BLIND TURBO EQUALIZATION USING THE CONSTANT MODULUS ALGORITHM

Phillip A. Regalia

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

Abstract: A turbo equalizer is modified to allow its operation in a blind manner, i.e., without resorting to training sequences or to channel identification steps. It exploits a recent variant of the constant modulus algorithm, in collaboration with differential encoding, for which the decoder is linked in an iterative scheme with a conventional error correction coder. A characterization of stationary points is obtained, and conditions for proximity to a maximum likelihood decoding rule are identified.

1. INTRODUCTION

Iterative techniques in reception have met with intense interest since the advent of the turbo-decoding algorithm for parallel concatenated codes. More recently, turbo equalization, in which the channel and source coder are interpreted as a serial concatenated coder, have successfully integrated linear equalization and maximum likelihood decoding into iterative algorithms which drop intersymbol interference and channel noise below previously attainable levels.

The technique was first proposed in (Douillard *et al.*, 1995), and is directly related to the iterative decoding of serially concatenated codes (Benedetto and Montorsi, 1996). The "inner" decoder was implemented using a soft-output Viterbi algorithm, for which the computational complexity grows exponentially with the channel length, restricting its practical application to short channels. A modification proposed in (Laot *et al.*, 2001) and subsequently refined in (Tüchler *et al.*, 2002), consists in replacing the Viterbi channel decoder with a linear decision feedback equalizer, whose decision feedback path is driven by the outer decoder's output. This drops the complexity to a function which is linear (or sometimes quadratic) in the channel length, while offering bit-error performance close to the original design from (Douillard *et al.*, 1995).

If the channel is properly identified, then the design equations for the decision feedback equalizer which maximize the signal-to-noise ratio subject to perfect interference cancellation may be obtained with explicit formulas (Laot *et al.*, 2001). As expected, performance may degrade significantly when the channel is not properly identified, requiring more complicated configurations using training sequences in combination with equalizer adaptation strategies [e.g., (Tüchler *et al.*, 2002)]. The use of training sequences, of course, consumes available bandwidth, while blind channel identification schemes can often be ill-conditioned (Delmas *et al.*, 2000), or simply inapplicable when sufficient diversity is lacking.

The technique proposed here uses a completely blind solution, employing the finite-interval variant of the constant modulus algorithm developed in (Regalia, 2002). Since the constant modulus criterion is phase blind, its successful application requires differential encoding/decoding for correct symbol recovery. The technique developed here treats the differential encoder as the inner coder of a serial concatenated code, with the outer coder fournished by a conventional trellis code for error correction

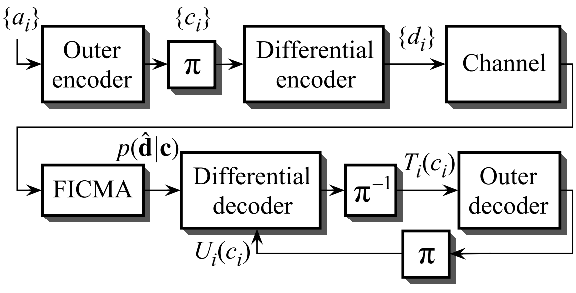

Fig. 1. Using the finite-interval constant modulus algorithm and iterative decoding with the inclusion of the differential coder/decoder pair.

purposes. The decoder stage applied to the output of the constant modulus algorithm implements iterative decoding applied to the serially concatenated pair. Simulations indicate, similar to other iterative decoding techniques, a threshold effect: provided the constant modulus algorithm approximately equalizes the channel, the information exchange between the two decoders iteratively reduces the remaining errors. Although the bit error rate versus signal to noise ratio is perhaps not as impressive as earlier designs exploiting channel knowledge, the proposed scheme works effectively without any channel knowledge.

The next section describes the algorithm, its stationary points, and relations to maximum likelihood decoding.

2. ALGORITHM DESCRIPTION

Figure 1 shows an overview of the proposed scheme, where $\{a_i\}_{i=1}^K$ are the information bits, which are fed to an error correction coder (typically a trellis coder) to create the coded bits $\{c_j\}_{j=1}^N$ (each being either 0 or 1). We assume that this first coder is systematic, so that $c_j = a_j$ for $j = 1, \ldots, K$. The bits $\{c_j\}$ are coded in turn using a differential encoder, to produce the output bits d_j:

$$d_j = d_{j-1} \oplus c_{\pi(j)}, \quad j = 1, 2, \ldots, N,$$

in which $\{\pi(1), \pi(2), \ldots, \pi(N)\}$ is a permutation of $\{1, 2, \ldots, N\}$, and "$\oplus$" denotes modulo-2 addition; the seed value $d_0 \in \{0, 1\}$ is chosen arbitrarily. The symbols are converted to antipodal signaling $\{-1, +1\}$ and transmitted over a channel with nontrivial delay spread:

$$x_j = (2d_j - 1)$$
$$u_j = \sum_k h_k x_{j-k} + b_j$$

where $\{h_k\}$ denotes the channel impulse response, of unknown length, and b_j is additive noise, assumed white and Gaussian.

The constant modulus algorithm attempts to restore the channel input sequence by adjusting the coefficients of an equalizer; a variant suitable to block processing is given in (Regalia, 2002), and yields an output block $\{\hat{d}_j\}$ which satisfies

$$\hat{d}_j \approx \pm(2d_j - 1)$$

in which the "$\pm$" sign reflects the phase ambiguity which is inherent to the constant modulus algorithm (hence, the need for differential encoding), and the approximation sign reflects how a linear equalizer can at best reduce noise and intersymbol interference, but never eliminate them.

The differential decoder calculates the a posteriori probability ratios

$$\frac{\Pr(c_i = 1|\hat{d}_1, \ldots, \hat{d}_N)}{\Pr(c_i = 0|\hat{d}_1, \ldots, \hat{d}_N)} = \frac{\sum_{\mathbf{c}:c_i=1} \Pr(\mathbf{c}|\hat{\mathbf{d}})}{\sum_{\mathbf{c}:c_i=0} \Pr(\mathbf{c}|\hat{\mathbf{d}})} \quad i = 1, \ldots, N,$$
$$= \frac{\sum_{\mathbf{c}:c_i=1} p(\hat{\mathbf{d}}|\mathbf{c}) \Pr(\mathbf{c})}{\sum_{\mathbf{c}:c_i=0} p(\hat{\mathbf{d}}|\mathbf{c}) \Pr(\mathbf{c})} \qquad (1)$$

in which the a priori probability function $\Pr(\mathbf{c})$ is assumed to factor into the product of its marginals:

$$\Pr(\mathbf{c}) = \Pr(c_1) \Pr(c_2) \cdots \Pr(c_N).$$

This assumption is, strictly speaking, invalid, since the bits $\{c_j\}$ are interdependent, but the differential decoder cannot exploit these interdependencies without a significant increase in complexity, and so uses the marginals $\Pr(c_i)$ instead.

By virtue of the factorization of $\Pr(\mathbf{c})$, each term of the numerator (resp., denominator) of (1) contains a factor $\Pr(c_i = 1)$ [resp., $\Pr(c_i = 0)$], such that the probability ratio becomes

$$\frac{\Pr(c_i = 1|\hat{\mathbf{d}})}{\Pr(c_i = 0|\hat{\mathbf{d}})} = \underbrace{\frac{\Pr(c_i = 1)}{\Pr(c_i = 0)}}_{\frac{U_i(1)}{U_i(0)}} \underbrace{\frac{\sum_{\mathbf{c}:c_i=1} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_{j \neq i} \Pr(c_j)}{\sum_{\mathbf{c}:c_i=0} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_{j \neq i} \Pr(c_j)}}_{\triangleq \frac{T_i(1)}{T_i(0)}} \qquad (2)$$

The a priori ratio $U_i(1)/U_i(0)$ is initially set to 1, but will be "baised" in subsequent iterations by the extrinsic information to be fed back from the outer decoder below.

The likelihood function $p(\hat{\mathbf{d}}|\mathbf{c})$ is evaluated under the assumption that the constant modulus equalizer has restored the channel input to within additive Gaussian noise (which is, strictly speaking, incorrect here), giving

$$p(\hat{\mathbf{d}}|\mathbf{c}) \sim \exp\left(-\sum_{j=1}^{N} \frac{[\hat{d}_j - (2d_j(\mathbf{c}) - 1)]^2}{2\sigma^2}\right)$$

in which $d_j(\mathbf{c})$ denotes the j^{th} bit of the differential encoder output when the input block is $\mathbf{c}$, and σ^2 is the noise variance. The forward-backward algorithm (Bahl *et al.*, 1974) may be used for the calculations summarized in (2) for $i = 1, 2, \ldots, N$.

The error correction decoder would nominally aim to calculate the a posteriori probability ratios

$$\frac{\Pr(a_i = 1|\hat{c}_1,\ldots,\hat{c}_N)}{\Pr(a_i = 0|\hat{c}_1,\ldots,\hat{c}_N)} = \frac{\sum_{\mathbf{a}:a_i=1} \Pr(\mathbf{a}|\hat{\mathbf{c}})}{\sum_{\mathbf{a}:a_i=0} \Pr(\mathbf{a}|\hat{\mathbf{c}})} = \frac{\sum_{\mathbf{a}:a_i=1} p(\hat{\mathbf{c}}|\mathbf{a}) \Pr(\mathbf{a})}{\sum_{\mathbf{a}:a_i=0} p(\hat{\mathbf{c}}|\mathbf{a}) \Pr(\mathbf{a})} \quad i = 1,\ldots,K,$$

in which $\Pr(\mathbf{a})$ is taken as a uniform distribution over the 2^K possibilities for $\mathbf{a}$, and so my be omitted. As written, this presupposes a noisy version of the outer encoder's output as $\hat{\mathbf{c}} = (2\mathbf{c} - \mathbf{1}) + \text{noise}$, but this quantity is not immediately available. If it were, then each evaluation of the likelihood function $p(\hat{\mathbf{c}}|\mathbf{a})$ would appear as

$$p(\hat{\mathbf{c}}|\mathbf{a}) \sim \exp\left(-\sum_{j=1}^{N} \frac{[\hat{c}_j - (2c_j(\mathbf{a}) - 1)]^2}{2\sigma^2} \right)$$

in which $c_j(\mathbf{a})$ is either 0 or 1. Therefore, to each hypothetical bit $\hat{c}_j$ we associate two evaluations: $\exp[-(\hat{c}_j \mp 1)/(2\sigma^2)]$, (corresponding to $c_j(\mathbf{a}) = 0$ or 1), which are replaced by the two evaluations of the function T_j calculated in (2) above:

$$\frac{\exp(-(\hat{c}_j - 1)^2/(2\sigma^2))}{\exp(-(\hat{c}_j + 1)^2/(2\sigma^2))} \leftarrow \frac{T_j(1)}{T_j(0)}, \quad j = 1,\ldots,N.$$

Here "$\leftarrow$" denotes the "usurpation" operator. The forward-backward algorithm may then proceed directly, following this systematic substitution.

To develop an external description of the resulting decoding operation, we note that this substitution is tantamount to replacing the likelihood evaluation by

$$p(\hat{\mathbf{c}}|\mathbf{a}) \leftarrow \prod_{i=1}^{N} T_i(c_i(\mathbf{a})) \tag{3}$$

in which the right-hand side emphasizes that only those bit combinations $(c_1,\ldots,c_N)$ which lie in the outer coderbook make sense. To this end, let $\phi(\mathbf{c})$ denote the indicator function for the outer code:

$$\phi(\mathbf{c}) = \begin{cases} 1, & \text{if } \mathbf{c} \text{ is an outer codeword;} \\ 0, & \text{if not.} \end{cases}$$

The 2^N configurations of $(c_1, \ldots, c_N)$ generate 2^N evaluations of $\prod_{i=1}^N T_i(c_i)$, but only 2^K of these survive in the product $\phi(\mathbf{c})\prod_i T_i(c_i)$. We may then establish a one-to-one correspondence between the 2^K "surviving" evaluations in $\phi(\mathbf{c})\prod_i T_i(c_i)$ and the 2^K "usurped" evaluations of $\mathbf{p}(\hat{\mathbf{c}}|\mathbf{a})$ in which the hypothesis $\mathbf{a}$ varies among 2^K possibilities generated from K bits $(a_1, \ldots, a_K)$, as in (3). The outer decoder then admits an external description as

$$\frac{\sum_{\mathbf{a}:a_i=1} p(\hat{\mathbf{c}}|\mathbf{a})}{\sum_{\mathbf{a}:a_i=0} p(\hat{\mathbf{c}}|\mathbf{a})} \leftarrow \frac{\sum_{\mathbf{c}:c_i=1} \phi(\mathbf{c}) \prod_j T_j(c_j)}{\sum_{\mathbf{c}:c_i=0} \phi(\mathbf{c}) \prod_j T_j(c_j)} = \frac{T_i(1)}{T_i(0)} \underbrace{\frac{\sum_{\mathbf{c}:c_i=1} \phi(\mathbf{c}) \prod_{j\neq i} T_j(c_j)}{\sum_{\mathbf{c}:c_i=0} \phi(\mathbf{c}) \prod_{j\neq i} T_j(c_j)}}_{\rightarrow \frac{U_i(1)}{U_i(0)}} \tag{4}$$

in which we note that:

- Since the outer coder is systematic, the first k bits $c_1, \ldots, c_K$ coincide with the information bits $a_1, \ldots a_K$. In addition, the formula above may be evaluated as written for the parity-check bits $c_{K+1}, \ldots, c_N$.
- Each term in the numerator (resp., denominator) contains a factor $T_i(1)$ [resp., $T_i(0)$], so that the ratio $T_i(1)/T_i(0)$ naturally factors out. The remaining term (the "extrinsic" information) will usurp the (pseudo-) *a priori* probabilities of the inner decoder for the next iteration:
$$\frac{\Pr(c_i = 1)}{\Pr(c_i = 0)} \leftarrow \frac{U_i(1)}{U_i(0)}$$

If we let a superscript (n) denote an iteration index, then the coupling of the two decoders admits an external description of the form

$$\frac{\sum_{\mathbf{c}:c_i=1} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_j U_i^{(n)}(c_i)}{\sum_{\mathbf{c}:c_i=0} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_j U_i^{(n)}(c_i)} = \frac{U_i^{(n)}(1)}{U_i^{(n)}(0)} \frac{T_i^{(n)}(1)}{T_i^{(n)}(0)} \tag{5}$$

$$\frac{\sum_{\mathbf{c}:c_i=1} \phi(\mathbf{c}) \prod_j T_i^{(n)}(c_i)}{\sum_{\mathbf{c}:c_i=0} \phi(\mathbf{c}) \prod_j T_i^{(n)}(c_i)} = \frac{T_i^{(n)}(1)}{T_i^{(n)}(0)} \frac{U_i^{(n+1)}(1)}{U_i^{(n+1)}(0)} \tag{6}$$

in which these "pseudo"-posterior probabilities are calculated for $i = 1, \ldots, N$, at each iteration. A stationary point corresponds to $U_i^{(n+1)}(c_i) = U_i^{(n)}(c_i)$ which, by inspection, gives

Property 1. A stationary point occurs if and only if the two decoders reach consensus on the pseudo-posterior probabilities [(5), (6)] for $i = 1, 2, \ldots, N$.

It may not be clear whether the substitution illustrated in (4) above gives an optimal coupling between the two decoders in any sense. To gain greater insight into this question, suppose the likelihood function $p(\hat{\mathbf{d}}|\mathbf{c})$ is scaled so that its outcomes sum to one:

$$\sum_{\mathbf{c}} p(\hat{\mathbf{d}}|\mathbf{c}) = 1.$$

Introduce the N marginals of the likelihood function $p(\hat{\mathbf{d}}|\mathbf{c})$ as

$$p_i(\hat{\mathbf{d}}|c_i = 1) = \sum_{\mathbf{c}:c_i=1} p(\hat{\mathbf{d}}|\mathbf{c}), \quad i = 1,2,\ldots,N,$$

and similarly for $p_i(\hat{\mathbf{d}}|c_i = 0)$. We may then show:

Property 2. If the likelihood function $p(\hat{\mathbf{d}}|\mathbf{c})$ factors into the product of its marginals, i.e.,

$$p(\hat{\mathbf{d}}|\mathbf{c}) = p_1(\hat{\mathbf{d}}|c_1)\, p_2(\hat{\mathbf{d}}|c_2)\cdots p_N(\hat{\mathbf{d}}|c_N),$$

then:

(a) The algorithm in (5) and (6) converges in a single iteration;
(b) The resulting pseudo-posteriors coincide with the symbol-by-symbol maximum likelihood estimates for the concatenated code.

To verify, note that if $p(\hat{\mathbf{d}}|\mathbf{c})$ is the product of its marginals, then so is

$$\alpha\, p(\hat{\mathbf{d}}|\mathbf{c}) \prod_{j=1}^{N} U_j^{(n)}(c_j) = \prod_{j=1}^{N} \alpha_j\, p(\hat{\mathbf{d}}|c_j) U_j^{(n)}(c_j)$$

in which the scalars $\{\alpha_j\}$ ensure that the sum over all outcomes equals one. Since the left-hand side of (5) calculates the i-th marginal ratio, we may observe that

$$\frac{\displaystyle\sum_{\mathbf{c}:c_i=1} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_j U_j^{(n)}(c_i)}{\displaystyle\sum_{\mathbf{c}:c_i=0} p(\hat{\mathbf{d}}|\mathbf{c}) \prod_j U_j^{(n)}(c_i)} = \frac{U_i^{(n)}(1)}{U_i^{(n)}(0)}\, \frac{p(\hat{\mathbf{d}}|c_i = 1)}{p(\hat{\mathbf{d}}|c_i = 0)}$$

whenever $p(\hat{\mathbf{d}}|\mathbf{c})$ factors as the product of its marginals. Upon comparing with (5) we identify

$$\frac{T_i^{(n)}(1)}{T_i^{(n)}(0)} = \frac{p(\hat{\mathbf{d}}|c_i = 1)}{p(\hat{\mathbf{d}}|c_i = 0)}$$

for all iterations n, implying immediately that a stationary point has been attained. We then have

$$\prod_{j=1}^{N} T_j^{(n)}(c_j) = \prod_{j=1}^{N} p(\hat{\mathbf{d}}|c_j) = p(\hat{\mathbf{d}}|\mathbf{c}),$$

and the calculation performed by the outer decoder in (6) becomes, for $i = 1, 2, \ldots, K$,

$$\frac{\displaystyle\sum_{\mathbf{c}:c_i=1} \phi(\mathbf{c}) \prod_j T_j^{(n)}(c_i)}{\displaystyle\sum_{\mathbf{c}:c_i=0} \phi(\mathbf{c}) \prod_j T_j^{(n)}(c_i)} = \frac{\displaystyle\sum_{\mathbf{c}:c_i=1} \phi(\mathbf{c})\, p(\hat{\mathbf{d}}|\mathbf{c})}{\displaystyle\sum_{\mathbf{c}:c_i=0} \phi(\mathbf{c})\, p(\hat{\mathbf{d}}|\mathbf{c})} = \frac{\displaystyle\sum_{\mathbf{a}:a_i=1} p(\hat{\mathbf{d}}|\mathbf{a})}{\displaystyle\sum_{\mathbf{a}:a_i=0} p(\hat{\mathbf{d}}|\mathbf{a})}$$

in which we note that each nonzero evaluation of $\phi(\mathbf{c})\, p(\hat{\mathbf{d}}|\mathbf{c})$ may be identified with an evaluation of $p(\hat{\mathbf{d}}|\mathbf{c}(\mathbf{a})) = p(\hat{\mathbf{d}}|\mathbf{a})$, since the indicator function $\phi(\mathbf{c})$ annihilates those evaluations of $p(\hat{\mathbf{d}}|\mathbf{c})$ for which $\mathbf{c}$ is not in the outer codebook. As the outer code is systematic, we have $c_i = a_i$ for $i = 1, \ldots, K$, allowing a direct substitution of the variables of summation. The final ratio which results is recognized as that obtained from a bit-by-bit maximum likelihood metric for the concatenated code. ◇

In practice, the likelihood function $p(\hat{\mathbf{d}}|\mathbf{c})$ need not factor as the product of its marginals, but if it is "close" to such a factorable function, one would expect the algorithm to converge rapidly. This proximity to a factorable likelihood function will in fact hold in extreme conditions:

- *High signal-to noise ratio and good channel diversity*. Let $\mathbf{c}_*$ denote the true input to the differential encoder. If the FICMA algorithm restores a faithful rendition of the encoded sequence $\mathbf{d}(\mathbf{c}_*)$, then $\hat{\mathbf{d}} \approx (2\mathbf{d}(\mathbf{c}_*) - \mathbf{1})$ and

$$\frac{p(\hat{\mathbf{d}}|\mathbf{c} \neq \mathbf{c}_*)}{p(\hat{\mathbf{d}}|\mathbf{c} = \mathbf{c}_*)} = \frac{\exp(-\|\hat{\mathbf{d}} - (2\mathbf{d}(\mathbf{c}) - \mathbf{1})\|^2/2\sigma^2)}{\exp(-\|\hat{\mathbf{d}} - (2\mathbf{d}(\mathbf{c}_*) - \mathbf{1})\|^2/2\sigma^2)} \approx \exp\left(-\frac{m^2}{2\sigma^2}\right)$$

  where $m^2 = \|\hat{\mathbf{d}} - (2\mathbf{d}(\mathbf{c}) - \mathbf{1})\|^2$. As the noise variance σ^2 decreases, this ratio tends to zero, so that

$$p(\hat{\mathbf{d}}|\mathbf{c}) \overset{\sigma^2 \to 0}{\longrightarrow} \delta_{\mathbf{c}_*}(\mathbf{c}) = \begin{cases} 1, & \mathbf{c} = \mathbf{c}_*; \\ 0, & \mathbf{c} \neq \mathbf{c}_*. \end{cases}$$

  We note that the Kronecker delta function $\delta_{\mathbf{c}_*}(\mathbf{c})$ can always be written as the product of its marginals (which themselves are Kronecker delta functions of the individual bits). In these favorable conditions, the algorithm converges rapidly, yielding the correct decoding with high probability.
- *Poor signal to noise ratio*. If the noise variance σ^2 is large, the likelihood evaluations $p(\hat{\mathbf{d}}|\mathbf{c})$ become comparable in value, and approach a uniform distribution:

$$p(\hat{\mathbf{d}}|\mathbf{c}) \overset{\sigma^2 \to \infty}{\longrightarrow} u(\mathbf{c}) = \frac{1}{2^N} \quad \text{for all } \mathbf{c}$$

  We note that a uniform distribution likewise factors as the product of its marginals (which themselves are uniform distributions). For pessimistic signal-to-noise ratios, then, the algorithm converges rapidly to a solution whose probability of error approaches $\frac{1}{2}$.

For intermediate signal to noise ratios, the convergence properties of iterative decoding are less well understood, owing to the presence of loops in the equivalent belief propagation graph [akin to the situation for parallel concated codes (McEliece *et al.*, 1998) and low-density parity check codes (Fossorier *et al.*, 1999)]. Nonetheless, the existence of stationary points can at least be established, similar to (Richardson, 2000) for parallel concatenated codes.

To this end, we recall that the Brouwer fixed point theorem (Saaty and Bram, 1964) asserts that any continuous map from a closed, bounded and convex set into itself admits a fixed point. Consider the pseudo-prior probabilities $U_i^{(n)}(c_i = 1)$, which lie between zero and one:

$$0 \leq U_i^{(n)}(c_i = 1) \leq 1, \quad i = 1, 2, \ldots, N.$$

This gives a closed, bounded and convex subset of $\mathbb{R}^N$. Since the pseudo-priors $U_i^{(n+1)}(c_i = 1)$ at the next iteration remain within this subset, and since the application which maps $\{U_i^{(n)}(c_i)\}$ to $\{U_i^{(n+1)}(c_i)\}$ is continuous, the Brouwer theorem is satisfied, to show that fixed points of the iteration always exist.

Note that the external description in terms of pseudo-posteriors in (5) and (6) above applies generically to iterative decoding of serially concatenated codes, so that Properties 1 and 2 apply to more general schemes described in (Benedetto and Montorsi, 1996), as does the fixed-point result above.

3. EXAMPLE

We present some results and observations using a rate $1/2$ $(5,7)$ coder converted to systematic form for the outer coder, a block length of 512 bits for **a**, an all-pole channel of the form $h_k = (0.6)^k$. For a given packet, the limiting pseudo-posteriors $U_i(1)T_i(1)$ from (5) [or (6), cf. Property 1] were observed to display one of two configurations:

- *Case 1*: The pseudo-posteriors tend to a nearly binary $\{0,1\}$ distribution, as illustrated in Fig. 2.
- *Case 2*: The pseudo-posteriors remain comfortably away from a binary distribution, as illustrated in Fig. 3.

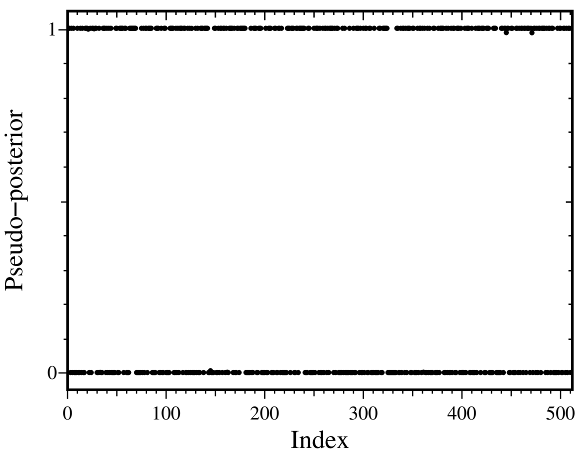

Fig. 2. "Case 1" configuration of the pseudo-posteriors. The decoder decisions are observed, with high reliability, to be correct.

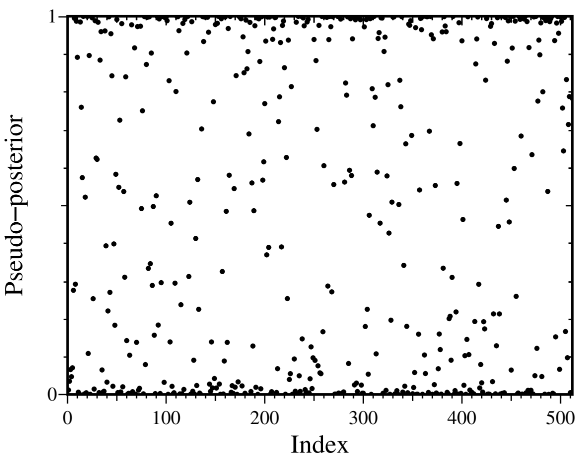

Fig. 3. "Case 2" configuration of the pseudo-posteriors: For such cases, the bit error rate is consistently observed to exceed 40%.

Interestingly, intermediate distributions between figures 2 and 3 were never observed at convergence. For case 2, the bit error rate was consistently found to be in excess of 40%, often approaching a worst-case 50% bit error rate that a "coin-flip" decision device would achieve. For case 1, on the other hand, the decoded symbols are usually correct. Figure 4 shows the percentage of packets for which the algorithm converges to a "Case 1" configuration, versus the (signal plus intersymbol interference)-to-noise ratio [(S+ISI)/N] of the received signal, as determined empirically using 5000 packets. For [(S+ISI)/N] greater than 6 dB, no bit errors were detected for "Case 1" pseudo-posteriors; at 5 dB the bit error rate degrades to about 10^{-3}.

If we average the bit error rate over both "Case 1" and "Case 2" configurations, we obtain values between 10^{-2} and 10^{-1} over the 10 dB to 5 dB range for [(S+ISI)/N], which is hardly impressive. Note that, since the differential encoder has rate 1, the distance spectrum of the concatenated code is basically the same as the distance spectrum for the outer code by itself. Experiments obtained by removing the differential encoder/decoder pair yielded again bit error rates between 10^{-2} and 10^{-1}, provided the phase ambiguity inherent to the constant modulus equalizer is resolved. But the percentage of packets which give the "Case 1" behavior above (i.e., nearly binary pseudo-posteriors) drops by an order of magnitude.

This indicates a potential advantage of the iterative decoding scheme: inspection of the pseudo-posterior distribution gives an indication of the reliability of the result. This can be useful in bidirectional applications: by setting $p_i = \alpha_i T_i(1) U_i(1)$ [where the normalizing constant α_i fulfills $1 - p_i = \alpha_i T_i(0) U_i(0)$], one can examine the average entropy over the packet as

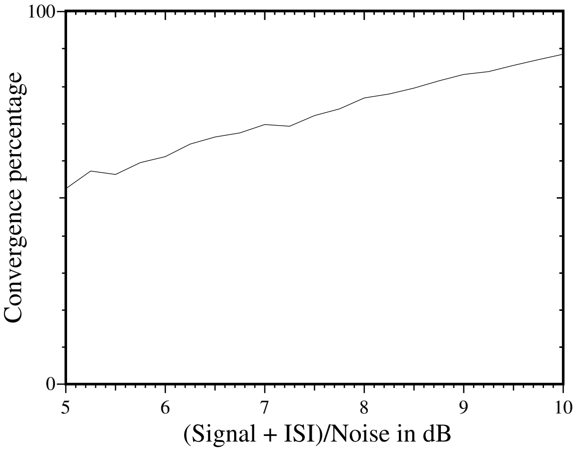

Fig. 4. Percentage of packets which converge to a "Case 1" configuration, versus the (signal + interference)-to-noise ratio of the received signal.

$$H(p) = -\frac{1}{N}\sum_{i=1}^{N} p_i \log p_i$$

When $H(p)$ is less than a given threshold, a "Case 1" configuration is identified and the decoded packet is declared reliable; otherwise a "Case 2" configuration is declared and the receiver requests a packet retransmission. Without the iterative decoding, a significant grey zone persists between Case 1 and Case 2 configurations, and the reliability of a decoded packet is less certain.

4. CONCLUDING REMARKS

A modified turbo equalization scheme has been proposed which uses neither a Viterbi decoder nor a conventional decision feedback equalizer, since the implementation of either in practice requires accurate knowledge of the channel impulse response and/or the use of training sequences. The present scheme, by contrast, works in a completely blind manner, by exploiting a finite-interval constant modulus algorithm (Regalia, 2002). Since the constant modulus criterion is phase blind, differential encoding is included. By interpreting the differential code as a unit rate trellis code, the decoding algorithm can be iteratively linked with the trellis decoder of the outer (or error correction) code, by straightforward adaption of iterative decoding of serial codes (Douillard *et al.*, 1995), (Benedetto and Montorsi, 1996).

The inclusion of a rate-one differential encoder does not improve the distance properties of the code, and therefore should not improve the bit error rate if maximum likelihood decoding were available. But the iterative turbo-decoding algorithm applied here is not, in general, a maximum likelihood decoder, unless the likelihood factorization of Property 2 happens to apply. The converged probabilities furnished by the iterative decoder, however, are observed to approach a binary distribution whenever the decoding appears correct, indicating high reliability, unlike the maximum likelihood decoding rule which need not approach a binary distribution for "correctly decoded" results. This property, however, would appear dependent on the constituent codes used in the concatenation, and is therefore a subject for further study.

REFERENCES

Bahl, L. R., J. Cocke, F. Jelinek, and J. Raviv (1974). Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Information Theory* **20**(3), 284–287.

Benedetto, S. and G. Montorsi (1996). Iterative decoding of serially concatenated convolutional codes. *Electronics Letters* **32**, 1186–1188.

Delmas, J.-P. H. Gazzah, A. P. Liavas and P. A. Regalia (2000). Statistical analysis of some second-order methods for blind channel identification/equalization with respect to channel undermodeling. *IEEE Trans. Signal Processing* **48**(7), 1984–1998.

Douillard, C., M. Jezequel, C. Berrou, A. Picart, P. Didier and A. Glavieux (1995). Iterative correction of intersymbol interference: Turbo-equalization. *ETT* **6**(5), 507–512.

Fossorier, M. P. C., M. Mihaljević and H. Imai (1999). Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Trans. Communications* **47**(5), 673–680.

Laot, C., A. Glavieux and J. Labat (2001). Turbo equalization: Adaptive equalization and channel decoding jointly optimized. *IEEE J. Selected Areas in Communications* **19**(9), 1744–1752.

McEliece, R. J., D. J. C. MacKay and J.-F. Cheng (1998). Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *IEEE Trans. Selected Areas in Communications* **16**(2), 140–152.

Regalia, P. A. (2002). A finite interval constant modulus algorithm. In: *Proc. ICASSP-02*. Vol. III. pp. 2285–2288. Orlando, FL.

Richardson, T. (2000). The geometry of turbo decoding dynamics. *IEEE Trans. Information Theory* **46**(1), 9–23.

Saaty, T. L. and J. Bram (1964). *Nonlinear Mathematics*. McGraw-Hill. New York.

Tüchler, M., R. Kotter, and A. Singer (2002). Turbo equalization: Principles and new results. *IEEE Trans. Communications* **50**(5), 754–767.