CONTRACTIVITY IN TURBO ITERATIONS

Phillip A. Regalia

Department of Communications, Image and Information Processing
CNRS/SAMOVAR UMR 5157
Institut National des Télécommunications/GET
9 rue Charles Fourier
91011 Evry cedex France
Phillip.Regalia@int-evry.fr

ABSTRACT

The turbo decoding algorithm has met with intense study over the past decade, in an attempt to harness the full power of the "turbo principle". Here we consider applying contractivity arguments to the turbo decoding algorithm, to study convergence even for short block lengths.

1. INTRODUCTION

The turbo decoding algorithm of a decade ago [1]–[3] provided a major breakthrough in reliable communications. Turbo decoding involves information exchange between two decoders, in which the extrinsic information values from one furnish the pseudo priors of the other. Attempts to understand this information exchange include EXtrinsic Information Transfer (or EXIT) charts [4], density evolution among successive iterations [5], and approximating the extrinsic information values by Gaussian random variables [6]; additional approaches include cross entropy minimization [7] and marginal projections [8]. Each approach lends valuable insight into the behavior of the turbo decoding algorithm, but many of the results appeal to asymptotic approximations involving the law of large numbers and/or the central limit theorem, and appear applicable only for rather long data blocks (e.g., $k = 2^{20}$ symbols per data packet in [4]); the results of such analyses do not give reliable indications of the behavior for shorter block lengths, which are of greater interest in two-way communications.

The development here explores contractivity (or passivity) arguments applied to turbo decoding, without appealing to large sample approximations. After a review of the turbo decoding algorithm, we isolate "desirable" stationary points in terms of valid code configurations, and then study Lipschitz constants in their vicinities. Analytic bounds on Lipschitz constants are often difficult to obtain, and turbo decoding presents no exception. Nonetheless, the approach developed here helps appreciate the performance losses incurred using shorter block lengths.

This work was supported by the Scientific Services Program of the US Army, contract no. DAAD19-02-D-0001.

2. TURBO DECODING ALGORITHM

Consider a binary (0 or 1) sequence $\xi = (\xi_1, \ldots, \xi_k)$ coded twice, to produce two codewords of n bits:

$$(\xi_1, \ldots, \xi_k, \eta_1, \ldots, \eta_{n-k}) \quad \text{and} \quad (\xi_1, \ldots, \xi_k, \zeta_1, \ldots, \zeta_{n-k})$$

Here $\{\eta_i\}$ and $\{\zeta_i\}$ are the binary parity-check bits furnished by either coder. The bits are converted to antipodal (± 1) form and transmitted over an additive white Gaussian noise channel to give

$$\begin{aligned} x_i &= (2\xi_i - 1) + b_{x,i} \quad i = 1, 2, \ldots, k \\ y_i &= (2\eta_i - 1) + b_{y,i} \quad i = 1, 2, \ldots, n-k \\ z_i &= (2\zeta_i - 1) + b_{z,i} \quad i = 1, 2, \ldots, n-k \end{aligned}$$

where the noise samples $b_{x,i}$, $b_{y,i}$ and $b_{z,i}$ are mutually independent, sharing a common variance σ^2.

The optimum decoding rule calculates the bitwise a posterior probability ratios

$$\begin{aligned} \frac{\Pr(\xi_i = 1 | \mathbf{x}, \mathbf{y}, \mathbf{z})}{\Pr(\xi_i = 0 | \mathbf{x}, \mathbf{y}, \mathbf{z})} &= \frac{\sum_{\xi : \xi_i = 1} \Pr(\xi | \mathbf{x}, \mathbf{y}, \mathbf{z})}{\sum_{\xi : \xi_i = 0} \Pr(\xi | \mathbf{x}, \mathbf{y}, \mathbf{z})} \quad i = 1, 2, \ldots, k, \\ &= \frac{\sum_{\xi : \xi_i = 1} p(\mathbf{x}|\xi)\, p(\mathbf{y}|\xi)\, p(\mathbf{z}|\xi) \Pr(\xi)}{\sum_{\xi : \xi_i = 0} p(\mathbf{x}|\xi)\, p(\mathbf{y}|\xi)\, p(\mathbf{z}|\xi) \Pr(\xi)} \end{aligned} \tag{1}$$

involving the a priori probability function $\Pr(\xi)$ and the three likelihood functions $p(\mathbf{x}|\xi)$, $p(\mathbf{y}|\xi)$ and $p(\mathbf{z}|\xi)$. The decoding complexity grow exponentially with the block

length k, since there are 2^k evaluations of $\xi = (\xi_1, \ldots, \xi_k)$ involved in the likelihood functions.

If we instead consider only $\mathbf{x}$ and $\mathbf{y}$, or only $\mathbf{x}$ and $\mathbf{z}$, then two decoding rules using only partial information become

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{y})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{y})} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)\,p(\mathbf{y}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)\,p(\mathbf{y}|\xi)\Pr(\xi)} \quad (2)$$

$$\frac{\Pr(\xi_i = 1|\mathbf{x},\mathbf{z})}{\Pr(\xi_i = 0|\mathbf{x},\mathbf{z})} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)\,p(\mathbf{z}|\xi)\Pr(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)\,p(\mathbf{z}|\xi)\Pr(\xi)} \quad (3)$$

Using convolutional constituent encoders, $\mathbf{x}$ and $\mathbf{y}$ form a Markov chain, as do $\mathbf{x}$ and $\mathbf{z}$; either decoding expression can be reduced to a complexity linear in the block length k by using the forward-backward algorithm from [9].

Turbo decoding amounts to usurping iteratively the a priori probability function $\Pr(\xi)$ in (2) [resp., (3)] by a function which approximates $p(\mathbf{z}|\xi)$ [resp., $p(\mathbf{y}|\xi)$], such that either expression will then approximate (1) evaluated for uniform $\Pr(\xi)$. In particular, let

$$T(\xi) = T_1(\xi_1)\cdots T_k(\xi_k) \quad \text{and} \quad U(\xi) = U_1(\xi_1)\cdots U_k(\xi_k)$$

be two factorable probability functions (to be specified shortly) which usurp the position reserved for $\Pr(\xi)$ in (3) and (2), respectively. Since these functions factor into the products of their marginals, as does

$$\begin{aligned} p(\mathbf{x}|\xi) &= \frac{1}{(\sqrt{2\pi}\sigma)^k}\exp\Big(\sum_{i=1}^{k}\frac{-(x_i-(2\xi_i-1))^2}{2\sigma^2}\Big) \\ &= \prod_{i=1}^{k}\frac{1}{\sqrt{2\pi}\,\sigma}\exp\Big(\frac{-(x_i-(2\xi_i-1))^2}{2\sigma^2}\Big) \\ &= p(x_1|\xi_1)\,p(x_2|\xi_2)\cdots p(x_k|\xi_k) \end{aligned}$$

the evaluation (2) exhibits common factors in the numerator and denominator, leading to the revised form

$$\frac{\sum_{\xi:\xi_i=1} p(\mathbf{x}|\xi)\,p(\mathbf{y}|\xi)\,U(\xi)}{\sum_{\xi:\xi_i=0} p(\mathbf{x}|\xi)\,p(\mathbf{y}|\xi)\,U(\xi)} = \frac{p(x_i|\xi_i=1)\,U_i(\xi_i=1)}{p(x_i|\xi_i=0)\,U_i(\xi_i=0)}\,\frac{\overbrace{\sum_{\xi:\xi_i=1} p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,U_j(\xi_j)}^{\text{extrinsic information}}}{\sum_{\xi:\xi_i=0} p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,U_j(\xi_j)}$$

The overbraced term gives the extrinsic information value, and we choose the ratio $T_i(\xi_i=1)/T_i(\xi_i=0)$ to match this value, for each i. We then replace $\Pr(\xi)$ in the second decoder (3) by $T(\xi)$; the evaluation factors analogously upon replacing $p(\mathbf{y}|\xi)$ by $p(\mathbf{z}|\xi)$. The new extrinsic information values determine the ratios $U_i(\xi_i=1)/U_i(\xi_i=0)$ which replace the a priori probability values $\Pr(\xi)$ in the first decoder (2), and the process iterates. By letting a superscript (m) denote an iteration index, the coupling of decoders takes the form

$$\frac{T_i^{(m)}(1)}{T_i^{(m)}(0)} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,U_j^{(m)}(\xi_j)}{\sum_{\xi:\xi_i=0} p(\mathbf{y}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,U_j^{(m)}(\xi_j)} \quad (4)$$

$$\frac{U_i^{(m+1)}(1)}{U_i^{(m+1)}(0)} = \frac{\sum_{\xi:\xi_i=1} p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,T_j^{(m)}(\xi_j)}{\sum_{\xi:\xi_i=0} p(\mathbf{z}|\xi)\prod_{j\neq i}p(x_j|\xi_j)\,T_j^{(m)}(\xi_j)} \quad (5)$$

and at convergence the pseudo-posterior ratios which result from either decoder become

$$\frac{\Pr(\xi_i=1|\mathbf{x},\mathbf{y},\mathbf{z})}{\Pr(\xi_i=0|\mathbf{x},\mathbf{y},\mathbf{z})} \leftarrow \frac{T_i(1)}{T_i(0)}\,\frac{U_i(1)}{U_i(0)}\,\frac{p(x_i|\xi_i=1)}{p(x_i|\xi_i=0)}$$

3. STATIONARY POINTS

Existence of stationary points is proved in [8]; here we identify the form some may take. In view of the symmetry of the relations (4) and (5), we focus our attention initially on (4), since the results will then apply to (5) upon permuting variables.

Let $\xi = (\xi_1, \ldots, \xi_k)$ be a candidate configuration of the binary information bits, and let ξ_i^0 and ξ_i^1 result by setting the i-th bit to 0 or 1. For example, with $\xi = (1,0,1)$, we have

$$\xi_1^0 = (0,0,1) \quad \text{and} \quad \xi_1^1 = (1,0,1),$$

and so on. We say that $U(\xi) = U_1(\xi_1)\cdots U_k(\xi_k)$ coincides with the configuration ξ when $U_i(1) = \xi_i$ for all i, assuming $U_i(0) + U_i(1) = 1$.

Lemma 1 *If $U(\xi)$ coincides with a binary configuration $\bar{\xi}$, then*

$$\frac{T_i(1)}{T_i(0)} = \frac{p(\mathbf{y}|\bar{\xi}_i^1)}{p(\mathbf{y}|\bar{\xi}_i^0)}$$

For the proof, the term $\prod_{j\neq i}p(x_j|\xi_j)U_j(\xi_j)$ from (4) has all $U_j(\xi_j)$ as either 0 or 1. Only that product for which each $U_j(\xi_j)$ yields 1 survives in the sum of the numerator or denominator, corresponding to $\bar{\xi}_i^1$ in the numerator, or $\bar{\xi}_i^0$ in the denominator; the surviving product then cancels in the ratio. ◇

Consider now a constituent trellis encoder for which the final state is pushed to a predetermined configuration

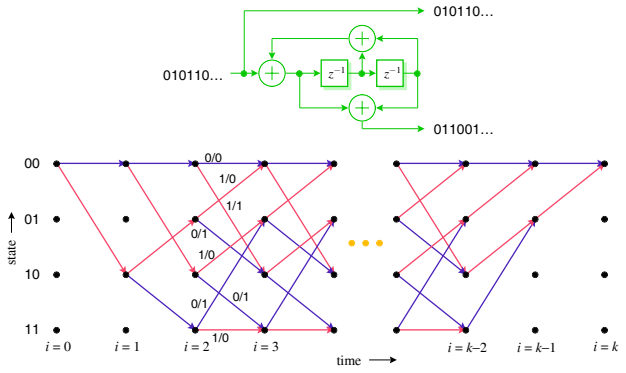

Fig. 1. Illustrating a "(5,7)" convolutional coder in recursive systematic form, as well as its constrained trellis.

(typically the zero state). As an example, Figure 1 shows a "(5, 7)" recursive systematic convolutional encoder of rate $1/2$, along with its trellis diagram. No matter which state is reached at time $k-2$, one can always choose the final two input samples ξ_{k-1} and ξ_k to push the final state to zero at time k. The reduces the number of candidate input configurations $\underline{\xi}$ to 2^{k-2} in this case. This constraint enters naturally into the decoding algorithm of [9] (since the final state is known with certainty), effectively annihilating those likelihood evalutions $p(\mathbf{y}|\underline{\xi})$ for which $\underline{\xi}$ is not a valid input bit combination of the constrained trellis.

Theorem 1 *Suppose that the minimum Hamming distance separating valid input configurations is two or greater. If $U(\underline{\xi})$ matches a valid input configuration $\underline{\xi}$, and $p(\mathbf{x}|\underline{\xi}) \times p(\mathbf{y}|\underline{\xi}) \neq 0$, then $T(\underline{\xi})$ matches this same input configuration.*

For the proof, we note that $\bar{\underline{\xi}}_i^0$ and $\bar{\underline{\xi}}_i^1$ differ in one bit position (Hamming distance 1), and therefore one of these must be exluded from the set of valid input configurations. As such,

$$p(\mathbf{y}|\bar{\underline{\xi}}_i^1) = \begin{cases} p(\mathbf{y}|\bar{\underline{\xi}}), & \text{if } \bar{\underline{\xi}} = \bar{\underline{\xi}}_i^1; \\ 0, & \text{otherwise;} \end{cases}$$

and similarly for $p(\mathbf{y}|\bar{\underline{\xi}}_i^0)$. With $p(\mathbf{y}|\bar{\underline{\xi}})p(\mathbf{x}|\bar{\underline{\xi}}) \neq 0$, each ratio $T_i(1)/T_i(0)$ from Lemma 1 is either $0/1$ or $1/0$; by scaling $T_i(0)+T_i(1)=1$, we obtain $T_i(1) = \xi_i \in \{0,1\}$. $\diamond$

Applying the same reasoning to the other decoder, we see that stationary points include valid input configurations.

4. CONTRACTIVITY

Suppose $U_*(\xi)$ and $T_*(\xi)$ form a stationary point of the turbo-decoding algorithm. Let $\mathcal{U}_\varepsilon$ denote the set of factorable distributions $U(\xi) = U_1(\xi_1)\cdots U_k(\xi_k)$ in an ε-neighborhood of $U_*(\xi)$:

$$\mathcal{U}_\varepsilon = \{U(\xi) : \underbrace{\max_i |U_{*,i}(1) - U_i(1)|}_{\|U_*(\xi) - U(\xi)\|} \leq \varepsilon\}.$$

Any such $U(\xi)$ fed to the first decoder will, by continuity, yield a $T(\xi)$ which lies "near" $T_*(\xi)$, in the sense that a constant β_1 exists for which

$$\|T_*(\xi) - T(\xi)\| \leq \beta_1 \|U_*(\xi) - U(\xi)\|, \quad \text{for all } U(\xi) \in \mathcal{U}_\varepsilon.$$

Let now $\mathcal{T}_\varepsilon$ be the image of $\mathcal{U}_\varepsilon$ obtained from the first coder. Lipschitz continuity [10] of the second decoder implies the existence of a constant β_2 for which

$$\|U_*(\xi) - U(\xi)\| \leq \beta_2 \|T_*(\xi) - T(\xi)\|, \quad \text{for all } T(\xi) \in \mathcal{T}_\varepsilon.$$

If $\beta_1\beta_2 < 1$, the algorithm is locally convergent since, for any $U^{(m)}(\xi) \in \mathcal{U}_\varepsilon$,

$$\|U_* - U^{(m+1)}\| \leq \beta_2 \|T_* - T^{(m)}\| \leq \beta_2\beta_1 \|U_* - U^{(m)}\|$$

If $\beta_2\beta_1 < 1$, then $U^{(m+1)}(\xi)$ remains in $\mathcal{U}_\varepsilon$; by induction

$$\|U_* - U^{(m+l)}\| \leq (\beta_2\beta_1)^l \|U_* - U^{(m)}\|$$

which tends exponentially fast to zero as l grows.

Explicit expressions for the Lipschitz constants are difficult to obtain, although Monte-Carlo simulations can yield estimates for different values of the noise variance and block length. In particular, let U_* be a stationary point at a valid input configuration; it generates T_* to be passed to the other decoder. We can then generate (randomly or deterministically) different choices for U in the vicinity of U_*, and measure the T which results in a vicinity of T_*. The ratio $\hat{\beta}_1 = \|U_* - U\|/\|T_* - T\|$ is then an underestimate of the Lipschitz constant; by generating a sufficiently dense set of U, and retaining the maximum of $\hat{\beta}_1$ over this set, we can obtain a numerical estimate of β_1.

Figure 2 shows the estimated Lipschitz constant for the (5,7) trellis decoder (cf. Figure 1) versus the channel noise variance for different block lengths, and a particular ε. The horizontal axis uses the raw signal-to-noise ratio over the channel instead of E_b/N_0, since this latter depends on the code rate and would therefore be different between one constituent code and the concatenated code for the same noise variance. The figure confirms that the Lipschitz constant becomes more favorable as the noise variance decreases and/or the block length increases.

Using a parallel concatenated code scheme, the values β_1 and β_2 are the same. For a given block length local convergence to the correct code word should occur, based on this analysis, when the Lipschitz constant is less than one. For a short block length (64 symbols for the systematic bits) the raw signal-to-noise ratio of the channel

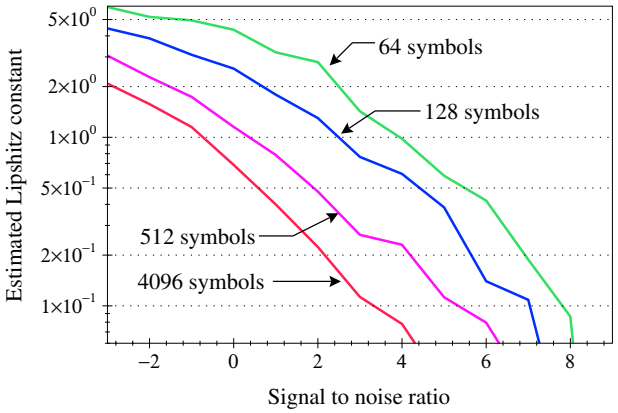

Fig. 2. Estimated Lipschitz constant of the (5,7) decoder versus the raw signal-to-noise ratio, for different block lengths, using $\varepsilon = 0.2$.

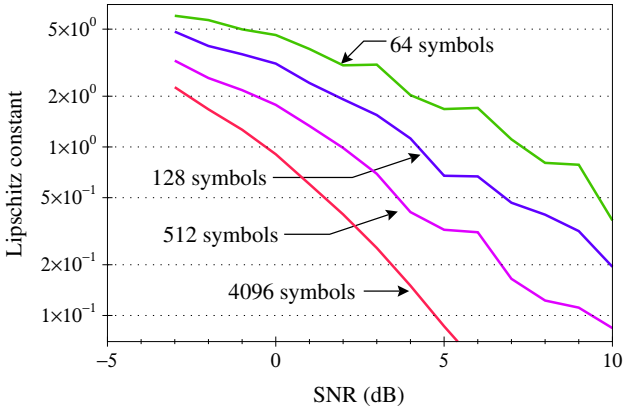

Fig. 3. Estimated Lipschitz constant of the (5,7) decoder versus the raw signal-to-noise ratio, for different block lengths, using $\varepsilon = 0.3$.

should be about 5 dB, whereas for a longer block length, the raw signal-to-noise ratio can drop to about 0 dB.

By increasing ε, the Lipschitz constant also increases, as illustrated in Figure 3, giving less favorable contraction. Since the contractivity increases near a convergent point, the local convergence rate is likely superlinear. In exchange, until a given iteration falls within a basin of attraction, convergence may be rather slow.

The drop in contractivity for shorter block comes perhaps as no surprise, since Shannon's arguments [11], [12], [13] show that, under maximum likelihood decoding, the probability of error can be bounded by an exponentially decreasing function of the block length. But the turbo decoding algorithm does not, in general, implement maximum likelihood decoding, which necessitates unearthing the more direct mechanism identified here.

5. CONCLUDING REMARKS

Our contribution is to show that stationary points exist at valid input configurations, which should assist the analytic study of Lipschitz constants mapping pseudo-priors to extrinsic information values. We observe that the Lipschitz constants worsen with shorter block lengths, but further study is required.

6. REFERENCES

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

[2] C. Heegard and S. B. Wicker, *Turbo Coding*. Kluwer, Boston, MA, 1999.

[3] B. Vucetic and J. Yuan, *Turbo Codes: Principles and Applications*. Kluwer, Boston, MA, 2000.

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

[5] D. Divsalar, S. Dolinar, and F. Pollara, "Iterative turbo decoder analysis based on density evolution," *IEEE J. Selected Areas in Communications*, vol. 19, pp. 891–907, May 2001.

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

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

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

[9] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. 20, pp. 284–287, 1974.

[10] T. L. Saaty and J. Bram, *Nonlinear Mathematics*. New York: McGraw-Hill, 1964.

[11] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[12] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.

[13] S. G. Wilson, *Digital Modulation and Coding*, Prentice-Hall, Upper Saddle River, NJ, 1996.