On robust stability of the Belief Propagation Algorithm for LDPC decoding

Björn S. Rüffer, Peter M. Dower, Christopher M. Kellett, and Steven R. Weller

***Abstract*— The exact nonlinear loop gain of the belief propagation algorithm (BPA) in its log-likelihood ratio (LLR) formulation is computed. The nonlinear gains for regular low-density parity-check (LDPC) error correcting codes can be computed exactly using a simple formula. It is shown that in some neighborhood of the origin this gain is actually much smaller than the identity. Using a small-gain argument, this implies that the BPA is in fact locally input-to-state stable and produces bounded outputs for small-in-norm input LLR vectors. In a larger domain the algorithm produces at least bounded trajectories. Further it is shown that, as the block length increases, these regions exponentially shrink.**

***Index Terms*— iterative decoding; LDPC codes; dynamical system; convergence; belief propagation; small-gain theorem.**

I. Introduction

It is known that the belief propagation low-density parity-check (LDPC) decoder as well as its turbo decoding counterparts might not converge or might converge to so-called pseudo-codewords [1, 4, 6, 8, 12, 15, 16, 22, 23]. Identifying such pseudo-codewords and giving conditions for convergence is an active field of research [13, 22].

When communicating over the additive white Gaussian noise channel (AWGNC) using LDPC codes, the belief propagation algorithm (BPA, also known as the sum-product algorithm or the loopy belief propagation algorithm) approximates the maximum *a posteriori* probability (MAP), which is often expressed in the log-likelihood ratio (LLR) domain. As such, it would be desirable if BPA would return one finite LLR vector, or converge to one such LLR vector, for any given finite input LLR vector. This is certainly not the case: Due to cycles in the parity-check matrix, the LLR for individual bits can, and most often does, increase — in magnitude— *ad infinitum* with the number of iterations. On the other hand, for the AWGNC the probability that an LLR input for a given bit is $\pm\infty$ is zero, hence a finite LLR vector as input to BPA is a reasonable assumption. Here, we consider convergence of LLR vectors, not "convergence to codewords", which might be easily confused.

Iterative decoding can be considered as a discrete-time dynamical system. There exist stochastic approaches (e.g., [4] for the related turbo decoding) as well as deterministic approaches [12, 23] to model and analyze the performance and dynamics of iterative decoders, usually by considering some low dimensional system that describes the average behavior for a particular ensemble of codes.

Also it is known that for so-called ensembles of codes average performance bounds can be calculated and are often exact [17]. These are known as stability results and again are obtained by a fixed point analysis of a one-dimensional approximation of the large-scale system describing the evolution of the decoding error.

A full-order dynamical system representation of the belief propagation algorithm has been considered in [18], where convergence has been studied in the special case when the dynamics is linear, which corresponds to so-called repeat codes.

Contraction conditions for convergence analysis of the belief propagation algorithm have been considered in [15]. Such conditions depend on the particular factor graph, or, in the case of the BPA decoder, of the particular form of the parity-check matrix. Unfortunately, in general, BPA decoding of LDPC codes is not a contraction, as is evident by the existence of non-convergent counterexamples and also follows from existing bifurcation analysis [10–12, 23].

Despite this, approaches to overcome the obstacles of non-convergence do exist (notably, this usually means "convergence to codewords" unlike in this paper). One such approach is to introduce damping into BPA or related algorithms. E.g., in [3] it is shown that the max-product algorithm —if attenuated and skewness-compensated— converges to the MAP codeword if it converges at all. Based on a linearization of the message-update rule, factor graphs containing at most pairwise interactions are considered in [14]. This version of "dampened" BPA is a modified message-update rule, which amounts to a convex combination of the old and the new message. There it is observed that this form of damping can improve performance in the purely anti-ferromagnetic case while for spin-glass interactions it helps only slightly. Also in [7] it is mentioned that damping would improve convergence, but this is rather vague.

In contrast, here we compute the exact nonlinear loop gain of the BPA in its LLR formulation. Gain analysis is in essence a simplification of a contraction analysis. We will see that in some neighborhood of the origin this gain is actually much smaller than the identity. Using a small-gain argument, this implies that BPA is in fact locally input-to-state stable and produces bounded outputs for small-in-norm input LLR vectors. We compute the probability for a given block length that an LLR vector received from AWGNC falls in this region of small-in-norm inputs. We then show that this probability

The first two authors are with the Department of Electrical & Electronic Engineering, University of Melbourne, Parkville, Victoria 3010, Australia, brueffer@unimelb.edu.au (bjoern@rueffer.info), P.Dower@ee.unimelb.edu.au

The other authors are with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia, {Chris.Kellett, Steven.Weller}@newcastle.edu.au

B. S. Rüffer and P. M. Dower have been partially supported by ARC Grant DP0880494 and AFOSR Grant FA2386-09-1-4111. C. M. Kellett, and S. R. Weller have received support from the Australian Research Council (ARC) under grant DP0771131.

decreases exponentially with increasing block length.

This paper is organized as follows: In Section II we recall the dynamical system formulation of BPA for decoding LDPC codes communicated over an AWGNC. In Section III we first introduce some more necessary notation and then recall convergence conditions based on contractions of system operators for discrete-time dynamical systems. Then we compute the loop gain of the system operator, which can be used for a simplified analysis. We explicitly state the formula for the nonlinear system gain for a given parity-check matrix and show that, in most cases, this gain turns out to be less than the identity in a neighborhood of the origin, and to be larger than the identity away from the origin. We recall the concept of local input-to-state stability and develop a stability theorem for BPA based on a small-gain theorem. Finally we compute the probability that inputs to the BPA actually fall in the region where the loop gain is small. Section IV concludes the paper.

II. State-space representation of BPA

The belief propagation algorithm for the decoding of an LDPC code is completely characterized by the parity check matrix of this code. Starting from a given parity-check matrix $H \in \mathbb{F}_2^{m\times n}$ we cast this algorithm into a state-space representation that is amenable to a systems theoretic-type analysis as in [18]. Here we briefly recall the basic steps.

The LLR formulation of the belief propagation algorithm under consideration is as follows. Initially, the a-priori probabilities for every bit are computed and passed as LLR messages from bit- to check-nodes, denoted by $l_{\mu^0_{x_i\to h_j}}$. Then the following two update rules are iterated. First, update the factor-to-variable messages according to

$$l_{\mu_{h_j\to x_i}} = 2\,\mathrm{arctanh} \prod_{\substack{k\neq i:\\ x_k \text{ adjacent to } h_j}} \left(\tanh \frac{l_{\mu_{x_k\to h_j}}}{2}\right). \tag{1}$$

Then update the variable-to-factor messages as

$$l_{\mu_{x_i\to h_j}} = l_{\mu^0_{x_i\to h_j}} + \sum_{l\neq j} l_{\mu_{h_l\to x_i}}\,, \tag{2}$$

and the intermediate marginalization, or per bit a-posteriori probabilities, after every iteration are given by

$$l_i = l_{\mu^0_{x_i\to h_j}} + \sum_{l} l_{\mu_{h_l\to x_i}}\,, \tag{3}$$

where l_μ denotes the LLR value of a particular message, see [18] for details.

This algorithm can be cast into a state-space form by enumerating the messages in each iteration and defining operators mapping variable-to-check messages to check-to-variable messages and *vice versa*.

Since the messages are sent along the edges in the Tanner graph, we effectively enumerate the edges in the Tanner graph. For a graph with q edges, there are $1\cdot 2\cdot \ldots \cdot q = q!$ possible ways to perform this enumeration all of which qualitatively lead to the same state-space representation, up to a renumbering of the states. We pick a canonical enumeration: As the edges in the Tanner graph correspond to non-zero entries in the parity-check matrix H, we can enumerate these entries from top to bottom, row by row, as in the following example taken from [18]:

$$H = \begin{bmatrix} 1^1 & 1^2 & 0 & 0 \\ 0 & 1^3 & 1^4 & 0 \\ 0 & 0 & 1^5 & 1^6 \end{bmatrix}. \tag{4}$$

This enumeration gives us a mapping $\mathfrak{n}(k) = \mathfrak{n}_H(k) = (\mathfrak{n}_1(k), \mathfrak{n}_2(k))$ denoting the coordinates $(i,j) \in \{1,\ldots,m\} \times \{1,\ldots,n\}$ of the kth non-zero entry in $H \in \mathbb{F}_2^{m\times n}$, for $k \in \{1,\ldots,q\}$, where q is the total number of nonzero entries in H.

The messages $l_{\mu_{x_i\to h_j}}$ and $l_{\mu_{h_j\to x_i}}$ become the states of the dynamical system to be defined, and the enumeration $\mathfrak{n}$ provides for the one-to-one correspondence between messages and states via

$$l_{\mu_{x_{\mathfrak{n}_2(k)}\to h_{\mathfrak{n}_1(k)}}} =: x_1^k \quad \text{and} \quad l_{\mu_{h_{\mathfrak{n}_1(k)}\to x_{\mathfrak{n}_2(k)}}} =: x_2^k\,,$$

where, with a slight abuse of notation, we have denoted states of the dynamical system with $x_1, x_2 \in \mathbb{R}^q$. Now the message update rules can be cast into the following operator equations,

$$\begin{aligned} x_1^+ &= Px_2 + Bu \\ x_2^+ &= S(x_1) \\ y &= B^T x_2 + u\,, \end{aligned} \tag{5}$$

where u is the input LLR vector, i.e., the LLRs obtained from the channel, corresponding to a-priori probabilities for each bit, and y is the vector of output LLRs, i.e., the LLR vector corresponding to intermediate estimates of the conditional a-posteriori probabilities. The structure of the dynamical system (5) is depicted in Figure 1. The definition of all operators in (5) follows.

As all operators except the operator S are linear, these can be thought of as matrix-vector multiplications. In particular, define the matrix $B = B_H = (b_{ij}) \in \mathbb{R}^{q\times n}$ by

$$b_{ij} = \begin{cases} 1 & \text{if } \mathfrak{n}_2(i) = j \\ 0 & \text{otherwise}. \end{cases} \tag{6}$$

and the map $P = P_H : \mathbb{R}^q \to \mathbb{R}^q$ by

$$P_i(\xi) = \sum_{j\neq i:\, \mathfrak{n}_2(j) = \mathfrak{n}_2(i)} \xi_j, \tag{7}$$

for $\xi \in \mathbb{R}^q$. This operator can be represented as matrix-vector multiplication via $P(\xi) = P\xi$, $\xi \in \mathbb{R}^q$, where the matrix $P = P_H = (p_{ij}) \in \mathbb{R}^{q\times q}$ is defined by

$$p_{ij} = \begin{cases} 1 & \text{if } j \neq i,\, \mathfrak{n}_2(i) = \mathfrak{n}_2(j) \\ 0 & \text{otherwise}. \end{cases} \tag{8}$$

The nonlinear operator $S = S_H : \mathbb{R}^q \to \mathbb{R}^q$ is given explicitly by

$$S_i(\xi) = 2\,\mathrm{arctanh}\left(\prod_{j\neq i:\, \mathfrak{n}_1(j) = \mathfrak{n}_1(i)} \tanh \frac{\xi_j}{2} \right), \tag{9}$$

for $\xi \in \mathbb{R}^q$.

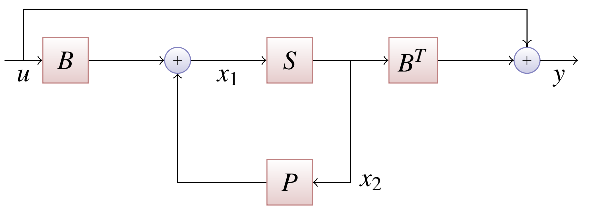

Fig. 1. Block diagram of the BP algorithm cast as a dynamical system. The only nonlinear component is the operator S.

In [18] it has been shown that in case of repeat codes the dynamics of (5) are linear, i.e., the operator S becomes linear. In this case the Tanner graph is a tree, and convergence of BPA is assured. This can also be seen by rewriting the (now linear) system (5) in the form $z^+ = Az + \overline{B}u$ where $A = \begin{bmatrix} 0 & P \\ S & 0 \end{bmatrix}$, $z^T = (x_1^T, x_2^T)$ and $\overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$ and observing that the spectrum $\sigma(A)$ is contained in the open unit disc in $\mathbb{C}$. In fact, the matrix A turns out to be nilpotent [18], guaranteeing finite-time convergence.

III. Main results

A. Preliminaries

Depending on context, $\|\cdot\|$ denotes the max-norm or the induced operator norm. For functions $f : \mathbb{R}^p \to \mathbb{R}^q$ and $g : \mathbb{R}^q \to \mathbb{R}^r$ we denote by $g \circ f$ the function $x \mapsto g(f(x))$. For a function $x : \mathbb{N} \to \mathbb{R}^q$ by $\|x(\cdot)\|_\infty$ we denote $\sup_{k\geq 0} \|x(k)\|$.

Recall that the *comparison function* classes $\mathscr{K}$ and $\mathscr{K}_\infty$ are, respectively, the sets of continuous functions $\{\gamma : \mathbb{R}_+ \to \mathbb{R}_+,\ \gamma(0) = 0,\ \gamma \text{ is strictly increasing}\}$ and $\{\gamma \in \mathscr{K} : \gamma \text{ is unbounded}\}$. For convenience we write class $\mathscr{G} = \mathscr{K} \cup \{0\}$ to include the zero function. The class of continuous positive definite functions $\alpha : \mathbb{R}_+ \to \mathbb{R}_+$ is denoted by $\mathscr{P}\mathscr{D}$. A function $\beta : \mathbb{R}_+^2 \to \mathbb{R}_+$ is of class $\mathscr{K}\mathscr{L}$ if for fixed $t \geq 0$ the function $\beta(\cdot,t)$ is of class $\mathscr{K}$ and for fixed $s \geq 0$ the function $\beta(s,\cdot)$ is non-increasing with $\lim_{t\to\infty} \beta(s,t) = 0$.

B. Conditions for convergence

It has been recognized in [15] that one sufficient condition for convergence of BPA is that the message-updating mapping constitutes a contraction. A contraction mapping is a mapping $f : X \to X$ on a complete metric space (X,d) satisfying

$$d(f(x), f(y)) \leq k\,(d(x,y)) \tag{10}$$

where $k : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ satisfies $k < \text{id}$. In the case where the function $k(\cdot)$ is linear, (10) reduces to $d(f(x), f(y)) \leq k \cdot d(x,y)$ for some $0 \leq k < 1$, cf. [19]. The Banach Contraction Principle states that any contraction mapping has a unique fixed point. Note again that such a fixed point lies in the LLR domain, not in the codeword domain.

The following theorem states that if BPA acts as a contraction it does so independently of the input. In particular, if BPA constitutes a contraction, this fact is independent of the signal to noise ratio (SNR) of the channel.

Theorem 3.1: For any given parity-check matrix, the Belief Propagation algorithm is a contraction mapping if and only if the $P \circ S$ loop constitutes a contraction.

Proof: Let us rewrite the x-dynamics in (5) as

$$x^+ = T(x) + \overline{B}u \tag{11}$$

with $x = (x_1^T, x_2^T)^T$,

$$T(x) = (P(x_2)^T, S(x_1)^T)^T, \tag{12}$$

and $\overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$. Now let $f(x) := f_u(x) := T(x) + \overline{B}u$ and observe that the dynamics are affine in u, i.e., for any $x^1, x^2 \in \mathbb{R}^{2q}$,

$$\begin{aligned} d(f(x^1), f(x^2)) &:= \|f(x^1) - f(x^2)\| \\ &= \|T(x^1) + \overline{B}u - T(x^2) - \overline{B}u\| \\ &= \|T(x^1) - T(x^2)\| \\ &\leq k\|x^1 - x^2\| \end{aligned} \tag{13}$$

for some Lipschitz constant $k > 0$, possibly depending on the compact set containing x_1 and x_2. While this constant will, in general, be greater than one, we observe that the Lipschitz constant does not depend on u. ■

Interestingly, since the dynamical system (5) does not represent a physical system but a computer algorithm, we can modify T to some $\tilde{T}$ such that the constant in (13) becomes less than one for that operator $\tilde{T}$. For example, one could introduce a damping such that the cycle gain becomes small. In general, however, such a simple approach does not necessarily improve the performance of BPA. For example, since BPA computes the exact MAP on trees already, any dampened version would not be exact on trees. Our simulations further suggest that BPA with T replaced by $\tilde{T} = c \cdot T$ for some appropriately small $0 < c < 1$ generally deteriorates decoding performance.

Instead of considering T as a mapping on a high dimensional space, it is often more convenient to project the concept of interest down to lower dimensions to ease analysis. That is roughly the idea behind the next section.

C. Computing gains

By the *input-output gain* or simply *gain* of a map $M : \mathbb{R}^p \to \mathbb{R}^q$ we denote the non-negative function $g : \mathbb{R}_+ \to \mathbb{R}_+$ given by

$$g(r) = \sup\left\{\|M(x)\| : x \in \mathbb{R}^p,\ \|x\| = r\right\}.$$

It is important to highlight that this definition depends on the choice of norms. In this paper the max-norm is used.

The mapping that assigns MAP LLR vectors to input a-priori LLR vectors obviously has finite gain. The output of BPA in the presence of cycles in the factor graph, however, does not necessarily converge, and individual LLR values may increase unboundedly in magnitude over iterations. As such, BPA as a mapping from input LLR vectors to sets of output LLR vectors does not necessarily have finite gain. The magnitude of an LLR can be interpreted as knowledge or certainty about the value a particular bit has. The higher the magnitude of the LLR, the more certain we can be about the value of the bit. It is hence of interest to understand how this certainty changes when BPA performs its iterations.

To compute the gain of the feedback loop in the dynamical system (5), let us first consider individual input-output gains of single nodes. For $i \in \{1,\dots,q\}$ denoting the number of an edge in the factor graph of a fixed parity-check matrix H, let $c(i)$ denote the degree of the adjacent check-node and $v(i)$ the degree of the adjacent variable node.

For a given check-node with degree c it is not difficult to see that its input-output gain with respect to the max-norm is given by

$$g_c(r) = 2\operatorname{arctanh}\left(\left(\tanh\frac{r}{2}\right)^{c-1}\right), \quad r \geq 0. \tag{14}$$

The gain of a variable-node of degree v is simply $v-1$. Fig. 2 shows gains for check-nodes of different degree.

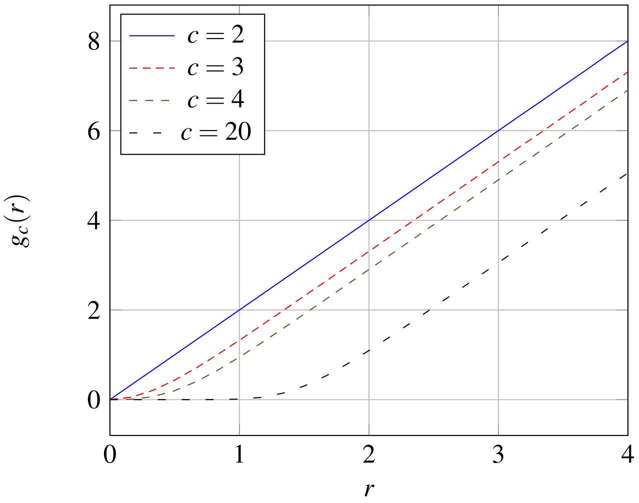

Fig. 2. Exact nonlinear gains for different check-node degrees.

Now we can combine these considerations into the following result:

Lemma 3.2: Let a parity-check matrix $H \in \mathbb{F}_2^{m\times n}$ and the associated operators (7)–(9) be given. Then the nonlinear gain $\gamma_H \in \mathscr{G}$ of the $P\circ S$-loop with respect to the max-norm is given by

$$\begin{aligned}\gamma_H(r) &:= \max_{\|x\|=r} \|(P\circ S)(x)\| \\ &= \max_{1\leq i\leq q} \sum_{\substack{j\neq i\\ \mathfrak{n}_2(j)=\mathfrak{n}_2(i)}} g_{c(j)}(r), \quad r \geq 0.\end{aligned} \tag{15}$$

Note that this upper bound is sharp. The formula implies that the gain is the same for any parity-check matrix H describing a regular code, since the check-node degree and also the bit-node degree are uniform. However, for irregular codes different choices of parity-check matrices may lead to different gains.

D. Local input-to-state stability

Loop gains and small-gain theorems, have proven quite useful in analysis and design of control systems over the past 50 years, starting out originally with linear gains and more recently with nonlinear gains, see [2] for a recent overview of the field.

Here we utilize the concept of local input-to-state stability. Roughly, a system with states and inputs, evolving in time, is called (globally) input-to-state stable provided the equilibrium is globally asymptotically stable in the absence of inputs, and, for essentially bounded inputs, a ball centered at the equilibrium with radius parametrized by a (nonlinear) function of the magnitude of the input is globally asymptotically stable, cf. [20, 21].

Definition 3.3: A system

$$x^+ = f(x,u) \tag{16}$$

is termed

- *locally input-to-state stable* (LISS), if there exist $\beta \in \mathscr{KL}$ and $\gamma \in \mathscr{K}$, $\delta_x > 0$ and $\delta_u > 0$, such that for all $\|x(0)\| \leq \delta_x$ and $\|u(\cdot)\| \leq \delta_u$,
$$\|x(k)\| \leq \beta(\|x(0)\|,k) + \gamma(\|u(\cdot)\|), \quad \forall k \geq 0; \tag{17}$$
- *locally stable* (LS) if there exist $\sigma,\gamma \in \mathscr{K}$ and $\delta_x > 0$ and $\delta_u > 0$, such that for all $\|x(0)\| \leq \delta_x$ and $\|u(\cdot)\| \leq \delta_u$,
$$\|x(k)\| \leq \sigma(\|x(0)\|) + \gamma(\|u(\cdot)\|), \quad \forall k \geq 0. \tag{18}$$

We say that system (16) has the *local asymptotic gain property* (LAG) if there exist a function $\gamma \in \mathscr{K}$, $\delta_x > 0$ and $\delta_u > 0$, such that for all $\|x(0)\| \leq \delta_x$ and $\|u(\cdot)\| \leq \delta_u$,

$$\limsup_{k\to\infty} \|x(k)\| \leq \gamma(\|u(\cdot)\|), \quad \forall k \geq 0. \tag{19}$$

The corresponding global definitions are termed ISS, GS, and AG, and are defined as in (17)–(19), respectively, but without the restrictions on initial conditions and inputs. Our main ISS result is as follows.

Theorem 3.4 (LISS small-gain theorem): Let $H \in \mathbb{F}_2^{m\times n}$ and the corresponding operators (7)–(9) be given. If the gain γ_H satisfies

$$\gamma_H(r) < r \tag{20}$$

for $r \in (0,R)$ for some $R > 0$, then the x-dynamics of (5) is locally ISS and, hence, BPA yields bounded outputs for all $\|u(\cdot)\| \leq \delta_u$.

Fig. 3 shows loop-gains γ_H for two common regular codes in comparison with the identity. Condition (20) of Theorem 3.4 is obviously satisfied.

Proof: It is known that ISS and GS+AG are in fact equivalent, see [5, 9]. By the same arguments it is plain to see that also the local variants are equivalent, i.e., LISS and LS+LAG are equivalent. Hence to prove LISS in the theorem, it is sufficient to prove that an LS and an LAG estimate hold.

Proof that the x-dynamics is LS: Due to (20) we may choose $\tilde{\delta}_x, \tilde{\delta}_u > 0$ and $\kappa \in (0,1)$, such that $\gamma_H(r) < \kappa r$ for all $r \in [0,R_0]$ with $\tilde{\delta}_x \leq R_0 \leq R$, and $\kappa\tilde{\delta}_x + \tilde{\delta}_u \leq \tilde{\delta}_x$.

Let $\delta_u := \tilde{\delta}_u/\|B\|$. Then the estimate $\|x_1(k+2)\| \leq \|PS(x_1(k)) + Bu(k+1)\| \leq \kappa\|x_1(k)\| + \|B\|\|u(k+1)\| \leq \kappa\tilde{\delta}_x + \tilde{\delta}_u$ holds for all $\|x_1(k)\| \leq \tilde{\delta}_x$ and $\|u(k+1)\| \leq \delta_u$. Inductively this yields for $\|x_1(0)\| \leq \tilde{\delta}_x$ and $\|u(\cdot)\| \leq \delta_u$,

$$\|x_1(2k)\| \leq \kappa^k\|x_1(0)\| + \sum_{l=0}^{k-1} \kappa^l \|B\|\|u(\cdot)\|, \quad \forall k \geq 0. \tag{21}$$

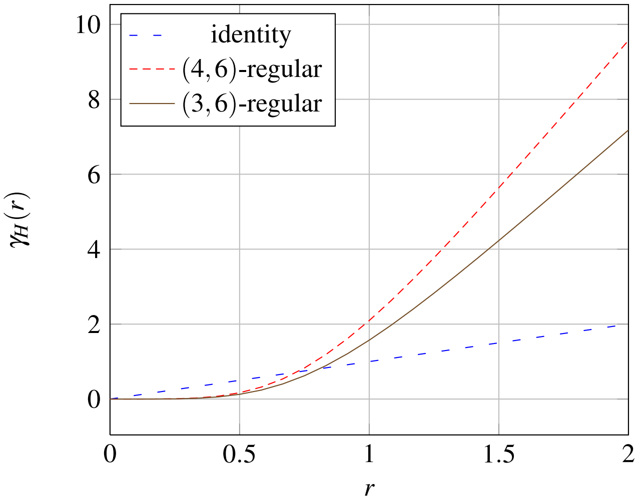

Fig. 3. The nonlinear loop-gain γ_H for (3,6)- and (4,6)-regular LDPC codes in comparison to the identity.

Analogously, we compute for $\|x_1(1)\| \le \tilde{\delta}_x$ and $\|u(\cdot)\| \le \delta_u$,

$$\|x_1(2k+1)\| \le \kappa^k \|x_1(1)\| + \sum_{l=0}^{k-1} \kappa^l \|B\| \|u(\cdot)\|, \quad \forall k \ge 0. \tag{22}$$

Using the fact that $\|S\| \le 1$, and hence $\|x_2(k+1)\| \le \|x_1(k)\|$ for all $k \ge 0$, as well as $\|x_1(1)\| \le \|P\| \|x_2(0)\|$, we obtain the following conservative estimate:

$$\|x(k)\| \le \max\{1, \|P\|\} \|x(0)\| + \frac{\|B\|}{1-\kappa} \|u\|$$

for all $k \ge 0$, $\|x(0)\| \le \delta_x := \frac{\tilde{\delta}_x}{\|P\|}$ and $\|u(\cdot)\| \le \delta_u$. This is an estimate of the form (18) proving LS.

Proof that the x-dynamics is LAG: An LAG estimate of the form (19) follows from inequalities (21), (22), and $\|x_2(k+1)\| \le \|x_1(k)\|$, as well as the fact that $0 \le \kappa < 1$. Together the LS+LAG estimates prove local input-to-state stability of the x-dynamics of (5).

Proof that outputs remain bounded: Recall that BPA always starts from $x(0) = 0$. By the first part of the theorem we have $\|x(k)\| \le \frac{\|B\|}{1-\kappa} \|u(\cdot)\|$ whenever $\|u(\cdot)\| \le \delta_u$. It follows that $\|y(k)\| = \|B^T x_2(k) + u(k)\| \le \big(\|B^T\| \frac{\|B\|}{1-\kappa} + 1\big) \|u(\cdot)\|$ for all $k \ge 0$. This implies the claim. ■

E. More on the small-in-norm input region

As we have seen the nonlinear loop gain is essentially determined by the "worst" combination of check-node/variable-node degree combination in the factor graph. For fixed degree distributions and increasing block lengths it is of interest to compute the probability that a channel measurement for a given channel actually falls within the region where our previous results apply.

We consider the AWGNC that maps $x \in \{\pm 1\}$ to $y = x + w$, where $w \in \mathcal{N}(0, \sigma^2)$. Assuming uniform and independent priors for the channel input X we have $P(X = 1) = P(X = -1) = \frac{1}{2}$. Let N be a $\mathcal{N}(0, \sigma^2)$ distributed random variable. Then the probability that $Y = X + N \in [-\varepsilon, \varepsilon]$, with $\varepsilon > 0$, is given by $P(X+N \in [-\varepsilon, \varepsilon]) = \frac{1}{2} P(N+1 \in [-\varepsilon, \varepsilon]) + \frac{1}{2} P(N-1 \in [-\varepsilon, \varepsilon]) = P(N+1 \in [-\varepsilon, \varepsilon])$ by symmetry. We obtain

$$P(X+N \in [-\varepsilon, \varepsilon]) = \int_{-\varepsilon}^{\varepsilon} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-1)^2}{2\sigma^2}} dx =: \kappa(\varepsilon, \sigma).$$

As noise acts independently on different bits of a transmitted codeword, for a multivariate channel input $\overline{X}$ of length n we have that the channel output $\overline{Y} = \overline{X} + \overline{N}$, where the components of $\overline{N}$ are i.i.d., satisfies

$$\begin{aligned} P(\overline{X} + \overline{N} \in [-\varepsilon, \varepsilon]^n) & \\ &= \prod_{i=1}^{n} P(\overline{X}_i + \overline{N}_i \in [-\varepsilon, \varepsilon]) \\ &= \prod_{i=1}^{n} \kappa(\varepsilon, \sigma) = \kappa(\varepsilon, \sigma)^n . \end{aligned}$$

Since for fixed $\varepsilon > 0$ and $\sigma > 0$ we have $\kappa(\varepsilon, \sigma) \in (0, 1)$, it follows that as $n \longrightarrow \infty$, the probability $P(\overline{X} + \overline{N} \in [-\varepsilon, \varepsilon]^n)$ tends to zero. Now taking into account the mapping from received channel symbols $y \in \mathbb{R}$ to LLR values $u \in \mathbb{R}$, which is given by

$$u = \frac{2y}{\sigma^2},$$

we thus have proved the following result.

Theorem 3.5: The probability that an input LLR vector u to BPA, obtained from a vector of channel symbols received over an AWGN channel with parameter σ, has max-norm $\|u\| < \varepsilon$ can be computed as

$$P(\|U\| < \varepsilon) = \kappa\left(\frac{\varepsilon}{2}\sigma^2, \sigma\right)^n$$

and it satisfies $P(\|U\| < \varepsilon) \longrightarrow 0$ as $n \longrightarrow \infty$.

Together with our estimate on the loop gain we obtain:

Corollary 3.6: The probability that a received vector contains little enough certainty that this certainty ultimately cannot be increased by BPA processing, decreases with block length for fixed degree distributions.

IV. Conclusions and Outlook

Using a loop-gain analysis we have shown that the loopy belief propagation algorithm is actually very well behaved, as long as it faces inputs of "low certainty," in the sense that its output in terms of LLR vectors is bounded. In particular, the algorithm does not generate values that increase in magnitude *ad infinitum* so long as the LLR inputs are initially small in magnitude.

While this low certainty assumption in general does not apply to the majority of possible inputs, it sheds new light on possible ways how the maximum a-posteriori probability estimation might be improved.

References

[1] D. Agrawal and A. Vardy, *The turbo decoding algorithm and its phase trajectories*, IEEE Transactions on Information Theory **47** (2001), no. 2, 699–722.

[2] A. L. Chen, G.-Q. Chen, and R. A. Freeman, *Stability of nonlinear feedback systems: A new small-gain theorem*, SIAM J. Control Optim. **46** (2007), no. 6, 1995–2012.

[3] B.J. Frey and R. Koetter, *Exact inference using the attenuated max-product algorithm*, Advanced mean field methods: Theory and practice, 2000.

[4] M. Fu, *Stochastic analysis of turbo decoding*, IEEE Transactions on Information Theory **51** (2005), no. 1, 81–100.

[5] K. Gao and Y. Lin, *On equivalent notions of input-to-state stability for nonlinear discrete time systems*, Proc. of the IASTED Int. Conf. on Control and Applications, 2000, pp. 81–87.

[6] T. Heskes, *On the uniqueness of loopy Belief Propagation fixed points*, Neural Computation **16** (2004), no. 11, 2379–2413.

[7] T. Heskes, K. Albers, and B. Kappen, *Approximate inference and constrained optimization*, Uncertainty in Artificial Intelligence, 2003, pp. 313–320.

[8] A. Ihler, *Accuracy bounds for Belief Propagation*, Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI), 2007.

[9] Z.-P. Jiang and Y. Wang, *Input-to-state stability for discrete-time nonlinear systems*, Automatica J. IFAC **37** (2001), no. 6, 857–869.

[10] C. M. Kellett and S. R. Weller, *Bifurcations and EXIT charts for the binary erasure channel*, Proceedings of IEEE International Symposium on Information Theory, 2006, pp. 2559–2563.

[11] ———, *Bifurcations in iterative decoding and root locus plots*, IET Control Theory and its Applications **2** (2008), no. 12, 1086–1093.

[12] L. Kocarev, F. Lehmann, G. M. Maggio, B. Scanavino, Z. Tasev, and A. Vardy, *Nonlinear dynamics of iterative decoding systems: Analysis and applications*, IEEE Transactions on Information Theory **52** (2006), no. 4, 1366–1384.

[13] R. Koetter and P.O. Vontobel, *Graph-covers and iterative decoding of finite length codes*, Proc. 3rd Intern. Conf. on Turbo Codes and Related Topics, (Brest, France), 2003, pp. 75–82.

[14] J. M. Mooij and H. J. Kappen, *On the properties of the Bethe approximation and loopy Belief Propagation on binary networks*, J. Stat. Mech. (2005). P11012.

[15] ———, *Sufficient conditions for convergence of the sum-product algorithm*, IEEE Transactions on Information Theory **53** (2007), no. 12, 4422–4437.

[16] T. Richardson, *The geometry of turbo-decoding dynamics*, IEEE Transactions on Information Theory **46** (2000), no. 1, 9–23.

[17] T. Richardson and R. Urbanke, *Modern coding theory*, Cambridge University Press, 2008.

[18] B. S. Rüffer, C. M. Kellett, P. M. Dower, and S. R. Weller, *Belief Propagation as a Dynamical System: The Linear Case and Open Problems*, IET Control Theory Appl. (2010). *to appear, accepted* November 12, 2009.

[19] S. Singh, B. Watson, and P. Srivastava, *Fixed point theory and best approximation: The KKM-map principle*, Mathematics and its Applications, vol. 424, Kluwer Academic Publishers, Dordrecht, 1997.

[20] E. D. Sontag, *The ISS philosophy as a unifying framework for stability-like behavior*, Nonlinear control in the year 2000, Vol. 2 (Paris), 2001, pp. 443–467.

[21] E. D. Sontag and Y. Wang, *Notions of input to output stability*, Systems Control Lett. **38** (1999), no. 4-5, 235–248.

[22] Y. Weiss and W.T. Freeman, *On the optimality of solutions of the max-product Belief-Propagation algorithm in arbitrary graphs*, IEEE Transactions on Information Theory **47** (2001), no. 2, 736–744.

[23] X. Zheng, F. C. M. Lau, C. K. Tse, and S. C. Wong, *Study of bifurcation behavior of LDPC decoders*, Internat. J. Bifur. Chaos Appl. Sci. Engrg. **16** (2006), no. 11, 3435–3449.