A PRACTICAL ALGORITHM FOR TURBO-DECODING ENHANCEMENT

Dejan Spasov[†], *Gian Mario Maggio*[◇‡] *and Ljupco Kocarev*[†]

[†]Institute for Nonlinear Science, University of California, San Diego, La Jolla (CA)
[◇]STMicroelectronics, Inc., Advanced System Technology, San Diego (CA)
[‡]Center for Wireless Communications, University of California, San Diego, La Jolla (CA)

ABSTRACT

It has been demonstrated that the BER (bit error rate) performance of turbo codes may be enhanced by applying an adaptive control technique to suppress transient dynamics in the turbo-decoding algorithm. In this work, we extend the method to a suboptimal decoding scheme, which is more suitable for practical implementation, and discuss the effect of metrics quantization.

1. INTRODUCTION

In 1948, Shannon formulated the famous noisy channel coding theorem [1], which establishes the fundamental limits for digital communication in terms of channel capacity. Then, in 1993, Berrou *et al.* introduced *turbo codes*, which have been shown to perform at rates extremely close to the Shannon limit [2]. This powerful class of error-correcting codes is based upon a constrained random code ensemble, described by some fixed parameters plus randomness, decoded using iterative decoding algorithms.

Recently, in a pioneering paper [3], Richardson has presented a geometrical interpretation of the turbo decoding algorithm and formalized it as a nonlinear dynamical system. The turbo decoding algorithm appears as an iterative algorithm aimed at solving a system of $2n$ equations in $2n$ unknowns, where n is the block-length size. In a follow-up of this work [4], Agrawal and Vardy parameterized the turbo-decoding algorithm, as a dynamical system, in terms of the SNR (signal-to-noise ratio). Moreover, in [5] it has been shown that the turbo-decoding algorithm may exhibit a whole range of phenomena known to occur in nonlinear systems [6]. These include multiple fixed points, oscillatory behavior, and even chaos. An adaptive control technique for suppressing transient chaos, thereby reducing the number of iterations needed by the turbo-decoding algorithm to converge has been proposed in [7, 8].

The heart of the turbo-decoding scheme consists of iterating the *maximum a posteriori* (MAP) algorithm [9], applied to each constituent code. The MAP algorithm is typically implemented by a SISO (soft-input soft-output) module that computes the *a posteriori probability* (APP) of the information symbols or, more generally, a reliability value for each information symbol. A natural reliability value, in the binary case, is the logarithm likelihood ratio (LLR). The LLR may be computed exactly using the BCJR [10] algorithm. The BCJR algorithm is optimal to generate the sequence of APPs. However, its computationally complexity does not make it a suitable candidate for practical communication systems. In order to reduce the decoding complexity of the MAP algorithm, researchers have developed other versions of the MAP algorithm requiring less computation, without major drawback in its performance. Among those algorithms are Log-MAP and Max-Log-MAP [11]. In this work, we extend our previous results concerning the adaptive feedback control for the suppression of transient dynamics. Namely, rather than considering the BCJR algorithm we work with the suboptimal Max-Log-MAP decoder. Furthermore, we discuss the effect of the LLR quantization on the system performance.

E-mails: dspasov@ucsd.edu, maggio@ieee.org, lkocarev@ucsd.edu.

2. TURBO CODES AS NONLINEAR SYSTEMS

The turbo code consists of a parallel concatenation of two recursive systematic binary convolutional codes, CC1 and CC2, separated by a random interleaver [2]. Let $\boldsymbol{i}$ be the information bit sequence of length n at the input to the turbo encoder, and let $\boldsymbol{p}_1(\boldsymbol{i})$ (resp. $\boldsymbol{p}_2(\boldsymbol{i})$) be the parity bits produced by the first (resp. second) encoder. The input sequence of the second encoder has the same Hamming weight as i, but the bits are rearranged due to a permutation defined by the choice of the random interleaver, π. The information bit sequence, $\boldsymbol{i}$, along with the parity bit sequences $\boldsymbol{p}_1(\boldsymbol{i})$ and $\boldsymbol{p}_2(\boldsymbol{i})$, forms a turbo codeword $(\boldsymbol{i}, \boldsymbol{p}_1(\boldsymbol{i}), \boldsymbol{p}_2(\boldsymbol{i}))$ of length $3n$.

We assume that the turbo code is transmitted over an AWGN (additive white Gaussian noise) binary-input memoryless channel, using the BPSK (binary phase shift keying) modulation. We denote the transmitted turbo codeword by $(\mathbf{s}^0, \mathbf{s}^1, \mathbf{s}^2) \equiv (\boldsymbol{i}, \boldsymbol{p}_1(\boldsymbol{i}), \boldsymbol{p}_2(\boldsymbol{i}))$.

Let $\boldsymbol{c}^0$, $\boldsymbol{c}^1$ and $\boldsymbol{c}^2$ be the channel outputs corresponding to the input sequences $\mathbf{s}^0$, $\mathbf{s}^1$ and $\mathbf{s}^2$, respectively. The turbo decoder consists of two components: a decoder DEC1 for the convolutional code CC1, and a decoder DEC2 for the code CC2. These decoders use iteratively the MAP algorithm [9] in order to compute the extrinsic information, X_i, of the information bits. The input for the decoder DEC1 (resp. DEC2) is the extrinsic information X_2 (resp. X_1) from the decoder DEC2 (resp. DEC1), which is used as a priori information after interleaving (resp. deinterleaving).

The turbo decoding algorithm may be described by a discrete-time dynamical system of the form:

$$\begin{aligned} X_1(l+1) &= F_1[X_2(l); \boldsymbol{c}^0, \boldsymbol{c}^1] \qquad (1) \\ X_2(l) &= F_2[X_1(l); \boldsymbol{c}^0, \boldsymbol{c}^2] \qquad (2) \end{aligned}$$

where $F_i = (F_{11}, \ldots, F_{1n})$ are nonlinear functions which depend on the constituent codes. Equations (1)-(2) define an n-dimensional mapping, because there are only n independent variables, namely X_1 (or alternatively X_2). The quantities $\boldsymbol{c}^0$, $\boldsymbol{c}^1$ and $\boldsymbol{c}^2$ are completely characterized by the channel likelihood ratios. Consequently, the mapping (1)-(2) depends on $3n$ parameters.

After l iterations, a hard decision on the j-th bit can be made according to the sign of the LLR:

$$L_j(l) = \log \frac{p_j^1(l)}{p_j^0(l)} = X_{1j}(l) + X_{2j}(l) + \frac{4}{\sigma^2} c_j^0 \qquad (3)$$

where $p_j^0(l)$ and $p_j^1(l)$ are the probabilities that the j-th bit is either 0 or 1, and σ^2 represents the noise variance.

3. LOG-MAP AND MAX-LOG-MAP ALGORITHMS

The computational complexity of the BCJR iterative algorithm may be reduced by transferring the recursions into the log domain and invoking an approximation to dramatically reduce the complexity. Namely, the approximation (of the $\max$ operation) is the following:

$$\begin{aligned} \max{}^*(x,y) &= \ln(e^x + e^y) \\ &= \max(x,y) + \ln(1 + e^{-|y-x|}) \\ &= \max(x,y) + f_c(|y-x|) \end{aligned}$$

This corresponds to the *Log-MAP* algorithm. Such algorithm, though most complex, offers the best BER performance. When the correction term $f_c(|y-x|)$ is ignored in the computation, the algorithm is called the *Max-Log-MAP* algorithm. The Max-Log-MAP algorithm is suboptimal but also the least complex MAP algorithm [11].

In a real field, a conventional turbo decoder uses Log-MAP or Max-Log-MAP algorithm for constituent decoding. In the sequel, we briefly describe both algorithms.

3.1. Log-MAP Algorithm

Consider a trellis structure of a rate $1/n_r$ constituent decoder for turbo decoding. At any decoding step, k, the branch metric $\gamma_k(s', s)$ between state s and s' is defined in log domain as follows [11]:

$$\log \gamma_k(s', s) = u_k(L_c y_{k,0} + L_a(u_k)) + \sum_{i=1}^{n_r - 1} L_c y_{k,i} x_{k,i}$$

where u_k is the k-th information symbol, $x_{k,i}$ is the i-th parity symbol corresponding to u_k, and $y_{k,i}$ is the i-th received symbol corresponding to u_k and $x_{k,i}$. L_c is a channel reliability obtained from SNR estimation. $L_a(u_k)$ is an a priori information for u_k. In log domain, the state metrics $\alpha_k(s)$ and $\beta_{k-1}(s')$ are defined by the following recursions:

$$\log \alpha_k(s) = \log\left(\sum_{s'} e^{\log \alpha_{k-1}(s') + \log \gamma_k(s', s)}\right) \qquad (4)$$

$$\log \beta_{k-1}(s') = \log\left(\sum_{s} e^{\log \beta_k(s) + \log \gamma_k(s', s)}\right) \qquad (5)$$

In the above, normalization is not considered. With these metric values, the APP value of the Log-MAP algorithm can be determined as follows:

$$L(\hat{u}_k) = \log\left(\frac{\sum_j e^{M_0(j)}}{\sum_j e^{M_1(j)}}\right) \qquad (6)$$

M_{u_k} is the j-th metric sum of state metrics and branch metric in case of information symbol u_k, where the metric sums have been rearranged by their descending order in magnitude. Thus, $M_{u_k}(0)$ means the best metric sum for u_k [11].

3.2. Max-Log-MAP Algorithm

Max-Log-MAP algorithm is a simplified version of Log-MAP algorithm by the following approximations of the state metrics and APP values:

$$\begin{aligned} \log \alpha_k(s) &\simeq \max_{s'} (\log \alpha_{k-1}(s') + \log \gamma_k(s', s)) \\ \log \beta_{k-1}(s') &\simeq \max_{s} (\log \beta_k(s) + \log \gamma_k(s', s)) \\ L(\hat{u}_k) &\simeq M_0(0) - M_1(0) \qquad (7) \end{aligned}$$

In (7), the APP value is represented by the difference between the best metric sums for information symbol 0 and 1 at any decoding step k, $M_0(0)$ and $M_1(0)$, respectively. The Max-Log-MAP algorithm does not require SNR estimates. However, it exhibits about 0.3-0.4 dB performance degradation relative to the Log-MAP algorithm, with perfect SNR estimation [11].

4. ADAPTIVE CONTROL ALGORITHM

In [7, 8] we have proposed an adaptive control technique for suppressing transient chaos, thereby reducing the number of iterations needed by the turbo-decoding algorithm to converge. In fact, in the so-called *waterfall region* the turbo decoding-algorithm converges either to the chaotic invariant set or to the "unequivocal" fixed point, after a long transient behavior [5]. In some cases, the algorithm spends a few thousand iterations before reaching the fixed point solution. As SNR increases, the average chaotic transient lifetime [6] decreases and, consequently, the number of iterations necessary to converge decreases. A schematic block diagram of the turbo decoder including the adaptive control is shown in Fig. 1, where in [7, 8] the SISO module was implemented by the BCJR algorithm. Basically, the control technique adaptively attenuates extrinsic LLRs, thus improving the rate of convergence and the BER performance, for turbo codes. Namely, the control algorithm reads:

$$g(X_i) = \Gamma e^{-\lambda|X_i|} X_i, \tag{8}$$

where $\Gamma \in (0, 1]$ and $\lambda \geq 0$ are control parameters.

5. SIMULATION RESULTS

In this work we consider a rate-1/2 turbo code based upon identical constituent recursive convolutional codes, with generators $G = (37, 21)$, and utilizing the Max-Log-MAP algorithm as SISO modules. The codewords are transmitted over an AWGN channel using BPSK modulation. The length of the interleaver is $n = 1024$.

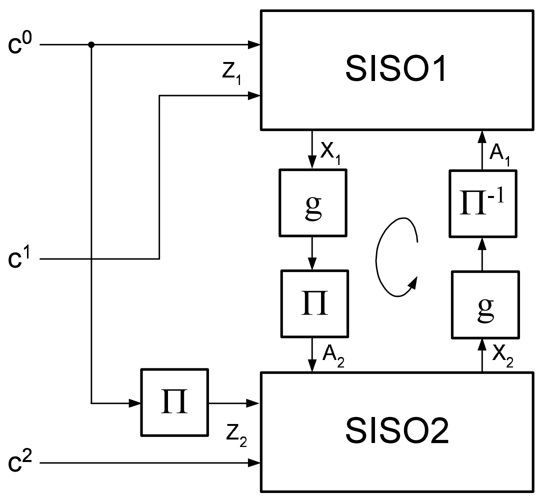

Fig. 1. Turbo decoder with adaptive control: X_i are extrinsic information; c^j is the channel output corresponding to the input sequence s^j; $g(\cdot)$ is the control function (8).

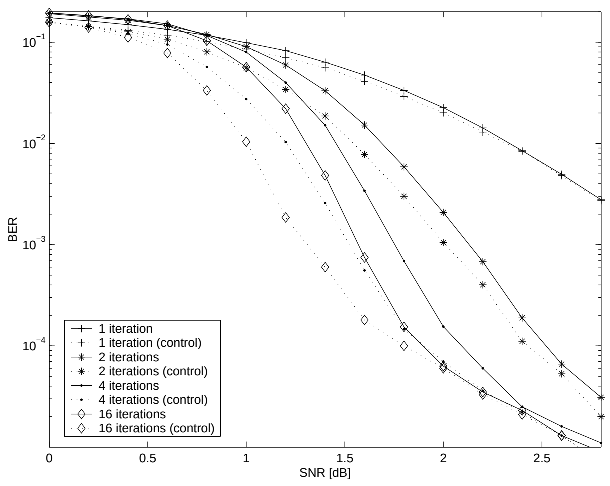

Fig. 2. BER performance of the turbo code utilizing the Max-Log-MAP algorithm with/without adaptive control ($\Gamma = 0.75, \lambda = 0$), as a function of the number of iterations. The coding gain increases with the number of iterations.

The simulation results corresponding to the application of the adaptive control method ($\Gamma = 0.75, \lambda = 0$), in conjunction to the Max-Log-MAP algorithm are shown in Fig. 2, as a function of the number of iterations. From Fig. 2, one can appreciate the coding gain due to the adaptive control, as the number of iterations progresses. After $l = 16$ iterations, the turbo-decoding algorithm with control exhibits an average gain of ~ 0.25 dB versus the case without control. These results correspond to the waterfall region of the turbo code. As in [8], we found that the control is not effective in the error floor region, where no transient chaos occurs. Also, we stress that the control does not work well when the interleaver size is large ($\gg 1000$)

5.1. Dependence on the Control Parameters

Figure 3 illustrates the effect of the control parameters Γ and λ on the BER performance of the turbo code, for $l = 16$ iterations. Namely, four representative cases are analyzed:

i) $\Gamma = 0.75, \lambda = 0$
ii) $\Gamma = 0.75, \lambda = 0.01$
iii) $\Gamma = 0.75, \lambda = 0$
iv) $\Gamma = 0.9, \lambda = 0.01$

It follows that, unlike what observed for the BCJR algorithm [8], the best performance is achieved basically when $\Gamma = 0.75$ and $\lambda = 0$. In this case, setting $\lambda > 0$ offers very little gain, as visible in Fig. 3. Also, with the control parameters as in i), the error floor is not affected. On the other hand, decreasing the value of Γ, results in the error floor rising.

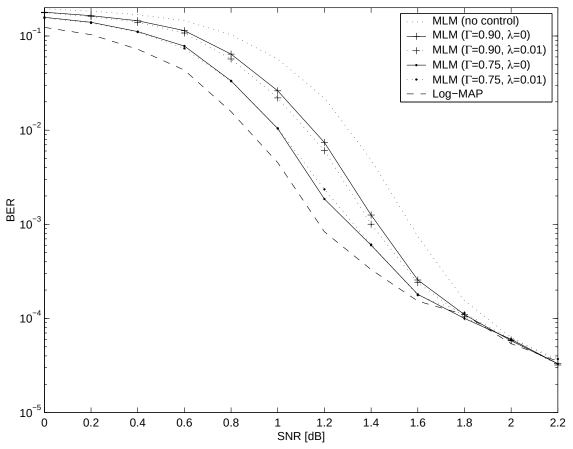

Fig. 3. Dependence of the BER performance on the control parameters Γ and λ, for $l = 16$ iterations. Note that the best performance is obtained essentially for $\Gamma = 0.75$, while the exponential factor λ has very little influence on the BER.

5.2. Effect of Quantization

We now discuss the effect of the metrics quantization on the BER performance. The corresponding results for $l = 4$ iterations are reported in Fig. 4. We compare the performances, with adaptive control, when all variables used by the Max-Log-MAP algorithm are quantized. In particular, we consider an 8-bytes floating-point representation versus 1-byte fixed point, with 4 bits devoted to the decimal part. From Fig. 4 we conclude that the quantization error does not affect significantly the BER performance of the turbo-decoding algorithm with adaptive control.

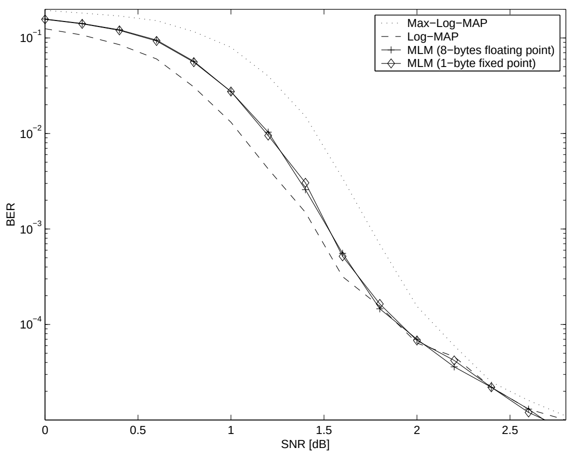

Fig. 4. Effect of the variables quantization on the BER performance, with adaptive control ($\Gamma = 0.75, \lambda = 0$), for $l = 4$ iterations.

6. CONCLUSIONS

In this work, we have extended an adaptive control method for turbo codes to the case of suboptimal decoding, employing the Max-Log-MAP algorithm. The latter is more practical than the BCJR algorithm. The simulation results show that the method exhibits a considerable coding gain in the waterfall region of the turbo code. Finally, the effect of the metrics quantization is discussed.

7. REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 379–423, pp. 623-56, 1948.

[2] C. Berrou, A Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-Codes," *Proc. IEEE International Communications Conference*, pp. 1064–70, 1993.

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

[4] D. Agrawal and A. Vardy, "The turbo decoding algorithm and its phase trajectories," *IEEE Trans. Information Theory*, vol. 47, No. 2, pp. 699–722, 2001.

[5] Z. Tasev, L. Kocarev and G.M. Maggio, "Bifurcations and chaos in the turbo decoding algorithm," *Proc. ISCAS 2003*, Bangkok, Thailand, 2003.

[6] E. Ott, *Chaos in dynamical systems*, Cambridge University Press, New York, 1993.

[7] L. Kocarev, Z. Tasev and A. Vardy, "Improving turbo codes by control of transient chaos in turbo-decoding algorithms," *Electronics Letters*, vol. 38, no. 20, pp. 1184-6, 2002.

[8] L. Kocarev, Z. Tasev and G. M. Maggio, "Applications of nonlinear dynamics to the turbo decoding algorithm," *Proc. ISCAS 2003*, Bangkok, Thailand, 2003.

[9] R.W. Chang and J.C. Hancock, "On receiver structures for channels having memory," *IEEE Trans. IT*, vol. 12, pp. 463-468, 1966.

[10] L.R. Bahl, J. Cocke, F. Jelinek, J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. IT-20, no. 2, pp. 284-287, 1974.

[11] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," *Proc. ICC95*, pp. 1009-13, 1995.