Turbo Decoding as Iterative Constrained Maximum-Likelihood Sequence Detection

John MacLaren Walsh, *Member, IEEE*, Phillip A. Regalia, *Fellow, IEEE*, and C. Richard Johnson, Jr., *Fellow, IEEE*

***Abstract*—The turbo decoder was not originally introduced as a solution to an optimization problem, which has impeded attempts to explain its excellent performance. Here it is shown, that the turbo decoder is an iterative method seeking a solution to an intuitively pleasing constrained optimization problem. In particular, the turbo decoder seeks the maximum-likelihood sequence (MLS) under the false assumption that the input to the encoders are chosen independently of each other in the parallel case, or that the output of the outer encoder is chosen independently of the input to the inner encoder in the serial case. To control the error introduced by the false assumption, the optimizations are performed subject to a constraint on the probability that the independent messages happen to coincide. When the constraining probability equals one, the global maximum of the constrained optimization problem is the maximum-likelihood sequence detection (MLSD), allowing for a theoretical connection between turbo decoding and MLSD. It is then shown that the turbo decoder is a nonlinear block Gauss–Seidel iteration that aims to solve the optimization problem by zeroing the gradient of the Lagrangian with a Lagrange multiplier of -1. Some conditions for the convergence for the turbo decoder are then given by adapting the existing literature for Gauss–Seidel iterations.**

***Index Terms*—Constrained optimization, maximum-likelihood decoding, turbo decoder convergence analysis.**

I. Introduction

Along with being one of the most prominent communications inventions of the past decade, the introduction of turbo codes in [3] began a new era in communications systems achieving unprecedented performance. The creation of the turbo decoder introduced a new method of decoding these codes which brought the decoding of complex codes within the reach of computationally practical algorithms. The iterative decoding algorithm, while being suboptimal, performs well enough to bring turbo codes very close to theoretically attainable limits.

An accurate justification for the decoding strategy's performance is still incomplete. While it has been proven that turbo codes have good distance properties, which would be relevant for maximum-likelihood decoding, researchers have not yet succeeded in developing a proper connection between the suboptimal turbo decoder and maximum-likelihood decoding. This is exacerbated by the fact that the turbo decoder, unlike most of the designs in modern communications systems engineering, was not originally introduced as a solution to an optimization problem. This has made explaining just why the turbo decoder performs as well as it does very difficult. Together with the lack of formulation as a solution to an optimization problem, the turbo decoder is an iterative algorithm, which makes determining its convergence and stability behavior important. Much of the analytical work concerning the turbo decoder, then, has focussed on determining its convergence and stability properties. Significant progress along these lines has been made with EXIT style analysis [4] and density evolution [5], but these techniques ultimately appeal to results which become valid only when the block length grows rather large. Other attempts, such as connections to factor graphs [6] and belief propagation [7], have been hindered from showing convergence due to loops in the turbo coding graph. The information geometric attempts [8]–[12], in turn provided an intriguing partial description of the problem in terms of information projections, which allowed for linearized local stability results in [9], [10], but complete results have been hampered by an inability to efficiently describe extrinsic information extraction as an information projection.

None of these convergence frameworks, so far, have identified the optimization problem that the decoder is attempting to solve. Within the context of belief propagation [13], a general family of algorithms which includes the turbo decoder [6], [7], several authors [14]–[19] have shown that the turbo decoder is related to an approximation in statistical physics which is a constrained optimization, but the development there is based on ideas from statistical physics which are not essential for the analysis of the turbo decoder. In particular, [14]–[18] and [19] show that the stationary points of belief propagation and thus turbo decoding minimize the Bethe approximation to the variational free energy. While this approximation has a long and established history within the context of statistical physics, its intuitive meaning within the context of turbo decoding is less than transparent. Indeed, given that the Bethe approximation can not be expected to be exact in factor graphs with loops (a class within which all turbo codes are bound to lie), it is not clear why minimizing it can yield such good performance in these cases.

Manuscript received July 29, 2005; revised July 14, 2006 and August 7, 2006. The work of J. M. Walsh and C. R. Johnson, Jr. were supported in part by Applied Signal Technology and NSF Grants CCF-0310023 and INT-0233127. The work of P. A. Regalia was supported in part by the CNRS of France, under Contract 14871, and the Network of Excellence in Wireless Communications (NEWCOM), E. C. Contract 507325, while with the Groupe des Ecoles des Télécommunications, INT, 91011 Evry, France. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Adelaide, Australia, September 2005 and the 43rd Allerton Conference on Communication, Control, and Computing, Monticello, IL, September 2005.

J. M. Walsh is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: jwalsh@ece.drexel.edu).

P. A. Regalia is with The Catholic University of America, Washington DC 20064 USA (e-mail: regalia@cua.edu).

C. R. Johnson, Jr. is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: johnson@ece.cornell.edu).

Communicated by M. P. Fossorier, Associate Editor for Coding Techniques.

Digital Object Identifier 10.1109/TIT.2006.885535

This paper takes a different approach, choosing a different objective function and constraints which both have clear intuitive meaning within the context of the turbo decoder and still yield the stationary points of the turbo decoder as critical points of the constrained optimization's Lagrangian.[1]

After introducing our notation and reviewing the turbo decoder in Section II, we will show in Section III that the turbo decoder admits an exact interpretation as a well-known iterative method [1] attempting to find a solution to a particular intuitively pleasing constrained optimization problem. In our formulation of the constrained optimization problem, it will become clear that the turbo decoder is calculating a constrained maximum-likelihood sequence detection (MLSD). This is to be contrasted to the suggestion in [14], [15], [17], [18] that belief propagation decoding results in estimates of the maximum-likelihood bitwise detection (MLBD), partially owing to the fact that the existing proofs of convergence within that arena are primarily based on the loopless case for which bitwise optimality is well established [6]. Of course, the constraints in the constrained MLSD bias the turbo decoder away from the exact MLSD, so that in general it can be expected to be neither exactly the MLSD nor the MLBD. Properly identifying the iterative method that is being used then allows us to give some conditions for convergence of the turbo decoder by borrowing convergence theory for the nonlinear block Gauss–Seidel iteration in Section IV.

II. Preliminaries and Notation

Before we get to answering some key questions about the turbo decoder, we will need to discuss some preliminary topics. In the following development, we will find it useful to consider families of probability measures on the possible binary words of length N. This will lead us in Section II.A to consider the geometric structure of this family of probability measures by finding parameterizations of it that will be useful in the sequel. Next, in Section II-B we will consider a formulation of maximum-likelihood sequence decoding which is rather atypical, but bears important resemblance to the turbo decoder. We then briefly review the operation of the turbo encoder and decoder in Section II-C, which may be helpful for some readers, and should also reinforce the information geometric notation that we will use in the remainder of the development.

A. Information Geometry

Let $\mathbf{B}_i \in \{0,1\}^N$ for $i \in \{0,\ldots,2^N-1\}$ denote the binary representation of the integer i. Then, by forming the matrix

$$\mathbf{B} = (\mathbf{B}_0, \mathbf{B}_1, \ldots, \mathbf{B}_{2^N-1})^T \in \{0,1\}^{2^N \times N}$$

we can create a matrix whose rows collectively are all the possible binary words of length N. Given a random binary word of length N, call it ξ, we will be interested in different probability mass functions (PMFs) on the outcomes $\{\xi = \mathbf{B}_i\}$. Since there are a finite number of such outcomes, we can completely characterize such a measure q, by simply listing the probabilities $\{q(\mathbf{B}_i) = \mathsf{Pr}[\xi = \mathbf{B}_i]\}$. Furthermore, because q is a PMF, we must have $q(\mathbf{B}_i) \geq 0$ and

$$\sum_i q(\mathbf{B}_i) = 1$$

We are then content, that, to parameterize the set $\mathcal{F}$ of all PMFs on the outcomes $\{\xi = \mathbf{B}_i\}$, it is sufficient to consider the set $\mathcal{F}_\eta$ of vectors of the form

$$\eta = (q(\mathbf{B}_0), q(\mathbf{B}_1), \ldots, q(\mathbf{B}_{2^N-1}))^T$$

whose entries are nonnegative and sum to one. We shall also find it convenient later to work with the log coordinates for PMFs in $\mathcal{F}$. Given a PMF $q \in \mathcal{F}$, its log coordinates are the vector $\boldsymbol{\theta}$ whose ith element is given by

$$\theta_i = \log(q(\mathbf{B}_i)) - \log(q(\mathbf{B}_0))$$

Given, then, a vector $\boldsymbol{\theta}$, we see that we can uniquely determine its corresponding wordwise PMF q, by using the list of probabilities in the vector η, which can be written in terms of $\boldsymbol{\theta}$ as

$$\eta = \exp(\boldsymbol{\theta} - \psi(\boldsymbol{\theta})), \quad \psi(\boldsymbol{\theta}) := \log(\|\exp(\boldsymbol{\theta})\|_1) \tag{1}$$

where $\|\cdot\|_1$ is the 1-norm (sum of the absolute values of the components of a vector argument). In fact, one may show that $\psi(\boldsymbol{\theta})$ is actually the convex conjugate [23] and dual potential under the Legendre transformation [24] to the negative of the Shannon entropy, so that

$$\psi(\boldsymbol{\theta}) + \mathsf{H}(\eta) \geq \langle \boldsymbol{\theta}, \eta \rangle$$

with equality iff $\boldsymbol{\theta}$ and η are coordinates for the same PMF, where H is the negative of the Shannon entropy

$$\mathsf{H}(\eta) = \langle \eta, \log(\eta) \rangle$$

It is often to convenient to work with the log coordinates of PMFs, since if the random binary words $\xi, \boldsymbol{\chi}, \zeta$ satisfy $\mathsf{Pr}[\zeta = \mathbf{B}_i] = \mathsf{Pr}[\boldsymbol{\chi} = \mathbf{B}_i]\mathsf{Pr}[\xi = \mathbf{B}_i]$ for all i, we have that

$$\theta_\zeta = \theta_\xi + \theta_{\boldsymbol{\chi}} \tag{2}$$

We will find it useful to parameterize the subset $\mathcal{P} \subset \mathcal{F}$ which contains those probability measures Pr on $\{\mathbf{B}_i\}$ that factor into the product of their bitwise marginals, so that

$$\mathsf{Pr}(\mathbf{x}) = \prod_i \mathsf{Pr}(x_i)$$

One can show [9], [10], [17], that this set may be parameterized by the vectors $\boldsymbol{\lambda} \in \mathbb{R}^N$ of bitwise log probability ratios which have elements of the form

$$\lambda_i = \log \frac{\mathsf{Pr}(x_i = 1)}{1 - \mathsf{Pr}(x_i = 1)}$$

The log coordinates of a factorable measure $\mathsf{Pr} \in \mathcal{P}$ then take the form

$$\boldsymbol{\theta} = \mathbf{B}\boldsymbol{\lambda} \tag{3}$$

[1]Although it is well beyond the scope of this paper, the reader familiar with the Bethe approximation to the variational free energy may wish to consult [20], [21] and [22] for the interplay between these two different optimization frameworks that yield the same critical points.

By combining the facts (2) and (3), we may represent the log coordinates of a wordwise PMF which results by weighting a likelihood function whose log coordinates are $\boldsymbol{\theta}$ with a factorable PMF with bitwise log probability ratios $\boldsymbol{\lambda}$ as $\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\theta}$. A fact that we will use later is that the gradient of $\psi(\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\theta})$ with respect to $\boldsymbol{\lambda}$ is in fact the vector whose ith element is the marginal probability that the ith bit is one according to the PMF with log coordinates $\mathbf{B}\boldsymbol{\lambda} + \boldsymbol{\theta}$.

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}}\psi(\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\lambda}}\log(\|\exp(\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta})\|_1)\\ &= \mathbf{B}^T\frac{\exp(\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta})}{\|\exp(\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta})\|_1}\\ &= \mathbf{B}^T\eta_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta}}\\ &= \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}+\boldsymbol{\theta}} \end{aligned} \tag{4}$$

Here we have used the notation $\eta_{\boldsymbol{\theta}}$ to represent the PMF vector of the measure with log coordinates $\boldsymbol{\theta}$, and $\mathbf{p}_{\boldsymbol{\theta}}$ to represent the vector whose ith element is the probability that the ith bit is one according to the measure with log coordinates $\boldsymbol{\theta}$.

We will also be interested in handling cases where zero and one bitwise probabilities are possible, which correspond to vectors of log probability ratios $\boldsymbol{\lambda}$s with some elements that are infinite. In particular, for the cases when $\boldsymbol{\lambda}$ contains some infinite values, decompose it into $\boldsymbol{\lambda} = \boldsymbol{\lambda}_\infty + \boldsymbol{\lambda}_{\mathbb{R}}$ where $\boldsymbol{\lambda}_\infty \in \{\pm\infty, 0\}^N$ and $\boldsymbol{\lambda}_{\mathbb{R}} \in \mathbb{R}^N$. Given any function $f : \mathbb{R}^N \rightarrow \mathbb{R}$, we will define $f(\boldsymbol{\lambda})$ by extending $f(\boldsymbol{\lambda}) = \lim_{t\rightarrow\infty} f(t\boldsymbol{\lambda}_s + \boldsymbol{\lambda}_{\mathbb{R}})$ where $\boldsymbol{\lambda}_s$ is any vector of finite log probability ratios whose signs match those of $\boldsymbol{\lambda}_\infty$, so that $\mathsf{sign}(\boldsymbol{\lambda}_s) = \mathsf{sign}(\boldsymbol{\lambda}_\infty)$. In every case we shall encounter in this article, these limits will be the same regardless of the choice of $\boldsymbol{\lambda}_s$, so long as $\mathsf{sign}(\boldsymbol{\lambda}_s) = \mathsf{sign}(\boldsymbol{\lambda}_\infty)$. This definition has the nice property that if we rewrote $f(\boldsymbol{\lambda})$ in terms of the vector of bitwise probabilities $\mathbf{p} \in [0,1]^N$ corresponding to $\boldsymbol{\lambda}$, and then evaluated it as the probabilities corresponding to $\boldsymbol{\lambda}$ (with possibly infinite values and thus some probabilities in $\{0,1\}$), we would recover the same value for $f(\boldsymbol{\lambda})$.

We will exploit (4) later when we are discussing the operation of the turbo decoder. First, we will reexamine maximum-likelihood decoding for a generic encoder.

B. Maximum-Likelihood Decoding

In a typical decoding situation, a binary message $\boldsymbol{\xi}$ is encoded and transmitted over a channel to create the received data $\mathbf{r}$. Given $\mathbf{r}$ we would like to reconstruct the original message $\boldsymbol{\xi}$. There are two senses of "optimal" when it comes to decoding $\boldsymbol{\xi}$ in the situation where we have no prior probabilities for the outcomes $\{\boldsymbol{\xi} = \mathbf{B}_i\}$. In one situation, we wish to minimize the likelihood of selecting the wrong sequence $\boldsymbol{\xi}$ so as to minimize the block error rate. In another situation we wish to minimize the probability of selecting the wrong bit ξ_i so as to minimize the bit error rate. The former yields MLSD, the latter MLBD. The MLSD is then

$$\hat{\boldsymbol{\xi}}_{\mathrm{MLSD}} = \arg\max_{\boldsymbol{\xi}\in\{0,1\}^N} p(\mathbf{r}\,|\,\boldsymbol{\xi})$$

and the MLBD is

$$\hat{\xi}_{i,\mathrm{MLBD}} = \arg\max_{\xi_i\in\{0,1\}} \sum_{\mathbf{x}\,|\,x_i=\xi_i} p(\mathbf{r}\,|\,\mathbf{x})$$

where $p(\mathbf{r}\,|\,\boldsymbol{\xi})$ is the likelihood function which results from concatenating the encoder with the channel.

One can set up maximum-likelihood parameter estimation problems which yield these detectors as their solutions. In particular, consider a parameter estimation problem where we are trying to determine the factorable prior PMF on $\boldsymbol{\xi}$ which yields the maximum *a posteriori* probability of having received $\mathbf{r}$. This problem then takes the form

$$q_{\mathrm{ML}} = \arg\max_{q\in\mathcal{P}} \sum_i p(\mathbf{r}\,|\,\boldsymbol{\xi} = \mathbf{B}_i)q(\mathbf{B}_i) \tag{5}$$

We know from Section II-A that to parameterize the set $\mathcal{P}$, it is sufficient to use a vector of log probability ratios $\boldsymbol{\lambda}$, thus we can set this problem as selecting

$$\begin{aligned}\hat{\boldsymbol{\lambda}}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\lambda}\in(\mathbb{R}\bigcup\{\pm\infty\})^N} p(\mathbf{r}\,|\,\boldsymbol{\lambda})\\ &= \arg\max_{\boldsymbol{\lambda}\in(\mathbb{R}\bigcup\{\pm\infty\})^N} \sum_i p(\mathbf{r}\,|\,\boldsymbol{\xi}=\mathbf{B}_i)q(\mathbf{B}_i\,|\,\boldsymbol{\lambda})\end{aligned}$$

In particular, since we know that for any $\boldsymbol{\lambda}$ we must have

$$\sum_i q(\mathbf{B}_i\,|\,\boldsymbol{\lambda}) = 1$$

we see that the q that maximizes (5) takes the form

$$q(\mathbf{B}_i\,|\,\boldsymbol{\lambda}) = \begin{cases}1, & \mathbf{B}_i = \hat{\boldsymbol{\xi}}_{\mathrm{MLSD}}\\ 0, & \text{otherwise}\end{cases}$$

since putting any probability mass on any other word would not yield as high a likelihood. This then implies that $\hat{\boldsymbol{\lambda}}_{\mathrm{ML}}$ are the infinite log probability ratios associated with the word $\hat{\boldsymbol{\xi}}_{\mathrm{MLSD}}$.

One can also set up a set of maximum-likelihood parameter estimation problems whose answers are the MLBDs. In particular, consider the set of estimation problems

$$\begin{aligned}\hat{\lambda}_{i,\mathrm{ML}} &= \arg\max_{\lambda\in\mathbb{R}\bigcup\{\pm\infty\}} \sum_{\boldsymbol{\xi}} p(\mathbf{r}\,|\,\boldsymbol{\xi})\mathsf{Pr}[\xi_i\,|\,\lambda]\\ &= \arg\max_{\lambda\in\mathbb{R}\bigcup\{\pm\infty\}} \mathsf{Pr}[\xi_i = 1\,|\,\lambda]\left(\sum_{\boldsymbol{\xi}\,|\,\xi_i=1} p(\mathbf{r}\,|\,\boldsymbol{\xi})\right)\\ &\quad + \mathsf{Pr}[\xi_i = 0\,|\,\lambda]\left(\sum_{\boldsymbol{\xi}\,|\,\xi_i=0} p(\mathbf{r}\,|\,\boldsymbol{\xi})\right).\end{aligned}$$

From the latter form, it is evident that

$$\mathsf{Pr}[\xi_i = 1\,|\,\hat{\lambda}_{i,\mathrm{ML}}] = \begin{cases}1, & \sum_{\boldsymbol{\xi}\,|\,\xi_i=1} p(\mathbf{r}\,|\,\boldsymbol{\xi}) > \sum_{\boldsymbol{\xi}\,|\,\xi_i=0} p(\mathbf{r}\,|\,\boldsymbol{\xi})\\ 0, & \text{otherwise}\end{cases}$$

which shows that $\hat{\lambda}_{i,\mathrm{ML}}$ is the (infinite) log probability ratio corresponding to the MLBD.

With these preliminaries, we now review the turbo encoder and decoder, which are suboptimal, yet have been shown via simulation to have near optimal performance at a reasonable complexity.

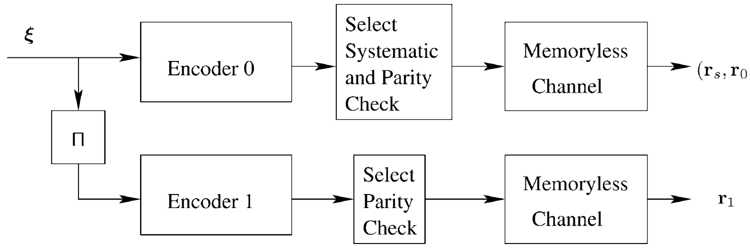

Fig. 1. Parallel concatenation of two convolutional codes with interleavers.

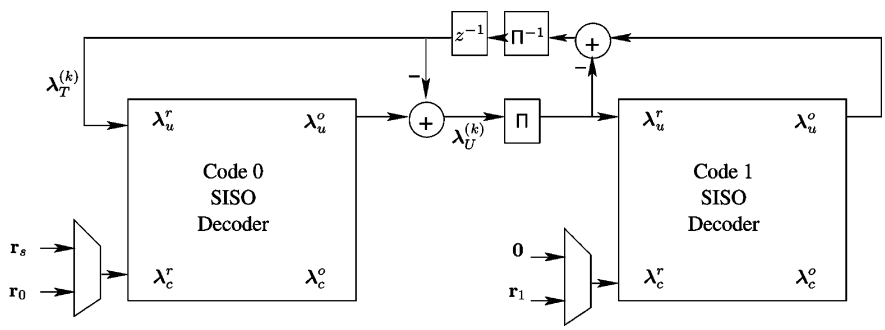

Fig. 2. The parallel concatenated turbo decoder.

C. The Turbo Encoder

Consider the parallel turbo encoder depicted in Fig. 1. A block of N bits $\boldsymbol{\xi}$ is interleaved to get $\hat{\boldsymbol{\chi}} = \Pi(\boldsymbol{\xi})$. (We will denote the deinterleaved $\Pi^{-1}(\hat{\boldsymbol{\chi}})$ by $\boldsymbol{\chi}$ for convenience in notation.) Then, $\boldsymbol{\xi}$ and $\hat{\boldsymbol{\chi}}$ are encoded with two, possibly different systematic convolutional encoders. We then pass the systematic bits, parity check bits from the first encoder, and parity check bits from the second encoder, over a noisy memoryless channel to get the channel outputs $\mathbf{r}_s$, $\mathbf{r}_0$, and $\mathbf{r}_1$, respectively, which we collect into a large vector $\mathbf{r} = (\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1)$.

The optimal decoder, in the sense of minimizing block error probability, at the receiver would choose $\hat{\boldsymbol{\xi}}$ to be the message which maximizes the likelihood function

$$\begin{aligned}\hat{\boldsymbol{\xi}} &= \arg\max_{\boldsymbol{\xi}\in\{0,1\}^N} p(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 \,|\, \boldsymbol{\xi}) \\ &= \arg\max_{\boldsymbol{\xi}\in\{0,1\}^N} p(\mathbf{r}_s, \mathbf{r}_0 \,|\, \boldsymbol{\xi}) p(\mathbf{r}_1 \,|\, \hat{\boldsymbol{\chi}} = \Pi(\boldsymbol{\xi}))\end{aligned}$$

where the second factorization is admitted by the memoryless nature of the channel. As discussed in Section II-B, this is equivalent to choosing the prior factorable PMF for $\boldsymbol{\xi}$ that maximizes the *a posteriori* likelihood function subject to the constraint that $\boldsymbol{\chi} = \boldsymbol{\xi}$. To see this select log probability ratios $\boldsymbol{\lambda}$ to parameterize the prior PMF for $\boldsymbol{\xi}$, and write the *a posteriori* likelihood function

$$p_{\text{true}}(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 \,|\, \boldsymbol{\lambda}) = \sum_i p(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 \,|\, \boldsymbol{\xi} = \mathbf{B}_i) \Pr(\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda})$$

This is maximized when

$$\Pr(\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}) = \begin{cases} 1 & i = \arg\max_j p(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 \,|\, \boldsymbol{\xi} = \mathbf{B}_j) \\ 0 & \text{otherwise} \end{cases}$$

so that $\boldsymbol{\lambda}$ corresponds to the (infinite) log probability ratios corresponding to the maximum-likelihood sequence detector.

Unfortunately, such a procedure is not computationally feasible at the receiver. To cope with this problem, the parallel turbo decoder makes use of an exchange of information between computationally efficient decoders for each of the component codes as depicted in Fig. 2. Denoting by

$$\begin{aligned}[\boldsymbol{\theta}_0]_i &= \log(p(\mathbf{r}_s, \mathbf{r}_0 \,|\, \boldsymbol{\xi} = \mathbf{B}_i)) - \log(p(\mathbf{r}_s, \mathbf{r}_0 \,|\, \boldsymbol{\xi} = \mathbf{B}_0)) \\ [\boldsymbol{\theta}_1]_i &= \log(p(\mathbf{r}_1 \,|\, \boldsymbol{\chi} = \mathbf{B}_i)) - \log(p(\mathbf{r}_1 \,|\, \boldsymbol{\chi} = \mathbf{B}_0))\end{aligned}$$

and denoting $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ as the vectors of information exchanged between the two decoders, one may write the parallel turbo decoder succinctly as iterating

$$\begin{aligned}\boldsymbol{\lambda}_T^{(k)} &= \pi\left(\mathbf{B}\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\theta}_0\right) - \boldsymbol{\lambda}_U^{(k)} \\ \boldsymbol{\lambda}_U^{(k+1)} &= \pi\left(\mathbf{B}\boldsymbol{\lambda}_T^{(k)} + \boldsymbol{\theta}_1\right) - \boldsymbol{\lambda}_T^{(k)}\end{aligned}$$

where π takes a log PMF to its bitwise marginal log probability ratios, and thus may be written as

$$\pi(\boldsymbol{\theta}) = \log(\mathbf{B}^T \exp(\boldsymbol{\theta} - \psi(\boldsymbol{\theta}))) - \log((\mathbf{1} - \mathbf{B})^T \exp(\boldsymbol{\theta} - \psi(\boldsymbol{\theta})))$$

where we denoted by $\mathbf{1}$ the $2^N \times N$ matrix whose entries are all one.[2] If we denote by $\mathbf{p}_{\boldsymbol{\theta}}$ the vector of bitwise marginal probabilities of the bits being one according to the wordwise log PMF $\boldsymbol{\theta}$, we may also write the turbo decoder as

$$\begin{aligned}\mathbf{p}_{\mathbf{B}\left(\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\lambda}_T^{(k)}\right)} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\theta}_0} \\ \mathbf{p}_{\mathbf{B}\left(\boldsymbol{\lambda}_U^{(k+1)} + \boldsymbol{\lambda}_T^{(k)}\right)} &= \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T^{(k)} + \boldsymbol{\theta}_1}\end{aligned}$$

since the extrinsic information vectors $\boldsymbol{\lambda}_T^{(k)}$ are first chosen so that they match the *a posteriori* probabilities from the first decoder when they are added to its prior information $\boldsymbol{\lambda}_U^{(k)}$, and

[2]For consistency with the references, we have chosen the original decoder formulation from [3], [25] which explicitly includes $\mathbf{r}_s$ in only one of the component decoders. The reader unfamiliar with this form of the parallel turbo decoder may verify that the systematic information from $\mathbf{r}_s$ is still used (via the extrinsic information vectors $\boldsymbol{\lambda}_T$) in both decoders, although it is not explicitly included in $\boldsymbol{\theta}_1$

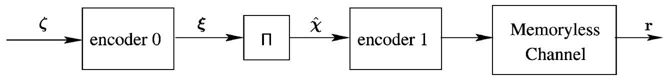

Fig. 3. The serially concatenated turbo encoder.

similarly for $\boldsymbol{\lambda}_U^{(k+1)}$ with respect to the second decoder and prior information $\boldsymbol{\lambda}_T^{(k)}$.

The serial turbo encoder is depicted in Fig. 3. A block $\boldsymbol{\zeta}$ of M information bits are encoded with a systematic convolutional encoder to get $\boldsymbol{\xi}$, which is then interleaved to get $\hat{\boldsymbol{\chi}} = \Pi(\boldsymbol{\xi})$, and then encoded with another encoder, which may not be systematic, and passed over a memoryless channel to give the received vector $\mathbf{r}$.

At the receiver, the optimal decoder, in the sense of minimizing the block error probability would select the decoded message $\hat{\boldsymbol{\zeta}}$ to be the message which maximizes the likelihood function

$$\hat{\boldsymbol{\zeta}}_{\mathrm{ML}} = \arg \max_{\boldsymbol{\zeta} \in \{0,1\}^N} p(\mathbf{r} \,|\, \boldsymbol{\zeta})$$

In fact, one can show, following a technique similar to that from Section II.B, that $\hat{\boldsymbol{\zeta}}_{\mathrm{ML}}$ can be related to an estimation problem. In particular, consider the problem of estimating the factorable PMF for $\boldsymbol{\xi}$, parameterized by a vector of log probability ratios $\boldsymbol{\lambda}$, which maximized the likelihood function

$$\begin{aligned}\boldsymbol{\lambda}_{\mathrm{ML}} &= \arg\max_{\boldsymbol{\lambda}\in\mathcal{A}} \sum_{\boldsymbol{\zeta}} p(\mathbf{r} \,|\, \boldsymbol{\xi} = \mathsf{C}_0(\boldsymbol{\zeta}))\mathsf{Pr}[\boldsymbol{\xi} = \mathsf{C}_0(\boldsymbol{\zeta}) \,|\, \boldsymbol{\lambda}] \\ &= \arg\max_{\boldsymbol{\lambda}\in\mathcal{A}} \sum_{i} p(\mathbf{r} \,|\, \boldsymbol{\xi} = \mathbf{B}_i)\phi(\mathbf{B}_i)\mathsf{Pr}[\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}] \qquad (6)\end{aligned}$$

where $\mathcal{A} := (\mathbb{R}\bigcup\{\pm\infty\})^N$ and in the second line we have used the indicator function for encoder 0:

$$\phi(\mathbf{B}_i) = \begin{cases} 1, & \mathbf{B}_i \text{ is in encoder 0's codebook} \\ 0, & \text{otherwise} \end{cases}$$

and we have used C_0 to represent the function which takes inputs to encoder 0 to their encoded codewords. Once again, realizing that we must have

$$\sum_i \mathsf{Pr}[\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}] = 1$$

we see that the $\boldsymbol{\lambda}_{\mathrm{ML}}$ which maximizes (6) is the one which satisfies

$$\mathsf{Pr}[\mathbf{B}_i \,|\, \boldsymbol{\lambda}_{\mathrm{ML}}] = \begin{cases} 1, & \mathbf{B}_i = \mathsf{C}_0(\hat{\zeta}_{\mathrm{ML}}) \\ 0, & \text{otherwise} \end{cases}$$

so that $\boldsymbol{\lambda}_{\mathrm{ML}}$ are the infinite log probability ratios corresponding to the encoded version of the MLD.

Unfortunately, as in the parallel case, this optimal decoder is not feasible. To cope with this problem, the serial turbo decoder makes use of an exchange in information between computationally efficient decoders for each of the component codes as depicted in Fig. 4. Denoting by

$$\begin{aligned}[\boldsymbol{\theta}_0]_i &= \log(p(\mathbf{r} \,|\, \boldsymbol{\xi} = \mathbf{B}_i)) - \log(p(\mathbf{r} \,|\, \boldsymbol{\xi} = \mathbf{B}_0)) \\ [\boldsymbol{\theta}_1]_i &= \log(\phi(\mathbf{B}_i)) - \log(\phi(\mathbf{B}_0)) \\ &= \begin{cases} 0, & \mathbf{B}_i \text{ is in encoder 0's codebook} \\ -\infty, & \text{otherwise} \end{cases}\end{aligned}$$

where we will assume that $\mathbf{B}_0$ (the all zero codeword) is in encoder 0's codebook, which is satisfied, for instance, if the inner code is linear. Denoting $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ as the vectors of information exchanged between the two decoders, one may write the serial turbo decoder succinctly as iterating

$$\boldsymbol{\lambda}_T^{(k)} = \pi\left(\mathbf{B}\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\theta}_0\right) - \boldsymbol{\lambda}_U^{(k)} \qquad (7)$$

$$\boldsymbol{\lambda}_U^{(k+1)} = \pi\left(\mathbf{B}\boldsymbol{\lambda}_T^{(k)} + \boldsymbol{\theta}_1\right) - \boldsymbol{\lambda}_T^{(k)} \qquad (8)$$

or, equivalently

$$\mathbf{p}_{\mathbf{B}\left(\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\lambda}_T^{(k)}\right)} = \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U^{(k)} + \boldsymbol{\theta}_0} \qquad (9)$$

$$\mathbf{p}_{\mathbf{B}\left(\boldsymbol{\lambda}_U^{(k+1)} + \boldsymbol{\lambda}_T^{(k)}\right)} = \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T^{(k)} + \boldsymbol{\theta}_1} \qquad (10)$$

since the extrinsic information vectors $\boldsymbol{\lambda}_T^{(k)}$ are first chosen so that they match the *a posteriori* probabilities from the first decoder when they are added to its prior information $\boldsymbol{\lambda}_U^{(k)}$, and similarly for $\boldsymbol{\lambda}_U^{(k+1)}$ with respect to the second decoder and prior information $\boldsymbol{\lambda}_T^{(k)}$. Here, we have repeated the same notation for the serial decoder and the parallel decoder to emphasize the fact that they have the same form when written on this abstract level. The only important difference, aside from the fact that $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ are determined in different ways for the serial and parallel concatenated cases, is that in the serial decoder it is probabilities for the output of the first encoder $\mathsf{C}_0(\boldsymbol{\zeta})$ which are being exchanged, as opposed to the original message bits $\boldsymbol{\xi}$ in the parallel turbo decoder. For this reason, we have deliberately chosen $\boldsymbol{\xi}$ to denote both the input to the parallel encoder in the parallel concatenated case, and the output of the outer encoder in the serial concatenated case, so that it is always probabilities for bits in $\boldsymbol{\xi}$ that we are exchanging in either the serial or parallel concatenated case.

III. The Turbo Decoder as an Iterative Solution to a Constrained Optimization Problem

Here we will somewhat demystify the turbo decoder by showing both the sense in which its stationary points are optimal, as well as by identifying the iterative method which is being used to find these optimal points. We must first consider a system which, although it is not exactly equal to that in which the turbo decoder operates, is the system which the turbo decoder assumes in its sense of optimality. We will then provide the optimization interpretation of the turbo decoder, followed by some commentary and conclusions.

A. A Convenient Independence Assumption

In describing the operation of the turbo decoder, it will be advantageous to describe a system like that in which the parallel turbo decoder operates, but in which the messages $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ are

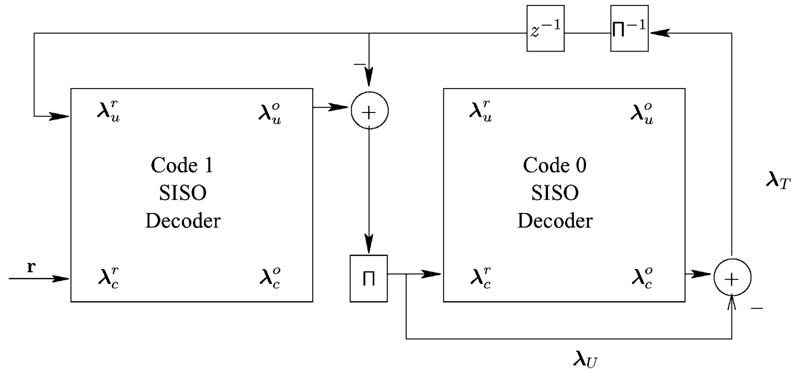

Fig. 4. The serially concatenated turbo decoder.

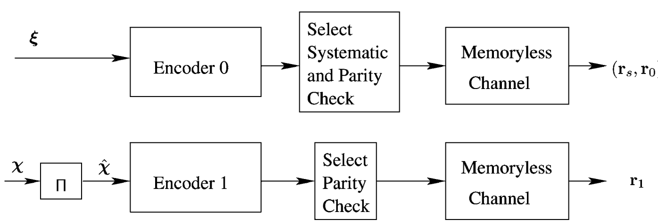

Fig. 5. The system which the parallel turbo decoder assumes.

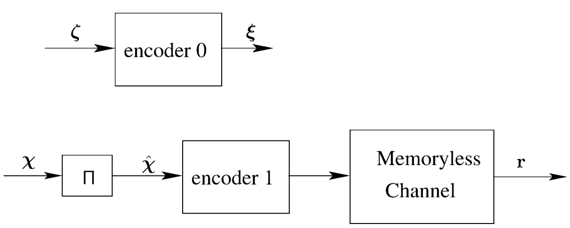

Fig. 6. The system which the serial turbo decoder assumes.

chosen completely independently of one another as depicted in Fig. 5 for the parallel case and in Fig. 6 for the serial case [2]. Under the false assumption that $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ were chosen independently of one another according to factorable PMFs with (deterministic) bitwise log probability ratios $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$, the likelihood function for the received data would be

$$\begin{aligned} p(\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1 \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) &= \left(\sum_i p(\mathbf{r}_s, \mathbf{r}_0 \,|\, \boldsymbol{\xi} = \mathbf{B}_i) \mathsf{Pr}(\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}_U) \right) \\ &\quad \times \left(\sum_j p(\mathbf{r}_1 \,|\, \boldsymbol{\chi} = \mathbf{B}_j) \mathsf{Pr}(\boldsymbol{\chi} = \mathbf{B}_j \,|\, \boldsymbol{\lambda}_T) \right) \\ &= \frac{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U)\|_1} \frac{\|\exp(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_T)\|_1} \end{aligned}$$

in the case of parallel concatenation and

$$\begin{aligned} p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) &= \left(\sum_i p(\mathbf{r} \,|\, \boldsymbol{\xi} = \mathbf{B}_i) \mathsf{Pr}(\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}_U) \right) \\ &\quad \times \left(\sum_j \phi(\mathbf{B}_j) \mathsf{Pr}(\boldsymbol{\chi} = \mathbf{B}_j \,|\, \boldsymbol{\lambda}_T) \right) \\ &= \frac{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U)\|_1} \frac{\|\exp(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)\|_1}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_T)\|_1} \end{aligned}$$

in the case of serial concatenation. Thus, recalling the definition of ψ from (1), the log likelihood function in either (serial or parallel) case is

$$\begin{aligned} \log(p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T)) = -\psi(\mathbf{B}\boldsymbol{\lambda}_U) - \psi(\mathbf{B}\boldsymbol{\lambda}_T) \\ + \psi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) + \psi(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1) \end{aligned}$$

Furthermore, the decision statistics which the parallel turbo decoder uses to make its decisions, which are the bitwise posterior probabilities at the output of one of the component decoders and have the log probability ratio vector $\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T$, may be interpreted under this notation as

$$[\mathsf{Pr}(\boldsymbol{\xi}_i = \boldsymbol{\chi}_i = 1 \,|\, \boldsymbol{\xi} = \boldsymbol{\chi}, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T)] = \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)}$$

or the collection over all the bits of the probability that the ith bit of $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ are equal to 1, given that $\boldsymbol{\xi} = \boldsymbol{\chi}$ and $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ are chosen with factorable PMFs with log probability ratios $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$, respectively. Since we have assumed that the first component encoder in the serial turbo encoder is systematic, the serial turbo decoder uses the same statistics for its decisions, but only the elements in $\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)}$ that correspond to systematic bits from that first encoder are needed.

To control this false assumption of independently choosing $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$, we will be interested in choosing $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ so that the two densities so chosen have a large probability of selecting the same word. This is because we know a priori that $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ are the same, and thus, although we are relaxing the constraint that they be exactly the same in doing the decoding, we want to enforce that they be similar. Thus, at the decoder, we will be interested in the constraint set

$$\mathcal{C} = \{(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) \,|\, \mathsf{Pr}(\boldsymbol{\xi} = \boldsymbol{\chi} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) = c\}$$

which is the set such that the probability of choosing the same message for $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$, given that we chose them independently according to the factorable densities whose bitwise marginals are $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ respectively, is fixed to some constant c. We will now use the notation

$$\mathsf{Pr}(\boldsymbol{\xi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}_U) = \frac{\exp(\mathbf{B}_i \boldsymbol{\lambda}_U)}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U)\|_1}$$
$$\mathsf{Pr}(\boldsymbol{\chi} = \mathbf{B}_i \,|\, \boldsymbol{\lambda}_T) = \frac{\exp(\mathbf{B}_i \boldsymbol{\lambda}_T)}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_T)\|_1}$$

in order to rewrite

$$\begin{aligned}&\mathsf{Pr}(\boldsymbol{\xi}=\boldsymbol{\chi}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)\\&\quad=\sum_i \mathsf{Pr}(\boldsymbol{\xi}=\mathbf{B}_i\,|\,\boldsymbol{\lambda}_U)\mathsf{Pr}(\boldsymbol{\chi}=\mathbf{B}_i\,|\,\boldsymbol{\lambda}_T)\\&\quad=\frac{\|\exp(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T))\|_1}{\|\exp(\mathbf{B}\boldsymbol{\lambda}_U)\|_1\|\exp(\mathbf{B}\boldsymbol{\lambda}_T)\|_1}\end{aligned}$$

so that

$$\begin{aligned}&\log(\mathsf{Pr}(\boldsymbol{\xi}=\boldsymbol{\chi}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T))\\&\quad=\log(\|\exp(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T))\|_1)\\&\qquad-\log(\|\exp(\mathbf{B}\boldsymbol{\lambda}_U)\|_1)\\&\qquad-\log(\|\exp(\mathbf{B}\boldsymbol{\lambda}_T)\|_1)\\&\quad=\psi(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T))\\&\qquad-\psi(\mathbf{B}\boldsymbol{\lambda}_U)-\psi(\mathbf{B}\boldsymbol{\lambda}_T).\end{aligned}$$

Thus, we can write the constraint set as

$$\mathcal{C}=\{(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)\,|\,\psi(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T))-\psi(\mathbf{B}\boldsymbol{\lambda}_U)-\psi(\mathbf{B}\boldsymbol{\lambda}_T)=\log(c)\}.$$

B. An Exact Characterization of the Turbo Decoder

In the next theorem we show that the turbo decoder is an iterative attempt to find the maximum-likelihood estimate for bitwise log probability ratios $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ under the false assumption that $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ where chosen independently of one another according to the bitwise factorable PMFs with log probability ratios $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$, respectively. Furthermore, this optimization is performed subject to the constraint that the probability of selecting the same message when selecting $\boldsymbol{\xi}$ and $\boldsymbol{\chi}$ in this manner is held constant.

Theorem 1 (What is turbo decoding?): The turbo decoder is exactly a nonlinear block Gauss–Seidel iteration bent on finding the solution to the constrained optimization problem

$$(\boldsymbol{\lambda}_U^*,\boldsymbol{\lambda}_T^*)=\arg\max_{(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)\in\mathcal{C}}\log(p(\mathbf{r}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)).$$

In particular, the turbo decoder stationary points are in a one to one correspondence with the critical points of the Lagrangian of this optimization problem with a Lagrange multiplier of -1. The turbo decoder is then a nonlinear block Gauss–Seidel iteration on the gradient of this Lagrangian.

Proof: For the proof, form the Lagrangian

$$\begin{aligned}\mathsf{L}(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)&=\log(p(\mathbf{r}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T))\\&\quad+\mu(\log(\mathsf{Pr}[\boldsymbol{\xi}=\boldsymbol{\chi}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T])-\log(c))\\&=-\psi(\mathbf{B}\boldsymbol{\lambda}_U)-\psi(\mathbf{B}\boldsymbol{\lambda}_T)+\psi(\mathbf{B}\boldsymbol{\lambda}_U+\boldsymbol{\theta}_0)\\&\quad+\psi(\mathbf{B}\boldsymbol{\lambda}_T+\boldsymbol{\theta}_1)\\&\quad-\mu(\psi(\mathbf{B}\boldsymbol{\lambda}_U)+\psi(\mathbf{B}\boldsymbol{\lambda}_T))\\&\quad+\mu\psi(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)).\end{aligned}$$

Taking its gradient gives

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_U}\mathsf{L}&=-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U}+\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U+\boldsymbol{\theta}_0}\\&\quad+\mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U}+\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}\right)\\\nabla_{\boldsymbol{\lambda}_T}\mathsf{L}&=-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T}+\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T+\boldsymbol{\theta}_1}\\&\quad+\mu\left(-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T}+\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}\right).\end{aligned}$$

Selecting a Lagrange multiplier of $\mu=-1$, we have

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_U}\mathsf{L}&=\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U+\boldsymbol{\theta}_0}-\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}\\\nabla_{\boldsymbol{\lambda}_T}\mathsf{L}&=\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T+\boldsymbol{\theta}_1}-\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}.\end{aligned}$$

If we break this system of equations into two parts

$$\begin{aligned}\mathsf{F}_0(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)&=-\nabla_{\boldsymbol{\lambda}_T}\mathsf{L}\\&=\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_T+\boldsymbol{\theta}_1}\qquad(11)\\\mathsf{F}_1(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T)&=-\nabla_{\boldsymbol{\lambda}_U}\mathsf{L}\\&=\mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T)}-\mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U+\boldsymbol{\theta}_0}\qquad(12)\end{aligned}$$

then we can see that the turbo decoder solves F_0 for $\boldsymbol{\lambda}_U$ given a fixed $\boldsymbol{\lambda}_T=\boldsymbol{\lambda}_T^{(k)}$ to get $\boldsymbol{\lambda}_U^{(k)}$, and then solves F_1 for $\boldsymbol{\lambda}_T$ given a fixed $\boldsymbol{\lambda}_U=\boldsymbol{\lambda}_U^{(k)}$ to get $\boldsymbol{\lambda}_T^{(k+1)}$, that is

$$\begin{aligned}\boldsymbol{\lambda}_U^{(k)}&=\boldsymbol{\lambda}_U\text{ such that }\mathsf{F}_0\left(\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T^{(k)}\right)=\mathbf{0}\\\boldsymbol{\lambda}_T^{(k+1)}&=\boldsymbol{\lambda}_T\text{ such that }\mathsf{F}_1\left(\boldsymbol{\lambda}_U^{(k)},\boldsymbol{\lambda}_T\right)=\mathbf{0}.\end{aligned}$$

This is exactly the form of a nonlinear block Gauss–Seidel iteration. Furthermore, the system of equations it is trying to solve are the necessary conditions for finding a solution of the provided constrained optimization problem, which are

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_U}\mathsf{L}&=\mathbf{0}\\\nabla_{\boldsymbol{\lambda}_T}\mathsf{L}&=\mathbf{0}\end{aligned}$$

subject to a Lagrange multiplier of $\mu=-1$. □

An important fact which sets this formulation apart from most constrained optimization problems is that one does not get to specify the value of $c=\mathsf{Pr}[\boldsymbol{\xi}=\boldsymbol{\chi}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T]$ before decoding; rather, a Lagrange multiplier $\mu=-1$ is selected which then results in a value of the constraint. If the turbo decoder converges, one may evaluate the constraint function $-\psi(\mathbf{B}\boldsymbol{\lambda}_U)-\psi(\mathbf{B}\boldsymbol{\lambda}_T)+\psi(\mathbf{B}(\boldsymbol{\lambda}_U+\boldsymbol{\lambda}_T))$ at the convergent $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$ to get the $\log(c)$ for which the turbo decoder stationary points are critical points of the optimization problem. Note also that if we have $c=\mathsf{Pr}[\boldsymbol{\xi}=\boldsymbol{\chi}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T]=1$ so that $\log(c)=0$, then the globally optimal solution to the optimization problem is actually the MLSD. This then suggests that for convergent values of $\log(c)$ close to 0, the turbo decoder will provide a detection close to a critical point of the wordwise likelihood function, provided continuity in $\log(c)$ of the solutions to the optimization problem.

While it may seem odd to select a Lagrange multiplier $\mu=-1$ rather than a value of the constraint $\log(c)$, we note that this too has a possible intuitive interpretation within the context of the turbo decoder, although it is less rigorous than other developments in this paper. In particular, selecting $\mu=-1$ requires that the gradient of the constraint $\nabla\log(c)$ be equal to the gradient of the approximate likelihood function $\nabla\log(p(\mathbf{r}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T))$ at a turbo decoder stationary point $(\boldsymbol{\lambda}_U^*,\boldsymbol{\lambda}_T^*)$. Thus, the change in the likelihood function from its value at a turbo decoder stationary point $\log(\mathbf{p}(\mathbf{r}\,|\,\boldsymbol{\lambda}_U,\boldsymbol{\lambda}_T))-\log(p(\mathbf{r}\,|\,\boldsymbol{\lambda}_U^*,\boldsymbol{\lambda}_T^*))$ is, to first order, equal to the change in the constraint from its value at that turbo decoder stationary point $\log(c(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T)) - \log(c(\boldsymbol{\lambda}_U^*, \boldsymbol{\lambda}_T^*))$ within a neighborhood of that stationary point. When the turbo decoder is used is practice, one takes decisions on its convergent values $\boldsymbol{\lambda}_U^* + \boldsymbol{\lambda}_T^*$ to get the bits ($\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\chi}} \in \{0,1\}^N$) which are passed on to the rest of the receiver. We use $\boldsymbol{\lambda}_U^* + \boldsymbol{\lambda}_T^*$, because the ith element of this vector is the log likelihood ratio associated with the bitwise probability $\Pr[\xi_i = \boldsymbol{\chi} i_i = 1 \,|\, \boldsymbol{\xi} = \boldsymbol{\chi}, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T]$ and we condition here on $\boldsymbol{\xi} = \boldsymbol{\chi}$ because we know it to be true and wish to use that fact before passing our decisions to the rest of the receiver. Taking these decisions (and thus setting the new $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$'s equal to the likelihood ratios associated with these decisions $\hat{\boldsymbol{\lambda}_U}, \hat{\boldsymbol{\lambda}_T} = \infty \mathrm{sign}(\boldsymbol{\lambda}_U^* + \boldsymbol{\lambda}_T^*) \in \{\pm\infty\}^N$) increases the value of the constraint probability to one, so that $\log(c(\hat{\boldsymbol{\lambda}_U}, \hat{\boldsymbol{\lambda}_T})) = 0$. Choosing the Lagrange multiplier $\mu = -1$, then, ensures that this largest possible increase in the constraint is accompanied with an equally large increase in the approximated likelihood function to first order. The sign of the Lagrange multiplier ensures that increasing the constraint by taking decisions increases the approximate likelihood to first order, and the magnitude reflects that it is equally important to have a large change in the constraint as it is to have a large change in the likelihood. Although other interpretations of selecting $\mu = -1$ could be offered, the theorem and the intuitive nature of the objective functions and constraints remain true. It is also interesting to note that, with $\mu = -1$ in the constrained Lagrangian, we recover the function used in [26, Th. 4], which was obtained without recourse to likelihood concepts. An investigation of other values for the Lagrange multiplier as well as other justifications for choosing -1 would be an interesting topic for further research. Along these lines, for readers familiar with the Bethe approximation to the Gibbs free energy from statistical physics [18], [17], it may be noted that when evaluated at the turbo decoder stationary points, $\mathsf{L}\,|_{\mu=-1}$ attains the same values as the Bethe approximation to the Gibbs free energy (see [17, p. 1795]), although for message values that are not stationary points the equality does not hold. In fact, a pseudoduality relationship between the two optimality frameworks, that is the constrained optimization problem from Theorem 1 and the Bethe free energy, can be shown [21].

Given the importance of selecting the Lagrange multiplier $\mu = -1$ in order to get the turbo decoder stationary points, we henceforth select $\mu = -1$ when we refer to the Lagrangian. Because the condition that the gradient of the Lagrangian is equal to zero is necessary, but not always sufficient, for a point to be the global optimum, we must characterize the type of critical points which are possible. In particular, we wish to know whether or not the critical points of the Lagrangian are at least local maxima of the constrained optimization problem. Generally speaking one can converge to either a maximum, minimum, or saddle point, although if one replaces the Lagrangian with the expectation of the Lagrangian over the received data one can guarantee that there is only a global maximum.

Theorem 2 (Critical Point Characterization): The expected Lagrangian has only one critical point which is a maximum of the expected constrained optimization problem. Here, the expectation is taken over the received information $\mathbf{r}$ using the approximate likelihood function for $\mathbf{r}$ given $\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T$ under the false independence assumption.

Proof: To see this, consider the value of the Lagrangian within the constraint space

$$\begin{aligned}\mathsf{L} &= \log(c) - \psi(\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)) + \psi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) \\ &\quad + \psi(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1) \\ &= -\psi(\mathbf{B}\boldsymbol{\lambda}_U) - \psi(\mathbf{B}\boldsymbol{\lambda}_T) + \psi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) \\ &\quad + \psi(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)\end{aligned}$$

where in the latter equation we substitute in the constraint $\log(c) - \psi(\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)) = -\psi(\mathbf{B}\boldsymbol{\lambda}_U) - \psi(\mathbf{B}\boldsymbol{\lambda}_T)$. Now, note from this that L thus is the sum of two log likelihood functions within the constraint space

$$\begin{aligned}\mathsf{L} &= (-\psi(\mathbf{B}\boldsymbol{\lambda}_U) + \psi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0)) \\ &\quad + (-\psi(\mathbf{B}\boldsymbol{\lambda}_T) + \psi(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)) \\ &= \log(p(\mathbf{r}_s, \mathbf{r}_0 \,|\, \boldsymbol{\lambda}_U)) + \log(p(\mathbf{r}_1 \,|\, \boldsymbol{\lambda}_T)).\end{aligned}$$

This then implies that the second derivative of the Lagrangian within the constraint set has a mean which is the negative of the Fisher information matrix, call it $\mathbb{I}$, since the Fisher information matrix is defined as

$$\begin{aligned}\mathbb{I} &= -\begin{pmatrix}\mathbf{E} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}\end{pmatrix} \\ \mathbf{E} &:= \int \nabla^2_{\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_U} \{\log(p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T))\} p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) d\mathbf{r} \\ \mathbf{V} &:= \int \nabla^2_{\boldsymbol{\lambda}_T, \boldsymbol{\lambda}_T} \{\log(p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T))\} p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) d\mathbf{r}\end{aligned}$$

where we have used $\nabla^2\{\cdot\}$ here to denote the operator which takes a function to its Hessian matrix of second-order partial derivatives. The well-known fact, then, that the Fisher information matrix is positive semidefinite, then shows that the expectation of the Hessian matrix of the Lagrangian L is negative semi-definite. This implies then, by Theorem 4.5 of [23], the function $\mathsf{E}[\mathsf{L}]$ is concave, where E denotes expectation with respect to $p(\mathbf{r} \,|\, \boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T)$, and thus has a unique maximum. □

Two important distinctions are made in the previous theorem. First of all, while the expected value of the Lagrangian is concave within the constraint space, and thus has a unique maximum, we are not guaranteed that for a particular sample $\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1$ there is only one solution to $\nabla \mathsf{L} = \mathbf{0}$. In fact, Fig. 7 shows that, even for $N = 2$, it is possible, depending on $\mathbf{r}_s, \mathbf{r}_0, \mathbf{r}_1$ for this convergent point to be either a maximum or minimum. Another important distinction is that while the expected value of the Lagrangian is concave within the constraint space, it is not necessarily concave outside of the constraint space. In fact, one can show that outside of the constraint space

$$\begin{aligned}\nabla^2_{\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_U} \mathsf{L} &= \mathbf{P}_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} - \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} \mathbf{p}^T_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} \\ &\quad - \left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} \mathbf{p}^T_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)}\right) \\ \nabla^2_{\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T} \mathsf{L} &= -\left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} \mathbf{p}^T_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)}\right) \\ \nabla^2_{\boldsymbol{\lambda}_T, \boldsymbol{\lambda}_T} \mathsf{L} &= \mathbf{P}_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} - \mathbf{p}_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} \mathbf{p}^T_{\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0} \\ &\quad - \left(\mathbf{P}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} - \mathbf{p}_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)} \mathbf{p}^T_{\mathbf{B}(\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T)}\right)\end{aligned}$$

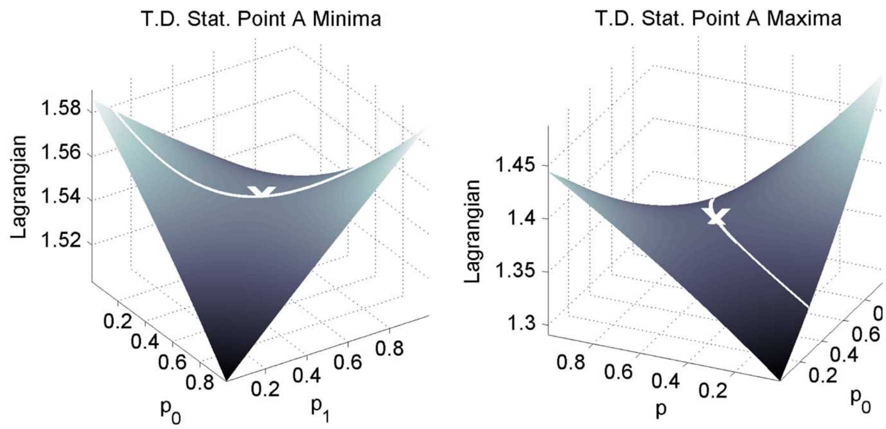

Fig. 7. L for two different sample values of $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ for $N = 2$ bits and simple parity check codes. Here, the line indicates $\mathcal{C}$, and x marks the spot to which the turbo decoder converges given an initialization at $(\frac{1}{2}, \frac{1}{2})$. The p_0 and p_1 axes are the bitwise marginal probabilities associated with $\boldsymbol{\lambda}_U$, and we are always selecting $\boldsymbol{\lambda}_T = \pi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) - \boldsymbol{\lambda}_U$. In one instance, we have converged to a local minimum along $\mathcal{C}$, and in another to a local maximum along $\mathcal{C}$.

where $\mathbf{P}_{\boldsymbol{\theta}}$ is the matrix whose i, jth entry is the probability that both the ith and jth bits are one, according to the wordwise measure whose log coordinates are $\boldsymbol{\theta}$. From this we can see that at a Turbo decoder stationary point, the diagonal elements of $\nabla^2 \mathsf{L}$ are all zero. Thus, the trace of the Hessian matrix is zero, and, provided the Hessian is not identically zero, the stationary points are saddle points of the Lagrangian when one does not restrict oneself to the constraint space.

To conclude our remarks concerning the optimality of the turbo decoder stationary points, we wish to emphasize the new perspective that the constrained optimization problem can provide. Many researchers believe that the turbo decoder approximates the maximum-likelihood bitwise solutions (see, e.g., [17], [18] or other literature about loopy belief propagation), partially because the elegant theory for belief propagation considers the special case when the graph has no loops [6], for which the belief propagation does calculate the maximum-likelihood bitwise solutions. The use of the forward backward algorithm [27], which would calculate the MLBD given only $\mathbf{r}_s, \mathbf{r}_0$ or $\mathbf{r}_s, \mathbf{r}_1$ further confuses the issue. These results do not apply to the turbo decoder, or even to the soft decoding of finite block length LDPC codes, due to loops in the factor graphs of both of these decoders. Here, we have shown that the turbo decoder may be interpreted as a constrained MLSD. The constraints generally bias the stationary points away from the exact MLSD.

IV. Convergence Analysis

In Theorem 1, we observed that the turbo decoder may be understood as an iterative procedure to seek a critical point of the Lagrangian L with a Lagrange multiplier $\mu = -1$.

$$\text{Choose } \boldsymbol{\lambda}_T^{(k)} \text{ such that } \nabla_{\boldsymbol{\lambda}_U} \mathsf{L}\left(\boldsymbol{\lambda}_U^{(k)}, \boldsymbol{\lambda}_T\right) = \mathbf{0} \quad (13)$$

$$\text{Choose } \boldsymbol{\lambda}_U^{(k+1)} \text{ such that } \nabla_{\boldsymbol{\lambda}_T} \mathsf{L}\left(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T^{(k)}\right) = \mathbf{0} \quad (14)$$

We then noted that such an iterative procedure may be connected with the Gauss–Seidel iteration for solving the system $\nabla \mathsf{L}(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) = \mathbf{0}$. From here on, to compact notation, and to continue relation with the F_0 and F_1 notation used in the proof of Theorem 1, we will denote $\nabla \mathsf{L}$ by F. Here we will elaborate on the connection to the Gauss–Seidel iteration, and use it to gain some convergence conditions for the turbo decoder. Before we do so, it is important to note that there have been numerous previous works which discuss local stability of the turbo decoder via first order Taylor approximation techniques, see for instance the information geometry based work [9], [26], [10]. Global stability in turn, has been discussed in the context of belief propagation via either convexity of the Bethe free energy or loop-less factor graph cases, see e.g., [6], [28]. These and other expositions discussing convergence have already been mentioned in the introduction. The conditions for global stability that we shall discuss here, however, are derived uniquely via a connection to the nonlinear block Gauss–Seidel iteration, and in fact are mathematically different from those mentioned previously and in the introduction.

The Gauss–Seidel procedure has been discussed most widely for linear systems of equations [29, pp. 480–483], [30, pp. 251–257], [31, pp. 510–511], but can be generalized to nonlinear systems of equations as well [32, pp. 131–133; pp. 185–197], [33, p. 225]. In particular, the convergence of nonlinear Gauss–Seidel methods received some significant attention in the numerical analysis literature during the early 1970s. Relevant references include [34], [35], and [36], from which our major result is adapted.

One of the major difficulties encountered when applying the typical convergence theorems for nonlinear block Gauss–Seidel iterations to the iteration (13) and (14), as in [1], is that the parameters are drawn from a different domain than the range of $\nabla \mathsf{L}$. In particular, $\nabla \mathsf{L}$ is a difference between bitwise probabilities and thus is necessarily bounded within $[-1, 1]^N$. The variables that are being solved for, $\boldsymbol{\lambda}_U$ and $\boldsymbol{\lambda}_T$, however, are drawn from $\mathbb{R}^N$. This foiled unmodified application of theorems from [36], since those theorems needed surjectivity of $\nabla \mathsf{L}$ onto $\mathbb{R}^N$. The approach taken in [1], was then to modify the theorems from [36], adding extra conditions so that the surjectivity was no longer needed. Unfortunately, this resulted in conditions which required the user of the theorem to check a containment relation of two parametrically defined sets in N dimensions, where N was the block size of the turbo decoder. It seemed doubtful that the theory would be useful without extra theory to determine when the conditions would be satisfied. Still, the reader interested in characterizing the regions of convergence for the case when the turbo decoder admits multiple fixed points may consult [1].

We take a different approach here, focussing instead on conditions on the likelihood functions of the two component decoders that yield global convergence of the turbo decoder from any initialization to a single fixed point. We transform the domain of the update equation $\nabla \mathsf{L} = \mathbf{0}$ from that of the difference between two bitwise probabilities to the difference between two log bitwise probability ratios. In other words, rather than considering the system (11) and (12) whose nonlinear block Gauss–Seidel iteration is embodied by the recursion (9) and (10}), we use the system

$$\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T - \pi(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) = \mathbf{0} \quad (15)$$
$$\boldsymbol{\lambda}_U + \boldsymbol{\lambda}_T - \pi(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1) = \mathbf{0} \quad (16)$$

whose nonlinear block Gauss–Seidel iteration is embodied by (7) and (8) and is thus equivalent to the recursion defined by (9) and (10). Since the range of the system in consideration is now $\mathbb{R}^{2N}$, it is easier to apply the results from [36], since surjectivity can now hold. In particular, we have the following theorem, in which we use the componentwise ordering, so that $\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \forall i$.

Theorem 3 (Global Convergence of the Turbo Decoder): Define the measures q whose log coordinates are $\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0$ and r whose log coordinates are $\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1$. Define the matrices $\mathbf{C}$ and $\mathbf{G}$ whose elements are

$$[\mathbf{C}]_{i,j} = \mathsf{q}[\xi_j = 1 \,|\, \xi_i = 1] - \mathsf{q}[\xi_j = 1 \,|\, \xi_i = 0]$$
$$\text{and}$$
$$[\mathbf{G}]_{i,j} = \mathsf{r}[\xi_j = 1 \,|\, \xi_i = 1] - \mathsf{r}[\xi_j = 1 \,|\, \xi_i = 0]$$

and define the matrix

$$\boldsymbol{\Lambda} := \begin{bmatrix} \mathbf{I} & \mathbf{I} - \mathbf{C} \\ \mathbf{I} - \mathbf{G} & \mathbf{I} \end{bmatrix}.$$

Define the set $\mathcal{D} \subseteq \mathbb{R}^{2^N} \times \mathbb{R}^{2^N}$ to be the set of $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)$ for which $\boldsymbol{\Lambda}$ is an M-matrix, or equivalently has non positive off diagonal elements and all real eigenvalues positive [37]–[40].[3] Then, if $(\boldsymbol{\theta}_0(\mathbf{r}), \boldsymbol{\theta}_1(\mathbf{r})) \in \mathcal{D}$ the turbo decoder converges to a unique solution from any initialization $(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) \in \mathbb{R}^{2N}$.

Proof: The outline of our proof is as follows.

- The fact that $\boldsymbol{\Lambda}$ is an M-matrix shows by [36, Th. 3.6] that F is an M-function everywhere in $\mathbb{R}^{2N}$.
- We can linearly lower bound F componentwise, and thus use [36, Th. 4.4] to prove surjectivity of F.
- We finally have that F is a surjective M-function. This allows us to apply [36, Th. 6.4] to prove convergence.

First, calculate the Jacobian $\nabla \mathsf{F}$

$$\nabla \mathsf{F} = \begin{bmatrix} \mathbf{I} & \mathbf{I} - \mathbf{C} \\ \mathbf{I} - \mathbf{G} & \mathbf{I} \end{bmatrix}$$

where

$$[\mathbf{C}]_{i,j} = \mathsf{q}[\xi_j = 1 \,|\, \xi_i = 1] - \mathsf{q}[\xi_j = 1 \,|\, \xi_i = 0]$$

and

$$[\mathbf{G}]_{i,j} = \mathsf{r}[\xi_j = 1 \,|\, \xi_i = 1] - \mathsf{r}[\xi_j = 1 \,|\, \xi_i = 0].$$

We now see that the conditions in the theorem force $\nabla \mathsf{F}$ to be a M-matrix for all $\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T \in \mathbb{R}^N$. Then Thm. 3.6 of [36] shows that F is an M-function. Next, consider that

$$\begin{aligned} -\log\left(\frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0)}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0)}\right) &\geq -\log \frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda}_U)}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda}_U)} + \mathbf{1}\log\left(\frac{\alpha_0}{\beta_0}\right) \\ &\geq -\boldsymbol{\lambda}_U + \log\left(\frac{\alpha_0}{\beta_0}\right) \end{aligned}$$

where

$$\alpha_0 = \min_i [\exp(\boldsymbol{\theta}_0)]_i, \quad \beta_0 = \max_i [\exp(\boldsymbol{\theta}_0)]_i$$

and we used the componentwise division notation so that $\frac{\mathbf{a}}{\mathbf{b}}$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ is the vector whose ith element is a_i/b_i. Similarly, we have

$$-\log\left(\frac{\mathbf{B}^T \exp(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)}{(\mathbf{1} - \mathbf{B})^T \exp(\mathbf{B}\boldsymbol{\lambda}_T + \boldsymbol{\theta}_1)}\right) \geq -\boldsymbol{\lambda}_T + \mathbf{1}\log\left(\frac{\alpha_1}{\beta_1}\right)$$

where

$$\alpha_1 = \min_i [\exp(\boldsymbol{\theta}_1)]_i, \quad \beta_1 = \max_i [\exp(\boldsymbol{\theta}_1)]_i.$$

Putting these two facts together, we have

$$\mathsf{F}(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) \geq \begin{bmatrix} \boldsymbol{\lambda}_T + \mathbf{1}\log\left(\frac{\alpha_0}{\beta_0}\right) \\ \boldsymbol{\lambda}_U + \mathbf{1}\log\left(\frac{\alpha_1}{\beta_1}\right) \end{bmatrix}$$

which shows that

$$\lim_{k \to \infty} \|\mathsf{F}(\boldsymbol{\lambda}_k)\| = \infty$$

whenever $\|\boldsymbol{\lambda}_k\| \to \infty$ and either $\boldsymbol{\lambda}_k \geq \boldsymbol{\lambda}_{k+1}$ or $\boldsymbol{\lambda}_{k+1} \geq \boldsymbol{\lambda}_k$. This then implies that F is surjective by [36, Th. 4.4]. Thus, F is a surjective M-function, and Thm. 6.4 of [36] proves that the Gauss–Seidel block iteration on F is globally convergent. □

To illustrate the practical applicability of this theorem, we consider now a very simple case in which the globally convergent region $\mathcal{D}$ can be computed exactly. In particular consider the case that $N = 2$, then $\boldsymbol{\Lambda}$ takes the form

$$\boldsymbol{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & -\alpha \\ 0 & 1 & -\beta & 0 \\ 0 & -\gamma & 1 & 0 \\ -\sigma & 0 & 0 & 1 \end{bmatrix}$$

[3]See [37]–[40] for conditions equivalent to the definition of an M-matrix.

where

$$\alpha = \mathsf{q}[\xi_1 = 1 \,|\, \xi_0 = 1] - \mathsf{q}[\xi_1 = 1 \,|\, \xi_0 = 0]$$
$$\beta = \mathsf{q}[\xi_0 = 1 \,|\, \xi_1 = 1] - \mathsf{q}[\xi_0 = 1 \,|\, \xi_1 = 0]$$
$$\gamma = \mathsf{r}[\xi_1 = 1 \,|\, \xi_0 = 1] - \mathsf{r}[\xi_1 = 1 \,|\, \xi_0 = 0]$$
$$\sigma = \mathsf{r}[\xi_0 = 1 \,|\, \xi_1 = 1] - \mathsf{r}[\xi_0 = 1 \,|\, \xi_1 = 0].$$

We must calculate the eigenvalues of $\mathbf{\Lambda}$. The characteristic function of $\mathbf{\Lambda}$ is

$$\begin{vmatrix} (1-\lambda) & 0 & 0 & -\alpha \\ 0 & (1-\lambda) & -\beta & 0 \\ 0 & -\gamma & (1-\lambda) & 0 \\ -\sigma & 0 & 0 & (1-\lambda) \end{vmatrix} = [(1-\lambda)^2 - \alpha\sigma][(1-\lambda)^2 - \beta\gamma].$$

Setting this equal to zero allows us to solve for the eigenvalues

$$\lambda = 1 \pm \sqrt{\alpha\sigma}, 1 \pm \sqrt{\beta\gamma}$$

Now, we see that as long as $\alpha, \sigma, \beta, \gamma \geq 0$ we have that all of the eigenvalues are both real and positive, since $\alpha, \sigma, \beta, \gamma$ are the difference between two probabilities and thus have modulus less than one. It remains only to require that the off diagonal elements of this matrix are nonpositive. If we define

$$\eta = [\eta_0, \eta_1, \eta_2, \eta_3]^T := \frac{\exp(\boldsymbol{\theta}_0)}{\|\exp(\boldsymbol{\theta}_0)\|_1}.$$

Then we see that the condition that $\alpha, \beta \geq 0$ is equivalent to

$$\frac{\eta_3}{\eta_1 + \eta_3} \geq \frac{\eta_2}{\eta_0 + \eta_2}, \quad \frac{\eta_3}{\eta_2 + \eta_3} \geq \frac{\eta_1}{\eta_0 + \eta_1}$$

which holds if and only if (iff)

$$\eta_1\eta_2 \leq \eta_0\eta_3 \tag{17}$$

Denoting $\boldsymbol{\theta}_0 = [0, \theta_1, \theta_2, \theta_3]$, we see that (17) holds if and only if

$$\theta_1 + \theta_2 - \theta_3 = [0, 1, 1, -1]\boldsymbol{\theta}_0 \leq 0 \tag{18}$$

The same requirement (18) is necessary and sufficient for γ, σ to be negative by replacing $\boldsymbol{\theta}_1 = [0, \theta_0, \theta_1, \theta_2]$. Now, suppose $\boldsymbol{\theta}_0$ has elements which satisfy (18), and consider all log domain pmfs of the form $\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0$ for $\boldsymbol{\lambda}_U \in \mathbb{R}^2$. These will satisfy

$$[0, 1, 1, -1](\mathbf{B}\boldsymbol{\lambda}_U + \boldsymbol{\theta}_0) = [0, 0]\boldsymbol{\lambda}_U + [0, 1, 1, -1]\boldsymbol{\theta}_0 \leq 0$$

We can thus see that $\mathcal{D}$ is exactly characterized as

$$\mathcal{D} := \left\{ (\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \in \mathbb{R}^4 \times \mathbb{R}^4 \,\middle|\, \begin{array}{l} [0,1,1,-1]\boldsymbol{\theta}_0 \leq 0, \\ [0,1,1,-1]\boldsymbol{\theta}_1 \leq 0 \end{array} \right\}$$

Thus, as long as $\mathbf{r}$ and the structure of the code yield a $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \in \mathcal{D}$, the turbo decoder converges to a unique fixed point from any initialization $(\boldsymbol{\lambda}_U, \boldsymbol{\lambda}_T) \in \mathbb{R}^N \times \mathbb{R}^N$.

Continuing the example, suppose that we are using a parallel concatenated code, with each component code a simple parity check code with two systematic bits and one parity check bit and we are transmitting using binary phase shift keying (BPSK) over a additive white gaussian noise (AWGN) channel, and assume the all zero codeword was transmitted. This yields

$$\boldsymbol{\theta}_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \boldsymbol{\lambda}_0, \quad \boldsymbol{\theta}_1 = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \boldsymbol{\lambda}_1$$

where $\boldsymbol{\lambda}_0$ are the raw channel log likelihood ratios for systematic bits and the parity check bit of the first code $(\mathbf{r}_s, \mathbf{r}_0)$ and $\boldsymbol{\lambda}_1$ is the raw channel log likelihood ratio for the parity check bit of the second code. The condition that $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$ lie in $\mathcal{D}$ is then equivalent to the requirement that

$$[0, 1, 1, -1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \boldsymbol{\lambda}_0 = [0, 0, 2]\boldsymbol{\lambda}_0 \leq 0$$

and $2\boldsymbol{\lambda}_1 \leq 0$. This establishes that the turbo decoder for this two bit parallel concatenated parity check code will converge to a fixed point from any initialization as long as the channel does not cause any errors in the parity check bits (i.e., as long as the log likelihood ratios associated with the parity check bits are less than or equal to zero).

It may seem odd that this result does not depend on the systematic bits, but this has a perfectly reasonable explanation. This result does not depend on the systematic bits because we have required the turbo decoder to converge from any initialization of the a priori log likelihood ratios for the systematic bits. This is equivalent to requiring the turbo decoder to converge regardless of the received raw channel log likelihood ratios for the systematic bits. This shows that, generally speaking, any set of sufficient conditions for global convergence of the parallel turbo decoder must thus be insensitive to the received channel likelihood values for the systematic bits. Thus it is natural that the raw channel log likelihood ratios for the systematic bits did not appear when checking $(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1) \in \mathcal{D}$.

Given the wide range of nonlinear dynamical phenomena that the turbo decoder has been demonstrated to posses (e.g., limit cycles [41] and even chaotic behavior [42]), it is important to characterize the cases in which the turbo decoder avoids this bad behavior and instead converges to a single fixed point. The theorem we have adapted from nonlinear block Gauss–Seidel iterations gives us sufficient conditions for this robust behavior, which we have demonstrated to be intuitively meaningful for a particularly simple, yet nontrivial, 2-systematic bit case. It remains to be seen whether or not the conditions of the theorem can be interpreted in a similar manner for realistic codeword lengths.

V. Conclusion

The turbo decoder is a suboptimal heuristic method which was developed through simulation. Although researchers have been able to characterize its convergence behavior and performance in the asymptotically large block length and cycle-free factor graph cases, up until now the sense of optimality of its stationary points relative to the desired maximum-likelihood sequence detector design, the mechanism behind its convergence, and conditions under which it converged all remained less understood. We have shown that the turbo decoder stationary points are critical points of the constrained optimization problem of maximizing the log likelihood function for the received data under a false independence assumption for the messages for which the turbo decoder is exchanging soft information, subject to the constraint that the probability that the messages so chosen are the same is fixed. If this probability is 1, then the global maximum to this constrained optimization problem is the MLSD. Furthermore, we have shown that the turbo decoder is actually a nonlinear block Gauss–Seidel iteration on the system of necessary equations for this constrained optimization problem specified by Lagrange with a Lagrange multiplier of -1. Finally, by identifying the iterative method that was being used to find the solution to the necessary conditions, we identified the mechanism behind the turbo decoder's convergence. Using the existing theory for this convergence mechanism, we were able to determine sufficient conditions on the component likelihood functions for the convergence of the turbo decoder from any initialization, which we demonstrated to be intuitively reasonable for a particular simple parallel concatenated code.

REFERENCES

[1] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "A convergence proof for the turbo decoder as an instance of the Gauss-Seidel iteration," in *Proc. IEEE Int. Symp. Inf. Theory*, Adelaide, Australia, Sep. 2005.

[2] ——, "Turbo decoding as constrained optimization," in *Proc. 43rd Allerton Conf. Communication, Control and Computing*, Monticello, IL, Sep. 2005.

[3] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. ICC 93*, Geneva, Switzerland, May 1993, vol. 2, pp. 1064–1070.

[4] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, pp. 1727–1737, Oct. 2001.

[5] H. El Gamal and A. R. Hammons, Jr., "Analyzing the turbo decoder using the Gaussian approximation," *IEEE Trans. Inf. Theory*, vol. 47, pp. 671–686, Feb. 2001.

[6] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[7] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Sel. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.

[8] M. Moher and T. A. Gulliver, "Cross-entropy and iterative decoding," *IEEE Trans. Inf. Theory*, vol. 44, pp. 3097–3104, Nov. 1998.

[9] T. J. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inf. Theory*, vol. 46, pp. 9–23, Jan. 2000.

[10] S. Ikeda, T. Tanaka, and S. Amari, "Information geometry of turbo and low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1097–1114, Jun. 2004.

[11] B. Muquet, P. Duhamel, and M. de Courville, "Geometrical interpretations of iterative 'turbo' decoding," in *Proceedings ISIT*, June 2002.

[12] J. M. Walsh, P. A. Regalia, and C. R. Johnson, Jr., "A refined information geometric interpretation of turbo decoding," in *Proc. Int.. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, Mar. 2005.

[13] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Diego, CA: Morgan Kaufmann, 1988.

[14] P. Pakzad and V. Anantharam, "Belief propagation and statistical physics," in *Proc. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 2002.

[15] ——, "Estimation and marginalization using Kikuchi approximation methods," *Neural Comput.*, pp. 1836–1876, Aug. 2005.

[16] ——, "Kikuchi approximation method for joint decoding of LDPC codes and partial-response channels," *IEEE Trans. Commun.*, vol. 54, no. 7, Jul. 2006.

[17] S. Ikeda, T. Tanaka, and S. Amari, "Stochastic reasoning, free energy and information geometry," *Neural Comput.*, no. 16, pp. 1779–1810, 2004.

[18] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Trans. Inf. Theory*, no. 7, pp. 2282–2312, Jul. 2005.

[19] A. Montanari and N. Sourlas, "The statistical mechanics of turbo codes," *Eur. Phys. J. B*, no. 18, pp. 107–109, 2000.

[20] J. M. Walsh and P. A. Regalia, "Connecting belief propagation with maximum likelihood detection," in *Proc. Fourth Int. Symp. Turbo Codes*, Munich, Germany, Apr. 2006.

[21] J. M. Walsh, "Dual optimality frameworks for expectation propagation," in *Proc. IEEE Conf. Signal Processing Adv. Wireless Commun. (SPAWC)*, Cannes, France, Jun. 2006.

[22] J. M. Walsh, "Distributed Iterative Decoding and Estimation Via Expectation Propagation: Performance and Convergence," Ph.D. dissertation, Cornell University, Ithaca, NY, 2006.

[23] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton University Press, 1970.

[24] S. Amari, "Methods of information geometry," *AMS Transl. Math. Monogr.*, vol. 191, 2004.

[25] C. Berrou and A. Glavieux, "Near optimum error correction coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, pp. 1262–1271, Oct. 1996.

[26] S. Ikeda, T. Tanaka, and S. Amari, "Information geometrical framework for analyzing belief propagation decoder," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002.

[27] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, Mar. 1974.

[28] S. C. Tatikonda, "Convergence of the sum-product algorithm," in *Proc. 2003 Inf. Theory Workshop*, Paris, France, pp. 222–225.

[29] F. S. Acton, *Numerical Methods that Work*. New York: Harper and Row, 1970.

[30] J. L. Buchanan and P. R. Turner, *Numerical Methods and Analysis*. New York: McGraw Hill, 1992.

[31] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins University Press, 1996.

[32] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. New York: Athena Scientific, 1997.

[33] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.

[34] W. C. Rheinboldt, "On m-functions and their application to nonlinear Gauss-Seidel iterations and network flows," *J. Math. Anal. Applic.*, no. 32, pp. 274–307, 1971.

[35] J. J. Moré, "Nonlinear generalizations of matrix diagonal dominance with application to Gauss-Seidel iterations," *SIAM J. Numer. Anal.*, vol. 9, no. 2, pp. 357–378, Jun. 1972.

[36] W. C. Rheinboldt, "On classes of n-dimensional nonlinear mappings generalizing several types of matrices," in *Proc. Symp. Numerical Solutions Partial Differential Equations. II*, 1970, pp. 501–546.

[37] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1991.

[38] M. Fiedler, *Special Matrices and Their Applications in Numerical Mathematics*. Dordrecht, The Netherlands: Martinus Nijhoff, 1986.

[39] O. Axelsson, *Iterative Solution Methods*. Cambridge, U.K.: Cambridge University Press, 1994.

[40] M-matrix [Online]. Available: http://planetmath.org/encyclopedia/MMatrix.html

[41] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Inf. Theory*, vol. 47, pp. 699–722, Feb. 2001.

[42] Z. Tasev, L. Kocarev, and G. Maggio, "Bifurcations and chaos in the turbo decoding algorithm," in *Proc. 2003 Circuits and Systems, ISCAS'03*, May 2003, pp. III-120–III-123.