to denote both the variables and their states. An LDPC code is built by a set of M constraints, or parity-check equations, $\mathbf{C} = \{c_1, \ldots, c_M\}$ such that for each check c_j:

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

where $\mathcal{N}_j$ is a subset of $\{x_1, \ldots, x_N\}$ called the neighborhood of c_j depending on the LDPC code, and the sum is computed over the Galois field GF(2). The neighborhood of a variable x_i is denoted by $\mathcal{N}_i$ and it is built by the set of checks $\{c_j\}_j$ such that $x_i \in \mathcal{N}_j$. We consider that a check c_j and a variable x_i, such that $x_i \in \mathcal{N}_j$, form an edge e_{ij} between two nodes inside an undirected bipartite graph called the Tanner graph $G = (\mathbf{X} \cup \mathbf{C}, \{e_{ij}\})$. This graph is used as the media for the propagation of messages for the BP algorithm detailed in [3],

[2]. An example of a Tanner graph is displayed on the figure 1.

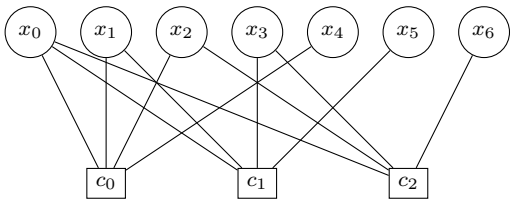


Fig. 1. Tanner graph of the Hamming code

By applying the constraints $\mathbf{C}$ on the variables $\mathbf{X}$ we obtain a Markov random field whose joint probability distribution is $p(\mathbf{X} = \mathbf{x})$ written simply $p(\mathbf{x})$. Each variable X_i has its own marginal probability distribution $p_i(x_i)$. We denote respectively by $b(\mathbf{x})$ and $\{b_i(x_i)\}_{1\leq i\leq N}$ the estimate, by any decoding algorithm, of the joint probability distribution and the marginal probability distributions, called the beliefs. The ouput sequence $\hat{\mathbf{x}}$ that is the estimate of the input sequence in the transmission channel is the most likely sequence given by:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} b(\mathbf{x})$$

The noisy transmission channel that we use in all of our simulations is an additive white Gaussian noise channel whose power is σ^2. The observations $\mathbf{Y} = \{y_1, \ldots, y_N\}$ we get at the output of the channel enable the computation of the likelihoods $\mathbf{L}(\mathbf{x}) = \{l_1(x_1) = p(y_1|x_1), \ldots, l_N(x_N) = p(y_N|x_N)\}$.

B. Belief Propagation

The BP is an iterative algorithm that consists in passing messages between the variables and the constraints on the edges of a given Tanner graph. An iteration $k \geq 1$ of this algorithm is classically done in two half iterations:

- for each edge e_{ij}, we compute the messages $n_{ji}^{(k)}(x_i)$ from the the constraint C_j to the variable x_i:

$$\begin{aligned} n_{ji}^{(k)}(0) &= \frac{1}{2} + \frac{1}{2} \prod_{X_{i'} \in \mathcal{N}_j \setminus X_i} \left(2m_{i'j}^{(k-1)}(0) - 1\right) \quad (1) \\ n_{ji}^{(k)}(1) &= 1 - n_{ji}^{(k)}(0) \end{aligned}$$

- for each edge e_{ij}, we compute the messages $m_{ij}^{(k)}(x_i)$ from the variable x_i to the constraint C_j:

$$m_{ij}^{(k)}(x_i) = \frac{l_i(x_i)}{Z_{ij}} \prod_{C_{j'} \in \mathcal{N}_i \setminus C_j} n_{j'i}^{(k)}(x_i) \qquad (2)$$

  where Z_{ij} is a normalization factor such that: $m_{ij}^{(k)}(0) + m_{ij}^{(k)}(1) = 1$,

To initialize the algorithm, we use the likelihoods:

$$\forall e_{ij}, \quad m_{ij}^{(0)} = l_i(x_i) \qquad (3)$$

The beliefs on a variable x_i are computed as the geometric averages on the incoming messages:

$$b_i^{(k)}(x_i) = \frac{l_i(x_i)}{Z_i} \prod_{j \in \mathcal{N}_i} n_{ji}^{(k)}(x_i) \qquad (4)$$

where Z_i is a normalization factor such that: $b_i(0) + b_i(1) = 1$.

III. The Generalized Belief Propagation

A. The region-graph construction

The region-graph $\mathcal{R}$ is a directed graph of depth D built level by level. We decompose this construction in two steps.

The first step is the construction of the first level $\mathcal{R}_0$. The principle is to gather the nodes of the Tanner graph in order to absorb some harmful topological structures, as loops or combinations of loops. These gatherings are the nodes of $\mathcal{R}_0$ if and only if each check c_j is included in at least one gathering accompanied by its neighborhood $\mathcal{N}_j$. We call a gathering a region.

The second step is the construction of the next levels $\mathcal{R}_1, \ldots, \mathcal{R}_{D-1}$. To build a level $\mathcal{R}_l$ we search for the intersections between the regions of the previous level $\mathcal{R}_{l-1}$. To this end, we first define $\mathbf{C}_r$ the set of constraints and $\mathbf{X}_r$ the set of variables that both form the region r. A set of variables and checks $r = \mathbf{X}_r \cup \mathbf{C}_r$ is a region of $\mathcal{R}_l$ if and only if there is a set of $n \geq 2$ regions $\{r_1, \ldots, r_n\} \in \mathcal{R}_{l-1}^n$ such that:

- $\forall c_j \in \mathbf{C}_r$, $\forall r_k \in \{r_1, \ldots, r_n\}$, $c_j \in \mathbf{C}_k$,
- $\forall x_i \in \mathbf{X}_r$, $\forall r_k \in \{r_1, \ldots, r_n\}$, $x_i \in \mathbf{X}_k$.

A region-graph of the Hamming code presented previously is displayed on the figure 2. The nodes of the first level are constructed considering only one check per region.

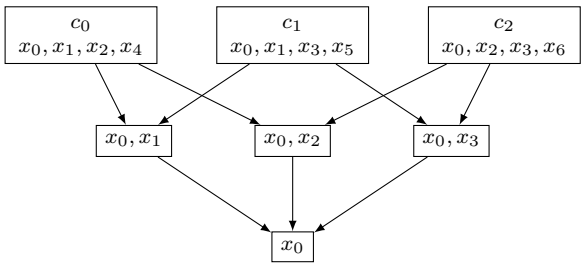


Fig. 2. A region-graph of the Hamming code

The efficiency of the GBP detailed below is fully dependent on the region-graph that is built from the Tanner graph. The construction of the first regions $\mathcal{R}_0$ determines the performance of the algorithm. Therefore, it is relevant to adapt the construction to the topology of the considered Tanner graph. The difficulty is to find a construction that both implies relevant performance and fair complexity. In the case of the previous example where we consider only one check per region of $\mathcal{R}_0$, the construction is easy to implement and the complexity is low enough to compare it with other algorithms though the performance are not improved enough or even worse.

B. The message-passing algorithm

The region-graph is a Bayesian network whose probability distribution can be computed by the following algorithm. The principle of the GBP is to convey messages on the edges of the region-graph where a message $m_{rq}^{(k)}$ from a region r to a region q is the a posteriori probability distribution of the region q given by the region r at iteration k. An iteration of the algorithm consists in the computation, or update, of all the messages $\{m_{rq}^{(k)}\}_{r,q}$ and the computation of all the beliefs

$\{b_r^{(k)}\}_r$ and $\{b_i^{(k)}\}_i$ whose equations are detailed just below. We denote by K the maximum value of k.

1) The update equations: The family $\mathcal{F}_r$ of a region r is defined as the set of regions such that for any region $q \subseteq \mathcal{F}_r$:

$$\begin{aligned} c_j \in \mathbf{C}_q \quad &\Rightarrow \quad c_j \in \mathbf{C}_r \\ x_i \in \mathbf{X}_q \quad &\Rightarrow \quad x_i \in \mathbf{X}_r \end{aligned}$$

We define the descendants $\mathcal{D}_r$ of a region r as $\mathcal{F}_r \backslash r$. The children $\mathcal{C}_r$ of r are the regions q of $\mathcal{D}_r$ such that there is an edge from r to q. According to [8], [9] the equations of the messages from a region r to a region q at iteration k are:

$$m_{rq}^{(k)}(\mathbf{x}_q) = \frac{\sum_{\mathbf{x}_{r\backslash q}} L_{r\backslash q}(\mathbf{x}_{r\backslash q}) c_{r\backslash q}(\mathbf{x}_r) \prod_{\substack{p \subset \mathcal{R}\backslash\mathcal{F}_r \\ s \subset \mathcal{F}_r\backslash\mathcal{F}_q}} m_{ps}^{(k-1)}(\mathbf{x}_s)}{Z_{rq} \prod_{\substack{p \subset \mathcal{D}_r\backslash\mathcal{F}_q \\ s \subset \mathcal{D}_q}} m_{ps}^{(k)}(\mathbf{x}_s)} \tag{5}$$

where:

- $\mathbf{x}_{r\backslash q}$ is the state of $\mathbf{X}_{r\backslash q} = \mathbf{X}_r \backslash \mathbf{X}_q$,
- $L_{r\backslash q}(\mathbf{x}_{r\backslash q}) = \prod_{x_i \subset \mathbf{X}_{r\backslash q}} l_i(x_i)$,
- $c_{r\backslash q}(\mathbf{x}_r) = \prod_{c_j \in \mathbf{C}_r \backslash \mathbf{C}_q} c_j(\mathbf{x}_{\mathcal{N}_j})$,
- Z_{rq} is a normalization factor to ensure that: $\sum_{\mathbf{x}_q} m_{rq}^{(k)}(\mathbf{x}_q) = 1$.

For any region r at iteration k the beliefs are:

$$b_r^{(k)}(\mathbf{x}_r) = \frac{1}{Z_r} L_r(\mathbf{x}_r) c_r(\mathbf{x}_r) \prod_{\substack{p \subset \mathcal{R}\backslash\mathcal{F}_r \\ s \subset \mathcal{F}_r}} m_{ps}^{(k)}(\mathbf{x}_s) \tag{6}$$

where Z_r is a normalization factor to ensure that $\sum_{\mathbf{x}_r} b_r^{(k)}(\mathbf{x}_r) = 1$. To get the beliefs for the single variables we only need to compute the marginal probability distributions of the regions beliefs. Thus for any region r:

$$\forall x_i \in \mathbf{X}_r,\ b_i^{(k)}(x_i) = \frac{1}{Z_i} \sum_{\mathbf{x}_{r\backslash i}} b_r^{(k)}(\mathbf{x}_r) \tag{7}$$

where Z_i is a normalization factor to ensure that $\sum_{x_i} b_i^{(k)}(x_i) = 1$.

2) The relaxation coefficient: The update equations (5) can be summarized into an implicit equation:

$$\forall (r,q) \in \mathcal{R}^2 \text{ s.t. } q \subset \mathcal{C}_r,\ m_{rq}^{(k)} = F_{rq}\left(\{m_{ps}^{(k-1)}\}_{p,s}, \{l_i\}_i\right) \tag{8}$$

In the following, we lighten the notations by the use of F_{rq} alone. An important point for an iterative algorithm is the convergence, that we define here as:

$$\forall (r,q) \in \mathcal{R}^2 \text{ s.t. } q \subset \mathcal{C}_r,\ m_{rq}^{(k+1)} = m_{rq}^{(k)}$$

In [8] is introduced the fact that the GBP suffers from a poor convergence. To circumvent this phenomenon is included a decreasing term of relaxation w_k such that:

$$m_{rq}^{(k)} = w_k F_{rq} + (1 - w_k) m_{rq}^{(k-1)} \tag{9}$$

The main issue is to find the most suitable function w_k which provides both convergence and relevant performance. We propose three functions for w_k that are displayed on the figure 3.

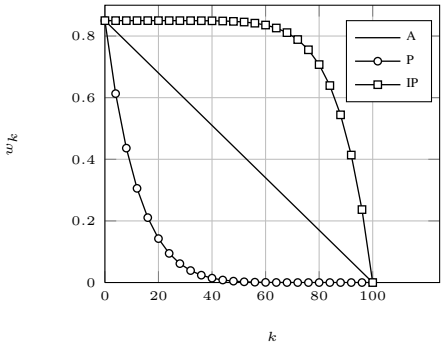
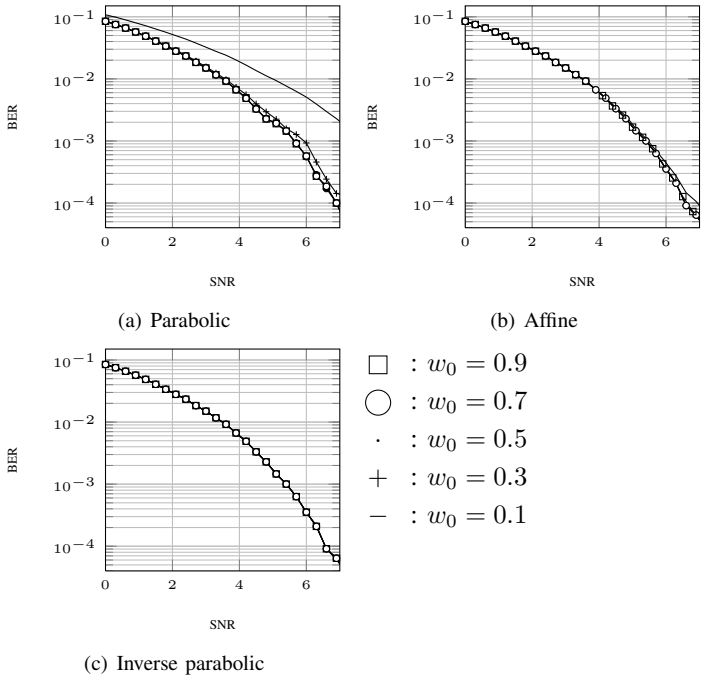


Fig. 3. Relaxation coefficient : (A) affine, (P) parabolic, (IP) inverse parabolic

The balance between the memory $m_{rq}^{(k-1)}$ and the update F_{rq} is different from case to case. In the case of a parabolic function, the memory is quickly highlighted at the expense of the update, whereas in the case of an inverse parabolic function the memory is practically ignored for a long time. The affine coefficient enables a progressive oversight of the update up to a complete use of the memory.

On the figure 4 are displayed the bit-error-rates (BER) on the Hamming code for the different relaxation coefficients. It appears that the BER are not the same, which indicates that the choice of w_k is not trivial.



(a) Parabolic (b) Affine

(c) Inverse parabolic

Fig. 4. Binary Error Rate on the Hamming code for different relaxation coefficents

We observe that the inverse parabolic coefficient involves the best performance whatever w_0, unlike the affine coefficient or the parabolic coefficient which is the worst one. Therefore, we can assert that the best choice for w_k is the inverse parabolic one. Another idea would be to create a function w_k whose values are computed at each iteration k according to the difference between the update and the memory, that can be considered as an adaptative relaxation coefficient. Yet,

the results are really poorer than what we can get from the function proposed previously so we dropped this function. Finally, extensive simulations let us conclude that the choice of the relaxation coefficient is completely dependent on the code we use in the transmission. Therefore, we need to test different functions w_k for each code to select the best one.

C. A novel construction

We present here a particular construction of the region-graph for a given LDPC code. We consider the Tanner code [12] of length $N = 155$ with $M = 93$ parity-check equations. The Tanner graph of this code can be established by a sophisticated concatenation of topological structures called Trapping-Sets (TS). A TS(a,b) introduced in [4] and studied in [13] among other, is a Tanner graph which contains a variables and b unsatisfied parity-check equations. In the studied case the TS$(5, 3)$, whose Tanner graph is represented on the figure 5, are sufficient to cover the whole Tanner graph of the code.

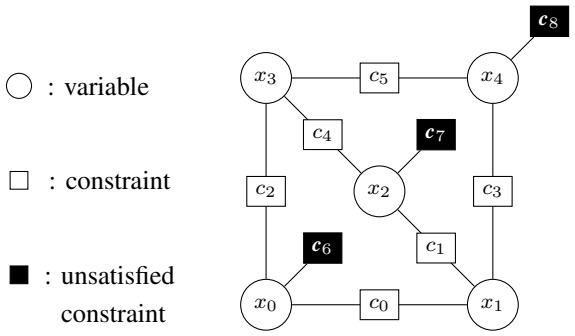


Fig. 5. TS$(5, 3)$: 5 variables and 3 unsatisfied constraints

Such a structure is known to damage the decoding [13] because only the total null state enables to verify all the parity-check equations. Thus to build the first level $\mathcal{R}_0$ of the region-graph, it would be relevant to absorb them into regions. However, the complexity of the message-passing would soar because these regions are too large. Therefore, we had better break these structures into several smaller regions as on figure 6 with:

- $r = \{\mathbf{X}_r = \{x_0, x_1, x_3\}, \mathbf{C}_r = \{c_0, c_2, c_6\}\}$,
- $p = \{\mathbf{X}_p = \{x_1, x_2, x_3\}, \mathbf{C}_p = \{c_1, c_4, c_7\}\}$,
- $q = \{\mathbf{X}_q = \{x_1, x_3, x_4\}, \mathbf{C}_q = \{c_3, c_5, c_8\}\}$.

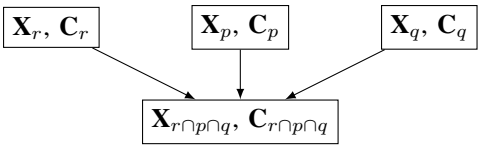
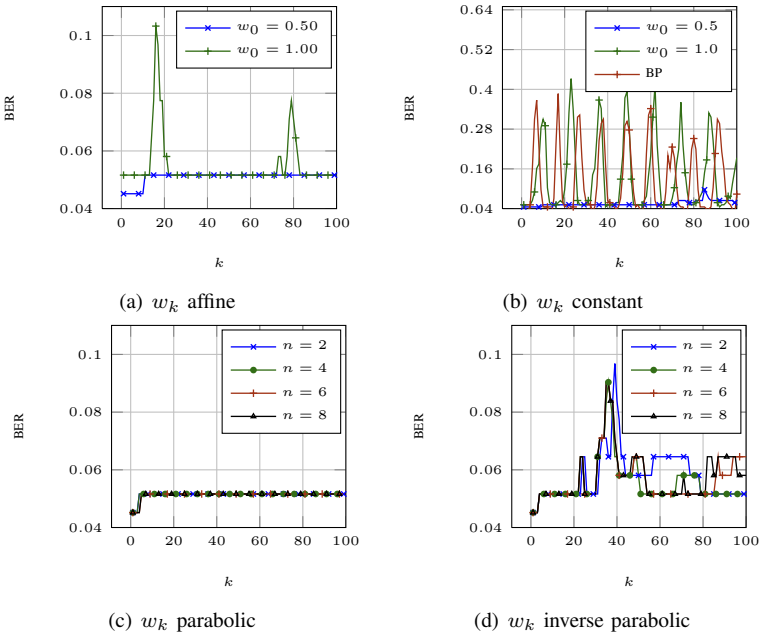


Fig. 6. Region-graph of the TS$(5, 3)$

D. Results

The region-graph of the TS$(5, 3)$ is loopfree which makes the decoding optimal. When we apply this construction to all the TS$(5, 3)$ of the whole code, unfortunately we do not get a loopfree region-graph because all the variables belong to several TS$(5, 3)$. Nevertheless, the performance are still relevant to make the GBP a good candidate for the decoding. We present on the figures 7 the BER in terms of the iteration of the GBP on the Tanner code for error events made by the noisy channel that are truly harmful for the Belief Propagation particularly because of the TS$(5, 3)$.



(a) w_k affine (b) w_k constant

(c) w_k parabolic (d) w_k inverse parabolic

Fig. 7. BER of the BP and the GBP for different relaxation coefficients on the Tanner code with the suited region-graph construction

On the figure (b) is displayed the BER of the BP which is oscillating, making the decoding divergent and wrong. On the same figure is displayed the BER of the GBP when the memory is not taken into account. It clearly appears that such an algorithm does not bring any improvement noticing that it is often worse than the BP. When w_k is set to 0.5 then the BER is dramatically reduced. On the figure (a) is presented the fact that to keep a smooth evolution of the BER with relevant values the relaxation coefficient must not imply a total oversight at the first iteration. Extensive simulations make us convinced that for the Tanner code, the initial value $w_0 = 0.85$ is the most relevant to get the best results, whatever the relaxation coefficient decreasing rule. The most interesting point is the comparison between figures (c) and (d) where n is the order of the parabola and of the inverse parabola:

- parabolic: $w_k = w_0 \left(\frac{K-k}{K}\right)^n$
- inverse parabolic: $w_k = w_0 \left(1 - \left(\frac{k}{K}\right)^n\right)$

The relaxation coefficients have dual behaviors as it was explained previously. This reveals that considering the update rule for too many iterations damage the performance of the GBP. Therefore it is more efficient to use this update for some iterations at the beginning and quickly to foster the memory. The parabolic relaxation coefficient offers the most accurate estimate of the input sequence with the most stable evolution.

In the following of the article, we deeply study this last assumptions to bring out the dynamics of the GBP with a parabolic relaxation coefficient.

IV. Dynamics

In this section, we expose some measures to evaluate the dynamical behaviors of the GBP compared with the BP. To this end, we first define the mathematical environment in which these measures are relevant.

A. State space definition and properties

To compare the dynamics of the BP and GBP, we need to define two state spaces that are similar. Using the fact that both are message-passing algorithms, we should consider the messages as the state variables. This choice fits perfectly with the convergence conditions written in the previous section for the GBP and written down here for the BP:

$$\forall e_{ij},\ n_{ji}^{(k)} = n_{ji}^{(k-1)} \tag{10}$$

The update equations of the BP can be implicitly noted:

$$\forall e_{ij}, \quad m_{ij}^{(k)} = f_{ij}\left(\{n_{yx}^{(k-1)}\}_{(x,y)}, l_i\right) \tag{11}$$

$$n_{ji}^{(k)} = g_{ji}\left(\{m_{xy}^{(k)}\}_{(x,y)}\right) \tag{12}$$

The messages $\{n_{ji}^{(k)}\}$ are then computed by a composed function $g \circ f$ that maps the messages $\{n_{ji}^{(k-1)}\}$ and the likelihoods. Thus the BP can be described by a unic equation:

$$\forall e_{ij}, \quad n_{ji}^{(k)} = G_{ij}(\{n_{nm}^{(k-1)}\}_{(m,n)}, \{l_i\}_i) \tag{13}$$

We consider respectively $\{G_{ij}\}_{(i,j)}$ and $\{F_{rq}\}_{(r,q)}$ as two sets of iterated maps on the state variables $\{n_{ji}^{(k)}\}_{(i,j)}$ and $\{m_{rq}^{(k)}\}_{(r,q)}$ which are called trajectories in the associated state spaces, denoted $\mathcal{E}_{BP}$ and $\mathcal{E}_{GBP}$, whose dimensions are the number of messages to compute at each iteration. We denote by $U^{(k)} = \{n_{ji}^{(k)}\}_{(i,j)}$ and by $V^{(k)} = \{m_{rq}^{(k)}\}_{(r,q)}$ the points of the trajectories of the BP and the GBP at iteration k.

B. Parameters and scaling

In [14] the value of σ^2, or the corresponding Signal to Noise Ratio (SNR), is used as a parameter such that different values imply different motions of the BP. However, most of their simulations are done for particular noise realizations and scaled on the SNR, that prevents from evaluating a statistical behavior. A reason is that the noise realizations that lead the BP not to converge or to converge to a wrong estimate are rare events, essentially because the LDPC codes and the iterative algorithms are created to this end. A way to have some statistical evaluations of the behavior of the BP and the GBP is the following:

1) find some of these noise realizations,
2) store the corresponding initializations on the state variables,
3) average the quantities to measure for a sufficient set of initializations that are close in the state space in the sense of the Euclidean distance.

By this way, we can target the guenine critical values. For all the measures presented in the following of the paper, we use this method to get statistical results which are relevant enough to describe the behavior of the BP and the GBP. We call critical values the SNR that correspond to a radical change in the behavior of the motion of the algorithms, which are the bifurcations.

C. Bifurcation diagram

A relevant method to extract the critical values of the SNR is the bifurcation diagram computation. For a given noise realization, it consists in evaluating the value of a function E that is computed from the state variables at their steady state for J different values of the SNR. We get a sequence $[E_{\sigma_1}, \ldots, E_{\sigma_J}]$ that represents the behavior of the dynamic system in terms of the SNR.

We consider the following function exposed in [14] called the mean square beliefs:

$$\forall \sigma, \quad E_\sigma = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} b_i^2(0)} \tag{14}$$

where the input sequence in the channel is the null sequence and the beliefs are computed at the last iteration K of the considered algorithm, BP or GBP. Obviously, there is no reason that the associated dynamic system has reached any steady state at this iteration but we need to suppose it for computation time's sake. This function presents three important values:

- $E_\sigma = 1$: all the beliefs indicate that the ouput sequence is the null sequence which is a good decoding,
- $E_\sigma = \frac{1}{4}$: all the beliefs are uniform distributions thus there is no information about the true state of the transmitted bits, which is a missed decoding,
- $E_\sigma = 0$: all the beliefs indicate that the output sequence is the complementary of the sent sequence, which is a completely wrong decoding.

The display of $[E_{\sigma_1}, \ldots, E_{\sigma_J}]$ enables to know two properties of the used algorithm: the amplitudes provides information about the decoding performance, and the variation between successive values gives us the critical values of the SNR. We display on the figure 8 the mean bifurcation diagrams of the BP and the GBP for a given noise realization computed as we exposed at the beginning of the current subsection.

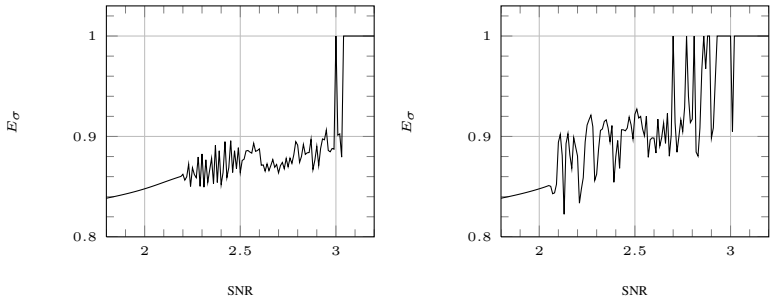


Fig. 8. Bifurcation diagrams of the BP and the GBP

We observe that for SNRs lower than 2.19 dB, the BP follows a regular increasing steady motion. Such a behavior is analoguous to that of the GBP except that the critical value is 2.07 dB. When the SNR is greater than these two critical values, the algorithms follow two distinct motions. The BP

seems to oscillate while the SNR is lower than 2.49 dB, for values in $[2.5$ dB$, 2.98$ dB$]$ however it does not appear any known evolution which is an indication of chaos. Concerning the GBP, it appears globally three intervals: in $[2.08$ dB$, 2.43$ dB$]$ the shape of E_σ indicates some irregular oscillations whereas for SNRs in $[2.44$ dB,2.69 dB$]$ we cannot assert anything except that the chaos would appear. From 2.70 dB up to 3.02 dB, the GBP tends to the right decoding state whereas the BP does not present this behavior before 2.99 dB. Moreover the shapes of the two whole signals indicate that the GBP function E_σ is globally beyond the BP one, that shows that the GBP tends faster than the BP to the right state.

D. Reduced trajectory

Another use of the mean square beliefs function is the representation of the trajectory in a 3-dimensional state space. To this end, we use the phase space reconstruction introduced in [15]. The method is first to compute this function at each iteration to get the following sequence $\mathbf{E}_\sigma = [E_\sigma(k)]_{0\leq k\leq K}$. Afterthat we share this one dimensional sequence in a three dimensional sequence as follows:

$$\tilde{\mathbf{E}}_\sigma = \begin{bmatrix} E_\sigma(0) & E_\sigma(1) & E_\sigma(2) \\ \vdots & \vdots & \vdots \\ E_\sigma(K-2) & E_\sigma(K-1) & E_\sigma(K) \end{bmatrix} \tag{15}$$

On the figures 9 and 10 are displayed some reduced trajectories of the BP for typical values of the SNR deduced from the previous bifurcation diagram. It appears for the BP four

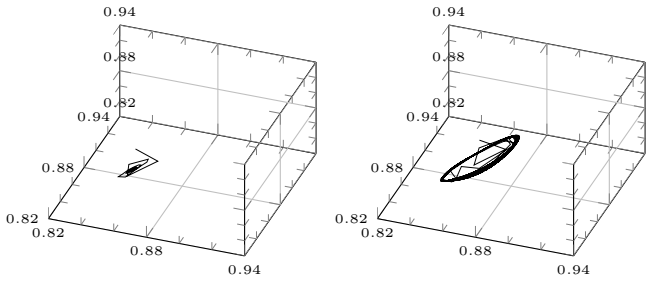


Fig. 9. Reduced trajectory for the BP on the Tanner code with SNR $= 2.15$ dB and 2.30 dB

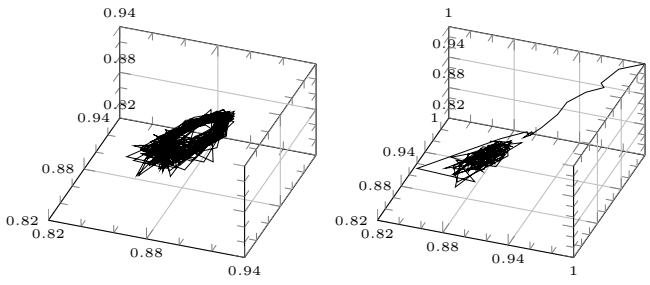


Fig. 10. Reduced trajectory for the BP on the Tanner code with SNR $= 2.70$ dB and 3.00 dB

typical behaviors that match with the four intervals exposed in the previous paragraph. We obtain a very small sized attractor for SNR $= 2.10$ dB that can be considered as a fixed point, whereas the reduced trajectory transforms to a limit cycle when the SNR is between 2.19 dB and 2.49 dB. A crucial point is that 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, that can be understood as a sequence of period doubling bifurcations in terms of dynamic system, as is displayed on the figure 11 with two interleft cycles. Such a phenomenon is a typical route to chaos, that is observable from 2.51 dB. A chaotic motion means that there is not any periodic motion 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 that is a true fixed point.

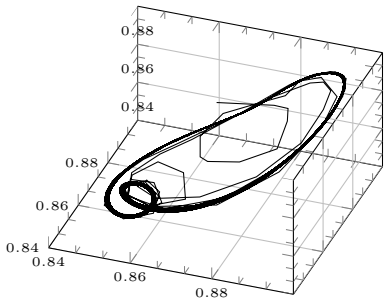


Fig. 11. Reduced trajectory for the BP on the Tanner code with SNR $= 2.40$ dB

Concerning the GBP, whose reduced trajectories are displayed on the figures 12 and 13, we cannot share the SNR values so accurately because the reduced trajectory does not transform blatantly. However, the reduced trajectory makes reveal also four different behaviors that follow the same order than that of the BP: small attractor, limit cycle, chaos, fixed point. The corresponding SNR intervals match with what have been revealed previously.

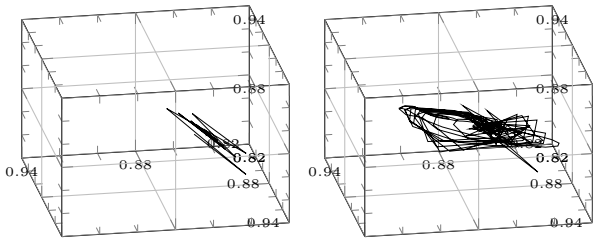


Fig. 12. Reduced trajectory for the GBP on the Tanner code with SNR $= 2.00$ dB and 2.20 dB

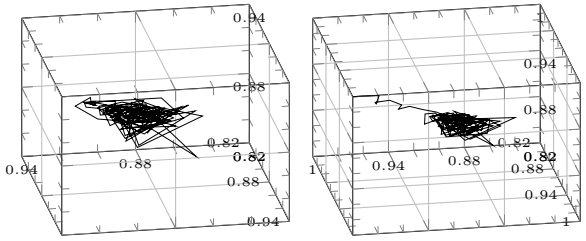


Fig. 13. Reduced trajectory for the GBP on the Tanner code with SNR $= 2.80$ dB and 3.00 dB

We have to be cautious because $\tilde{\mathbf{E}}_\sigma$ is not a true trajectory, it does not respect the Cauchy-Lipschitz condition [15] due to the non bijection between the messages and the beliefs. Thus, this sequence only has the role of giving clues about the true behavior of the considered algorithm as the possible shape of the actual trajectory in $\mathcal{E}_{BP}$ or $\mathcal{E}_{GBP}$ that are: convergence to a fixed point, convergence to a limit cycle, convergence to a chaotic attractor. To distinguish these shapes, we need a criterion that reflects the behavior by its own value. A good candidate is the Lyapunov exponent.

E. Lyapunov exponents

A common measure is the Lyapunov exponent λ [16], [15], [17]. Its computation consists in evaluating at each iteration $k \leq K$ the Euclidean distance d_k between two initially close trajectories, and computing the log-ratio:

$$\lambda = \ln \frac{d_K}{d_0} \tag{16}$$

The sign of the Lyapunov exponent λ reveals the behavior of the system around the corresponding initialization of the trajectories.

- if λ is positive then the trajectories have moved away one from the other, which is an evidence of a chaotic behavior,
- if λ is negative then 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 $\lambda \to \infty$.

When λ crosses the x-axis the system suffers from a bifurcation meaning that the algorithm has changed of motion. The corresponding SNR are the critical values. We display on the figure 14 the Lyapunov exponents of the BP and the GBP on the Tanner code.

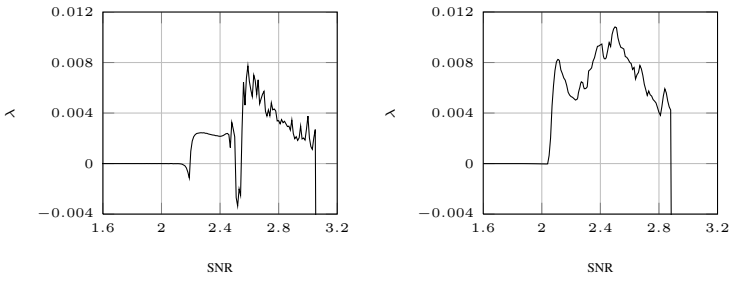


Fig. 14. Lyapunov exponents of the BP and GBP on the Tanner scaled on the SNR

As we have observed on the bifurcation diagrams, the evolutions are really different as soon as the SNR is greater than 2.07 dB. The BP curves is perfectly consistent with the associated bifurcation diagram in the sense that the critical values we extract are the same and the behaviors we could imagine by the bifurcation diagram are also revealed by the Lyapunov exponent. Concerning the GBP, we observe that the Lyapunov exponent does not follow blatantly different motions as we noticed about the reduced trajectory. Moreover, it reaches greater values than the BP, which is not in its favour. However, a crucial point is that the SNR that corresponds to the fall of the Lyapunov exponent is clearly lower for the GBP than for the BP. In association with the results about the error correction power in the previous section, it appears that from this SNR of 2.88 dB the GBP presents a stable evolution and better skills to correct the errors on the transmitted bits.

A relevant analyze we need to effect is the comparison with the reduced trajectory we exposed previously so as to associate accurately with each reduced motion a particular evolution of λ. Here are these associations for the BP:

- SNR $\in [0\text{dB}; 2.19\text{db}]$: the reduced trajectory converges to a very small sized volume of $\mathcal{E}_{BP}$ that we can consider as a fixed point whereas λ is close to the null value,
- SNR $\in [2.20\text{db}; 2.49\text{db}]$: the reduced trajectory is trapped into a limit cycle whereas λ has gone over a stair,
- SNR $\in [2.50\text{db}; 2.98\text{db}]$: the reduced trajectory does not converge to any fixed point, limit cycle or quasi-limit cycle but to a chaotic attractor whereas λ soars to high values,
- SNR $\in [2.99\text{db}; +\infty\text{db}]$: the reduced trajectory converges to a fixed point corresponding to a good decoding.

As we mentionned previously, the GBP does not present behaviors that are easy to distinguish therefore it would not be relevant to associate any possible critical value between the bifurcation diagram, the reduced trajectory and the Lyapunov exponent.

F. Hyperspheres method

We propose here a novel method to evaluate the unstability of the BP and the GBP, based on their own trajectory in $\mathcal{E}_{BP}$ and $\mathcal{E}_{GBP}$. This method is complementary to the Lyapunov exponent measure because it reveals the size of the attractor that the trajectory falls into and some other properties about the limit cycles.

This method consists in computing the rays R_k of the hyperspheres circumscribe to the trajectory inside a given temporal window centered around each point $U^{(k)}$ (or $V^{(k)}$) of the trajectory. On the figures 15 and 16 are displayed the evolution of two rays that correspond to two initially close trajectories in the Euclidean sense. The motion we observe

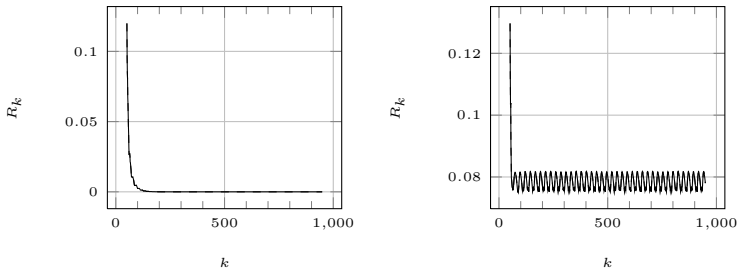


Fig. 15. Evolution of two hyperspheres rays corresponding to two initially close trajectories of the BP on the Tanner code at SNR $=$ 2.10 dB, SNR $=$ 2.30 dB

for SNR $=$ 2.30 dB is consistent with the limit cycle we observed on the reduced trajectory. The curve of the ray enable to estimate the period of the trajectory around 23 iterations. Moreover we can assert that this limit cycle is stable because

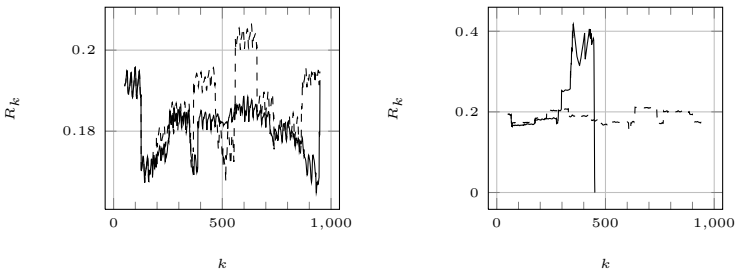


Fig. 16. Evolution of two hyperspheres rays corresponding to two initially close trajectories of the BP on the Tanner code at SNR = 2.70 dB, SNR = 3.00 dB

the two rays cannot be distinguished. For 2.70 dB the rays moved away one from the other as it was predicted by the Lyapunov exponent observations. More accurately we can see that the rays has different oscillations step. This is due to the period doubling bifurcations explained previously. During 92 iterations in average for $k \leq 500$, the trajectory is trapped in a given limit cycle and for the next 92 iterations the trajectory falls into another limit cycle of different ray. For both it is possible to measure the period or pseudo-period that is the same as the period of the first limit cycle, that is 23 iterations. For $k \geq 500$ we cannot really distinguish these different phases of evolution because the period doubling has led to the chaos. Such an observation makes our method relevant to bring out crucial information by a one dimensional function. Another important aspect of the hyperspheres method is the raising of the behaviors difference between two initially close trajectories: we easily observe that the evolution of the rays cannot be distinguished while $k \leq 200$ but as k is getting greater, the evolution of the rays move away one from the other but they follow the same kind of motion. For both of them the hypervolumes of the state space in which they are locking in are quite of the same size. When the SNR reaches the last critical value we observe that one of the rays decreases to the null value because the BP has converged to a fixed point. The other ray has not collapsed yet because the SNR is just at the critical value, if it was increased a little we would see the two rays going to zero.

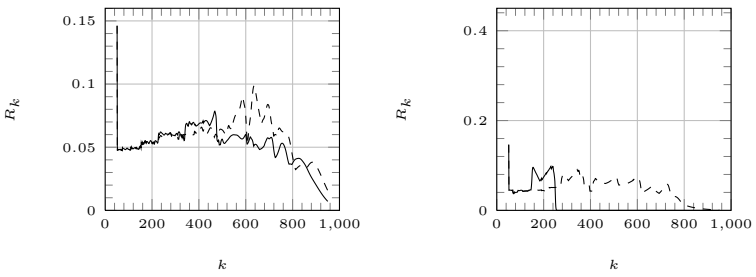


Fig. 17. Evolution of two hyperspheres rays corresponding to two initially close trajectories of the GBP on the Tanner code at SNR = 2.70 dB, SNR = 3.00 dB

Concerning the GBP, for low SNR values we did not display the evolutions of R_k because it follows the same shape as the BP one. On the figure 17, at SNR = 2.70 dB we observe that the rays of the couple of trajectories are quite smaller than that of the BP. It appears that the maximum value of the rays is still lower than the minimum value of the BP ray. The comparison between both rays on the GBP enables to deduce that even though each trajectory eventually reaches a particular attractor of the state space, their size are not very different and they collapse at the end, which is not true about the BP, making the GBP more stable than the BP for the suited construction we have proposed.

V. Conclusion

In this paper, we have presented a novel method of construction of the region-graph for the GBP algorithm. The results have shown firstly that the error correction power was improved compared with the BP one, and secondly that the GBP was more stable than the BP. The measure of the hyperspheres we have introduced enable to bring out this property and also the fact that the GBP converges towards small sized attractors contrary to the BP. Finally, this work leads to conclude that a suited region-graph construction enables the GBP to be a good candidate to surpass the famous BP. In future works, we propose to find a mathematical criterion to evaluate the relevancy of any region-graph according to the associated GBP performance, and also we propose other handlings of the TS$(5, 3)$ to build region-graphs that could present better correction power and improved stability.

Acknowledgment

Supported by the French ANR Defis program under contract ANR-08-EMER-003 (COCQ project).

References

[1] R.G. Gallager, "Low-Density Parity-Check Codes," Ph.D. dissertation, 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*, vol. 47, 2001, NO. 2.

[3] J. Pearl, *Probablistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[4] T.J. Richardson, "Error floors of LDPC codes," in *Proc. 41st Annual Allerton Conf. on Comm. Cont. and Comp.*, 2003.

[5] 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*, vol. 47, 2001, NO. 2.

[6] T.J. Richardson and R.L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. on Inf. Theory*, vol. 47, 2001, N0. 2.

[7] K.P. Murphy, Y. Weiss and M.I. Jordan, "Loopy belef propagation for approximate inference: an empirical study," in *Proc. Uncertainty in AI*, 1999.

[8] J.S. Yedidia, W.T. Freeman and Y. Weiss, "Constructing free energy approximations and Generalized Belief Propagation algorithms," *IEEE Trans. on Inf. Theory*, vol. 51, pp. 2282–2313, 2004.

[9] P. Pakzad et V. Anantharam, "Estimation and marginalization using Kikuchi approximation methods," *Neural Computation*, vol. 17, pp. 1836–1873, 2003.

[10] H.A. Bethe, "Statistical Theory of Superlattices," in *Proc. Roy. Soc. London*, 1935.

[11] R. Kikuchi, "A Theory of Cooperative Phenomena," *Phys. Rev*, vol. 81, pp. 988–1003, 1951.

[12] R.M. Tanner, R. Michael, D. Sridhara and T. Fuja, "A Class of Group-Structured LDPC Codes," 2001.

[13] S. Sankaranarayanan, S. K. Chilappagari, R. Radhakrishnan and B. Vasic, "Failures of the Gallager B Decoder: Analysis and Applications," in *ITA Workshop UCSD*, 2006.

[14] 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*, vol. 16, pp. 3435–3449, 2005.
[15] R. Hilborn, *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*. Oxford University Press, 2000.
[16] 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*, vol. 65, pp. 117–134, 1993.
[17] A. Wolf, J.B. Swift, H.L. Swinney and J.A. Vastano, "Determining Lyapunov Exponents from a Time Series," *Physica*, pp. 285–317, 1985.