Blind Iterative Decoding of Side-informed Data Hiding Using the Expectation-Maximization Algorithm

Félix Balado[a], Fernando Pérez-González[b] and Pedro Comesaña[b]

[a]University College Dublin, Belfield, Dublin 4, Ireland;
[b]University of Vigo, Lagoas-Marcosende s/n, 36200 Vigo, Spain

ABSTRACT

Distortion-Compensated Dither Modulation (DC-DM), also known as Scalar Costa Scheme (SCS), has been theoretically shown to be near-capacity achieving thanks to its use of side information at the encoder. In practice, channel coding is needed in conjunction with this quantization-based scheme in order to approach the achievable rate limit. The most powerful coding methods use iterative decoding (turbo codes, LDPC), but they require knowledge of the channel model. Previous works on the subject have assumed the latter to be known by the decoder. We investigate here the possibility of undertaking blind iterative decoding of DC-DM, using maximum likelihood estimation of the channel model within the decoding procedure. The unknown attack is assumed to be i.i.d. and additive. Before each iterative decoding step, a new optimal estimation of the attack model is made using the reliability information provided by the previous step. This new model is used for the next iterative decoding stage, and the procedure is repeated until convergence. We show that the iterative Expectation-Maximization algorithm is suitable for solving the problem posed by model estimation, as it can be conveniently intertwined with iterative decoding.

Keywords: Side-informed data hiding, Distortion-Compensated Dither Modulation (DC-DM), Scalar Costa Scheme (SCS), blind iterative decoding, Expectation-Maximization algorithm

1. INTRODUCTION

The use of side information at the encoder has proven crucial to the data hiding problem. The solution provided by Costa[1] for a communications setting resembling the data hiding scenario has been decisive to show that host-signal-induced self-distortion can be effectively removed through a clever design of the transmission codebook. In fact, using this very codebook design, Costa showed that exactly the same capacity holds for a scheme with side information only at the encoder and another one having the same side information available at the decoder, i.e., without self-distortion. This derivation assumed Gaussian-distributed side information and an additive white Gaussian noise channel independent from the former, but extensions of the same basic result have been made since. In the context of data hiding this result was first pointed out by Chen and Wornell, who showed[2] that their Distortion-Compensated Quantization Index Modulation (DC-QIM) theoretical scheme closely resembled Costa's encoding and decoding procedure, and hence that it was optimal in the same sense. In addition, they studied the achievable rate of a particular implementable case of DC-QIM with uniform scalar quantizers. This scheme was called Distortion-Compensated Dither Modulation (DC-DM), which was shown to be, asymptotically, only 1.53 dB away from Costa's capacity.

Afterwards, Eggers et al. followed the inverse path to show with their Scalar Costa Scheme (SCS)[3] that a practical implementation of Costa's random codebook was possible. Actually, SCS happens to be equivalent to DC-DM, and for this reason we will use the latter term to refer to this particular scheme. Nevertheless, the timely and thorough analysis made for SCS complemented and extended the previous one. Apart from many

Further author information: (Send correspondence to Félix Balado.)
Félix Balado: E-mail: fiz@ihl.ucd.ie, Telephone: +353 (0)1 716 2454

other practical issues, it considered non-asymptotic assumptions —e.g., the use of binary constellations— and channel coding.

As already pointed out in the aforementioned works and elsewhere, channel coding is the way to approach channel capacity in any communications scenario, and, therefore, also in data hiding using side information at the encoder. A number of prior works have studied the use of state-of-the-art channel coding for side-informed data hiding.[4, 5, 6] All these proposals use turbo codes[7] —which have been shown to be able to asymptotically achieve error-free decoding for a signal-to-noise ratio near the Shannon limit— over *scalar* side informed schemes following Costa's guidelines. The schemes used therein involve scalar uniform quantizers which are resized using a scaling factor before quantization —i.e., amounting to distortion compensation—, and hence they are equivalent to DC-DM. To be precise, the methods used by Kesal et al. and Chou et al. are equivalent to DC-DM for high document-to-watermark ratios (see Section 1.1), but this is the usual case in data hiding.

In addition to turbo codes, other less powerful iteratively decodable techniques are tested in,[4] while concatenation with an outer convolutional code is explored in[6] to build a turbo-trellis method. Disregarding the channel coding method used, these approaches bear in common the necessity of knowing the type of channel and the level of distortion for undertaking decoding. This information is required to obtain the soft reliability values of the symbol decisions in the decoding procedure. Incidentally, all these works have worked under the hypothesis that this information was known to the decoder. Moreover, and except for the works of Eggers et al., the statistical model used by the decoder in most of this prior art is not the exact one, but an approximation that avoids having to deal with the nonlinear modular nature of quantization lattices. In this paper we explore how to perform blind* iterative decoding of DC-DM, i.e., without prior knowledge of the attack channel model and level of distortion, and taking into account simultaneously the modularity of the method at the receiver.

1.1. Framework

Firstly, we will establish the framework used in this paper. We assume that N samples $\mathbf{x} = (x[1], \dots, x[N])$ are pseudorandomly chosen from a host signal; the samples in $\mathbf{x}$ are zero-mean Gaussian with covariance matrix $\Gamma_x = \sigma_x^2 \cdot I$. The corresponding watermarked signal $\mathbf{y}$, resulting from embedding a given binary information vector $\mathbf{b}$, undergoes a zero-mean random additive attack channel, so that the signal received at the decoder is $\mathbf{z} = \mathbf{y} + \mathbf{n}$. The samples of the random variable $\mathbf{n}$ are assumed to be independent identically distributed (i.i.d.) and independent of $\mathbf{x}$, but having unknown probability density function (pdf) and variance σ_n^2. In practice, the independence between the samples of $\mathbf{n}$ is approximately granted by the pseudorandom choice of $\mathbf{x}$. Through the paper we will find useful to define the watermark-to-noise ratio (WNR) and the document-to-watermark ratio (DWR), as the ratios in decibels between the watermark and attack power, and the host and watermark power, respectively.

Next, we will briefly recall the basic formulation of DC-DM and some of its properties. In binary DC-DM one information symbol $b[k] \in \{\pm 1\}$ is hidden by quantizing a sample of the host signal $x[k]$ to the nearest centroid $Q_{b[k]}(x[k])$ belonging to the uniform lattice† $\Lambda_{b[k]}$ given by

$$\Lambda_{b[k]} = 2\Delta\,\mathbb{Z} + \Delta\,\frac{(b[k]+1)}{2} + d[k], \qquad (1)$$

with $d[k]$ a key-dependent value that we will take as zero for simplifying the analysis without loss of generality. M-ary versions of the same scheme can be used, but the achievable rate has been shown to be essentially the same than the binary case for WNR's lower than approximately 4 dB.[3] The watermarked signal is obtained as

$$y[k] = x[k] + \nu \cdot e[k]\,, \qquad (2)$$

i.e, the watermark is the quantization error $e[k] \triangleq Q_{b[k]}(x[k]) - x[k]$ weighted by an optimizable constant ν, $0 \leq \nu \leq 1$. If $\Delta \ll \sigma_x$, what holds true for usual DWR's due to perceptual reasons, then $e[k]$ can be assumed to

*Not to be confused with blind data hiding, that refers to the unavailability of the host signal at the decoder.

†Extending the usual definition of lattice, which in principle must include the origin.

be independent of $x[k]$ and uniformly distributed, $e[k] \sim U(-\Delta, \Delta)$. Then, the watermark $w[k] = y[k] - x[k]$ is also uniform, and the embedding power is $E\{w^2[k]\} = \nu^2\Delta^2/3$.

The decoder acts by quantizing sample by sample the received signal $\mathbf{z}$ to the closest codebook lattice. Hence we have that

$$\hat{b}[k] = \arg \min_{b \in \{\pm 1\}} \big|Q_b(z[k]) - z[k]\big|. \tag{3}$$

In the preceding exposition we have assumed that the embedding distortion at each sample is identical, i.e., $\Delta[k] = \Delta$ for all $k = 1, \ldots, N$. We could have allowed instead the quantization step $\Delta[k]$ to vary at each sample proportionally to $\alpha[k]$, where $\boldsymbol{\alpha}$ responds to a set of local perceptual energy restrictions such that $E\{w^2[k]\} \leq c \cdot \alpha^2[k]$, for some constant c. Still, if the attack channel abides by the same perceptual constraints —what it is reasonable if the maximum imperceptible attack power is to be used—, we can renormalize the problem to the situation with constant Δ and σ_n assumed above.

Channel Coding. Following what was stated in the introduction, we will hide a binary codeword $\mathbf{c} = (c[1], \ldots, c[N])$ instead of N uncoded bits. The codeword is obtained by encoding a binary information vector $\mathbf{b} = (b[1], \ldots, b[M])$, $M < N$, using a rate $R = M/N$ code. For notational simplicity, and without loss of generality, we have assumed that the codeword length is equal to the length N of the host signal vector $\mathbf{x}$. For embedding and decoding we will consider that the codeword symbols are given in antipodal form, i.e., $c[k] \in \{\pm 1\}$. In this way, each coded symbol $c[k]$ is embedded on $x[k]$ to obtain $y[k]$ as done above using $b[k]$.

We will center our attention on parallel concatenated codes with iterative decoding, i.e., turbo codes. Although we will particularize our proposal to these codes due to practical purposes, it will become clear that the basic idea can be similarly applied to other iteratively decodable procedures. We recall that the parallel concatenated turbo codewords have the form

$$\mathbf{c} = (\mathbf{c}^s \mid \mathbf{c}^{p_1} \mid \mathbf{c}^{p_2}), \tag{4}$$

where the subvector $\mathbf{c}^s = \mathbf{b}$ is the systematic output, and the subvectors $\mathbf{c}^{p_1}$ and $\mathbf{c}^{p_2}$ are the parity outputs corresponding to the constituent recursive systematic convolutionals (RSC's). The output $\mathbf{c}^{p_1}$ is due to the input of $\mathbf{b}$ to the first RSC, and the output $\mathbf{c}^{p_2}$ is due to the input of a pseudorandom permutation of $\mathbf{b}$ to the second RSC.

Choice of ν. This choice is important because it is known[3] that there is a different optimum at each WNR for the achievable rate of DC-DM. In the framework that we have established above, WNR $= 10\log_{10}\nu^2\Delta^2/(3\sigma_n^2)$. As discussed elsewhere,[8] the WNR is not known beforehand by the encoder what becomes a practical problem for DC-DM optimization. Previous works[4, 5, 6] have assumed anticausal knowledge of this amount, and so they have used the optimal scaling of their lattices —i.e., the optimal distortion compensation factor— at each WNR.

Here we will set a fixed distortion compensation parameter ν regardless of the WNR, taking profit of the peculiarities of near-optimal codes. Turbo codes present a distinctive abrupt decrease —usually termed cliff or waterfall— of the bit error rate at the decoder as the WNR increases. If the turbo code is well designed, this waterfall occurs relatively close to minimum WNR necessary for asymptotically errorless decoding. Due to this effect we can approximately choose the optimal ν as the one that corresponds to the WNR at the achievable rate R imposed by the turbo code. As a real code cannot be perfect, the optimum will actually correspond to a slightly higher WNR. Notice however that this choice of ν requires knowledge of the channel model (i.e., whether this is Gaussian, uniform, etc) for computing the achievable rate vs. WNR plots.[3] In addition, this optimization strategy does not hold for WNR's more negative than the waterfall area, but this is unimportant due to the high probabilities of error associated to turbo decoding in this range.

2. EXACT ITERATIVE DECODING OF DC-DM

In this section we will explain the way to exactly establish the reliability of the channel decisions when the channel model is known by the decoder to be Gaussian with variance σ_n^2. This computation has a twofold purpose: 1) making explicit the modular nature of the DC-DM decoding procedure; 2) obtaining the exact reliability values to be used for later comparisons of exact iterative decoding against blind iterative decoding, when this particular attack channel is used.

The decoder receives the noisy signal $\mathbf{z} = \mathbf{y}+\mathbf{n}$ and proceeds to perform MAP iterative decoding. This requires the probabilities $p(z[k] \mid c[k] = c)$ for $c \in \{\pm 1\}$, what amounts to a statistical description of $z[k]$ depending on each possible symbol decision. As the watermark $w[k]$ can be assumed to follow a uniform distribution (see Section 1.1) we have that $y[k]$ is also uniform, as

$$y[k] = Q_c(x[k]) - (1-\nu) \cdot e[k], \tag{5}$$

with c the embedded symbol value. Then, $z[k] = y[k] + n[k]$ is the sum of two independent random variables, the first of them uniform and the second one Gaussian. The pdf of $z[k]$ is consequently the convolution of the corresponding pdf's. We can write this pdf as $f(z[k]) * \delta\{z[k] - Q_c(x[k])\}$, with

$$f(z) \triangleq \frac{1}{2(1-\nu)\Delta} \left\{ Q\left(\frac{z-(1-\nu)\Delta}{\sigma_n}\right) - Q\left(\frac{z+(1-\nu)\Delta}{\sigma_n}\right) \right\}, \tag{6}$$

and $Q(z) \triangleq \int_z^{\infty} \exp(-x^2/2)/\sqrt{2\pi}\, \mathrm{d}x$. This pdf of $z[k]$ is conditioned to a concrete centroid assumption, but we need the pdf for a generic symbol decision. For obtaining this expression notice that, due to using (3) at the decoder, the decision $\hat{c}[k]$ can be seen as being based on the modular offsets

$$\begin{aligned} \tilde{z}_c[k] &\triangleq \{z[k] \mod \Lambda_c\} - \Delta \\ &= \left\{ z[k] + \Delta\frac{(c+1)}{2} \right\} \mod 2\Delta - \Delta \end{aligned} \tag{7}$$

to each one of the two lattices Λ_c, with $c \in \{-1, 1\}$. Using these offsets, the minimum distance decision can be rewritten as

$$\hat{c}[k] = \arg\min_c |\tilde{z}_c[k]|. \tag{8}$$

Considering (8), it is clear that the reliability measure for the decision $\hat{c}[k] = c$ is just

$$p(z[k] \mid c[k] = c) \triangleq \tilde{f}(\tilde{z}_c[k]), \tag{9}$$

with $\tilde{f}(\cdot)$ the pdf followed by $\tilde{z}_c[k]$. Notice that the operation (7) implies that this pdf is just the aliasing of the sections of (6) corresponding to the Voronoi regions of the lattice $2\Delta\mathbb{Z}$, that is

$$\tilde{f}(z) = \begin{cases} \sum_{w \in 2\Delta\mathbb{Z}} f(z-w), & |z| \leq \Delta \\ 0, & |z| > \Delta \end{cases}. \tag{10}$$

Using (10) the *a posteriori* log-likelihood ratio for a received value $z[k]$ is just

$$L(c[k]) = \log \frac{\tilde{f}\left(\tilde{z}_{+1}[k]\right)}{\tilde{f}\left(\tilde{z}_{-1}[k]\right)}. \tag{11}$$

This is the method used by Eggers et al. for computing the reliability of the symbol decisions. The approximations by Kesal et al. and Chou et al. amount to say that (10) is Gaussian with variance $\sigma_n^2 + (1-\nu)^2\Delta^2/3$, what does not render a true pdf due to the amplitude limitation of the decision variable. Nevertheless, experiments show that this approximation is sufficiently good when the turbo cliff happens at not too negative WNR values.

3. BLIND ITERATIVE DECODING OF DC-DM

In a general case the decoder does not know (10) because no knowledge is usually available about the type of attack pdf or its power (i.e., the actual value of WNR). First, it has to be remarked that none of these two questions pose difficulties to iterative decoding of spread-spectrum data hiding. The reason for this is that, as the DWR is usually high, the channel model is largely dominated by the host signal model, that can be assumed as known by the decoder. Nevertheless, the much lower achievable rate of spread-spectrum requires in turn much lower code rates for achieving the same performance at the same WNR values.

In order to envisage how to surmount these difficulties for DC-DM we may review first several related solutions. Most of them stem from the scenario of communications without side information using iterative decoding. For instance, some authors[9, 10] have proposed the estimation of the SNR (i.e., WNR) value at the decoder for channels known to be Gaussian. Alternatively, other approaches[11] involve choosing a pdf from a family of possible distributions, assuming knowledge of the SNR. But blind methods whose approach relies on estimating the actual pdf are more interesting for data hiding, as they jointly address both sketched problems. Among them we find that by Huang et al.,[12] who use a one-step histogram estimation, and the one by Li et al.,[13] who propose to heuristically refine a kernel-based model at each iterative decoding step, using the increasingly accurate decoded information.

Motivated by the latter approach, but, as we will see, using sounder theoretical grounds, we can take advantage from turbo-coded DC-DM to iteratively estimate the unknown attack pdf jointly with the decoding process. As the pdf (10) is not known beforehand at the decoder, we will assume at least a model with enough degrees of freedom. Taking profit that the support set of $\tilde{f}(z)$ is limited to $|z| < \Delta$, we can resort to approximating it using a simple but general model based on a finite number N_q of rectangular kernels. Then, we will assume that (10) may be approximated using

$$h(\boldsymbol{\theta}, z) \triangleq \sum_{i=1}^{N_q} \theta[i] \cdot \Pi\big(z - (i-1) \cdot \Delta_q + \Delta\big), \tag{12}$$

with the kernels $\Pi(z)$ defined as

$$\Pi(z) \triangleq \begin{cases} 1/\Delta_q, & 0 < z \leq \Delta_q \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

and $N_q \triangleq 2\Delta/\Delta_q$, which we assume integer. Of course, $h(\boldsymbol{\theta}, z) = 0$ for $|z| > \Delta$. Notice that a further advantage of (12) is that it makes no assumptions on the symmetry of the attack pdf. This model is usually considered to be non-parametric, although we can see it as a parametric one in which the parameters vector $\boldsymbol{\theta} = (\theta[1], \ldots, \theta[N_q])$ has to be adjusted.

Our initial objective is therefore to optimally estimate $\boldsymbol{\theta}$ from the received vector $\mathbf{z}$. The maximum likelihood approach for this estimation can be stated as

$$\hat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta}} P(\mathbf{z}, \boldsymbol{\theta}). \tag{14}$$

This estimation problem is inherently involved. Still, we may notice that the elements of $\mathbf{z}$ stem from the mixture of data drawn from two different distributions. At each $z[k]$ these two possible distributions (which are in fact the same one shifted by the offset Δ) correspond to each of the two possible embedded symbols $c[k] \in \{\pm 1\}$.

This is the situation for which the Expectation-Maximization (EM) algorithm[14] was conceived, aiming at finding the solution of (14) iteratively. The EM algorithm is a long-standing procedure with theoretically proven convergence properties[14] that, in order to iteratively solve (14), uses two alternating steps called E-step and M-step.

Unfortunately, we cannot afford the hypothesis of independence between the elements of $\mathbf{z}$ corresponding to the codeword parities. As we will see, this causes the problem (14) not to be explicitly solvable using EM. For

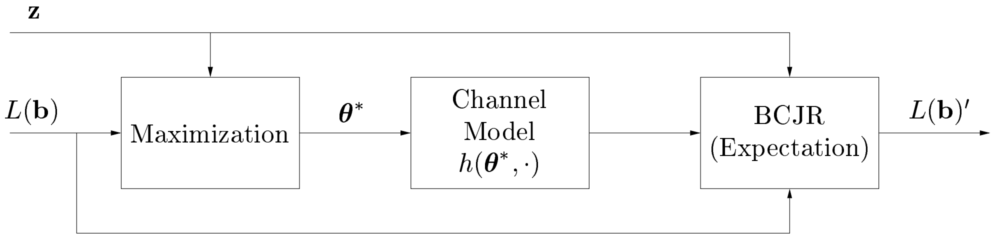

Figure 1. One step of the iterative EM algorithm intertwined with iterative turbo decoding. Necessary interleavings/deinterleavings of $\mathbf{z}^s$ and $L(\mathbf{b})$ for BCJR are not explicitly shown for simplicity.

this reason we will resort instead to solve

$$\hat{\boldsymbol{\theta}} = \max_{\boldsymbol{\theta}} P(\mathbf{z}^s, \boldsymbol{\theta}), \tag{15}$$

with $\mathbf{z}^s$ the subvector of $\mathbf{z}$ corresponding to the systematic part of the codeword $\mathbf{c}^s = \mathbf{b}$, following the notation in (4). Anyway, and as we will see next, the subvectors corresponding to the parities $\mathbf{z}^{p_1}$ and $\mathbf{z}^{p_2}$ can be used to improve the E-step beyond what we could get with $\mathbf{z}^s$ alone. In this way, we can intertwine the turbo decoding with the estimation problem. We describe next the steps of the Expectation-Maximization algorithm and their application to our problem, that is summarized in Figure 1.

3.1. Expectation Step

This step is equivalent to computing a probability mass function (pmf) of $\mathbf{c}^s = \mathbf{b}$ (hidden data) under the knowledge of $\mathbf{z}^s$ and $\boldsymbol{\theta}$, that is

$$q(\mathbf{b}) \triangleq P(\mathbf{b} \mid \mathbf{z}^s, \boldsymbol{\theta}). \tag{16}$$

If we disregard the subvectors of $\mathbf{z}$ corresponding to the codeword parities, our best estimate of (16) would be

$$q(\mathbf{b}) = \frac{P(\mathbf{z}^s, \mathbf{b}, \boldsymbol{\theta})}{\sum_{\mathbf{b}'} P(\mathbf{z}^s, \mathbf{b}', \boldsymbol{\theta})}. \tag{17}$$

Nevertheless, the subvectors $\mathbf{z}^{p_1}$ and $\mathbf{z}^{p_2}$ corresponding to the parity symbols allow us to compute the pmf (16) more reliably than (17). Actually, each iterative turbo decoding stage optimally updates the previous *extrinsic* pmf of $\mathbf{b}$ using the BCJR algorithm, which takes into account $\mathbf{z}$, the code used for the current parity, and the channel model $h(\boldsymbol{\theta}, \cdot)$.

Therefore, the probabilities $q(b[k])$, for $k = 1, \ldots, M$, given by the BCJR algorithm, are the best way to compute the distribution we need. Assuming that the information bits $b[k]$ are independent, we can write

$$q(\mathbf{b}) = \prod_{k=1}^{M} q(b[k]). \tag{18}$$

Remember that we can straightforwardly compute these probabilities from the log-likelihood ratios $L(b[k]) = \log\{q(b[k] = +1)/q(b[k] = -1)\}$.

We have to remark that this kind of approach involving iterative decoding and EM has already been used in communications for purposes such as channel state estimation[15, 16] —differently to this case, using pilot information—, or synchronization.[17]

3.2. Maximization Step

Now, using the pdf (18) and $\mathbf{z}^s$ we need to compute the new $\boldsymbol{\theta}$ that maximizes the EM functional,[18] that can be written as

$$\max_{\boldsymbol{\theta}} E_{q(\mathbf{b})}\{\log P(\mathbf{z}^s, \mathbf{b}, \boldsymbol{\theta})\}. \tag{19}$$

It is shown in Appendix A that the solution $\boldsymbol{\theta}^*$ to this optimization problem is given by the expression

$$\theta^*[i] = \frac{\sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k] = b)}{M}, \tag{20}$$

for $i = 1, \ldots, N_q$, and with $\mathcal{P}_b^i$ defined in (27).

After the M-step we go back to the E-step, for which a new iteration of turbo decoding is performed using the pdf update given by (20) (see Figure 1). This procedure is continued until convergence.

Simplifications. In order to gain further insight from (20) we can consider to use, instead of the soft values $q(b[k])$, the decisions $\hat{b}[k] = \text{sign } L(b[k])$ in that equation. With this choice the pmf's become as a matter of fact deterministic, as if $q(b[k] = +1) = 1$ then $q(b[k] = -1) = 0$, and vice versa.

Interestingly, in this suboptimal case (20) becomes the normalized histogram of $\mathbf{z}^s$ on the bins B_i, using the hard decisions $\hat{b}[k]$ to make the bin assignment of the corresponding $z^s[k]$. This decision-based approximation, that would be the intuitive way to update $\boldsymbol{\theta}$ in the EM iterative process (see[13]), is known as winner-take-all[18] or classification EM (CEM) and it presents several advantages:

- Convergence is achieved in less steps, although not to the true maximum (but generally to a good approximation).
- Each iteration step is slightly faster.
- The final $\boldsymbol{\theta}$ value of this simplified iterative approach can be used as an initial value to accelerate the convergence of the exact method.

Initialization of $\boldsymbol{\theta}$. As in any iterative method, the election of the initial values of $\boldsymbol{\theta}$ is critical, because a bad choice can imply convergence to a local optimum. Nevertheless, there is partial information available for this initialization, using the symbol-by-symbol hard decisions (3) that would be made if the received codeword were just considered as uncoded information.

These hard decisions can be used to make the initial computation of (20), just as we have explained in the preceding simplification of the method. Nevertheless, notice that with this approach only values of $h(\boldsymbol{\theta}, z)$ corresponding to $|z| < \Delta/2$ can be initialized. All we can do in this initial iteration is to set the remaining values to a uniform non-zero value, and normalizing (12) so that it remains a pdf. These values cannot be initialized to zero, because these "impossible values" would penalize unacceptably the performance of the iterative decoding.

4. EXPERIMENTAL RESULTS

We present next the results of the tests carried out using turbo coding to empirically validate our approach. The turbo codes used in this section use pseudorandom interleavers, and the component coders have been chosen by trial-and-error, without using extensive optimizations.

The initialization of the pdf model follows the method explained in Section 3, and the updates of $\boldsymbol{\theta}$ are made with the suboptimal winner-take-all method. First we show in Figure 2 the decoding performance of the blind decoder proposed in front of Gaussian noise, for a pdf model (12) consisting of N_q kernel functions. We could tend to think that, the higher the number of kernel functions, the more accurate the estimation we could get. In principle this is true, but as the resolution N_q increases so does the variance of $\boldsymbol{\theta}$, and therefore the estimated pdf becomes eventually too noisy and useless for decoding. This behavior is reflected in Figure 2, where values of N_q up to 8 give increasingly better performance. Starting from that value we observe an increase in the probabilities of error, apart from a more erratic decoding profile —despite a sufficiently high number of simulations.

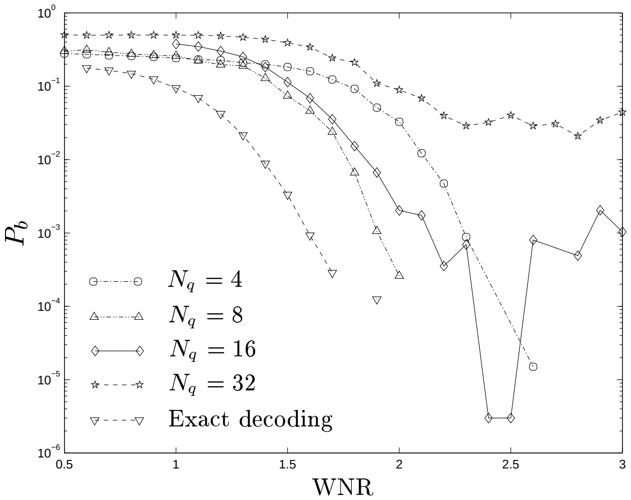

Figure 2. Gaussian noise. Performance of turbo-coded DC-DM with blind decoding for pdf models with different resolutions (number N_q of kernel functions in the non-parametric model), RSC (31 27)/31, $\nu = 0.65$, $M = 1,000$.

Comparing the best blind result in Figure 2 with the probabilities of error corresponding to exact decoding following Sect. 2 (in this case, for a better performing $\nu = 0.60$) we see that quite good results are possible for blind decoding. An interesting side effect of the blind method proposed is that decoding resembles to the use of a lookup table, which allows for a high decoding speed. We have also verified that, as it could be expected, the highest useful value of N_q is limited by M. For instance, for $M = 10,000$ the best N_q has been observed to be around 32. Also, a slight variability of the best N_q with different types of noise was detected.

The gain due to using a blind decoder instead of a Gaussian-matched one in the presence of non-Gaussian noise should be displayed for non-Gaussian distortions. For a fair comparison we will assume that the Gaussian-matched decoder estimates the noise power σ_n^2, as the knowledge of this parameter is too strong an assumption. This estimation can be made using the expression

$$E\{z^2[k]\} = E\{Q_{b[k]}^2(x[k])\} + (1-\nu)^2\Delta^2/3 + \sigma_n^2, \tag{21}$$

that can be easily derived from the expression of $z[k]$ for DC-DM. As $\mathbf{b}$ is not known initially by the decoder, the approximation $\sigma_x^2 \approx E\{Q_{b[k]}^2(x[k])\}$ can be used in (21). Then, using an estimate of the expectation $E\{z^2\}$ obtained from the received $\mathbf{z}$, we can write

$$\hat{\sigma}_n^2 = \frac{1}{M}\sum_{k=1}^{M} z^2[k] - \sigma_x^2 - (1-\nu)^2\Delta^2/3. \tag{22}$$

Unfortunately, this estimation is too slack, especially for high DWR's. A second, more accurate way to obtain the estimate is by means of iterative refinements of $\hat{\sigma}_n^2$ at each step of the turbo decoding process. Using the intermediate decoded information $\hat{\mathbf{b}}$ at a given turbo decoding iteration we can write

$$\hat{\sigma}_n^2 = \frac{1}{M}\sum_{k=1}^{M} \left[Q_{\hat{b}[k]}(z[k]) - z[k]\right]^2 - (1-\nu)^2\Delta^2/3. \tag{23}$$

This estimate is used on the next decoding step, that will serve to refine again $\hat{\sigma}_n^2$ and so on. In Figure 3 we show the performance obtained with these two approaches versus blind decoding when the attack is uniform i.i.d. noise. In the case of non-iterative estimation we have used DWR = 10 dB, instead of the 25 dB considered in the other two cases. The figure stresses the importance of having a good estimate of the channel variance in order to correctly decode the turbo-coded information, but in any case we can see that the blind method is able to yield a gain over the less adaptive Gaussian-matched one.

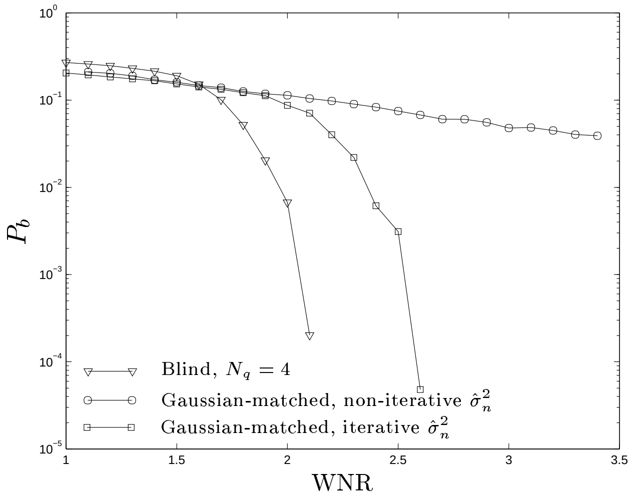

Figure 3. Uniform noise. Performance comparison of blind decoding versus Gaussian-matched decoding with noise variance estimation. RSC (31 27)/31, $\nu = 0.65$, $M = 1,000$.

5. CONCLUSIONS

Turbo codes permit to greatly enhance the properties of DC-DM. This is so because turbo codewords are random-like, as required by Costa's theorem, and we have an efficient way to decode them iteratively. Nevertheless, both the lack of channel model knowledge and the selection of the distortion compensation parameter involve important practical problems for their implementation. In this paper we have provided some lines to address these problems, showing how to undertake iterative decoding in a blind way. It is interesting to realize that the estimation method provided, involving the EM algorithm, is a quite general one, and quite amenable to solving many other estimation problems that are bound to appear in practical data hiding methods.

ACKNOWLEDGMENTS

Enterprise Ireland is kindly acknowledged for supporting this work under the Advanced Technologies Research Program (3GMARK). Work also funded by the *Xunta de Galicia* under grants PGIDT01 PX132204PM and PGIDT02 PXIC32205PN; CYCIT, AMULET project, reference TIC2001-3697-C03-01; and FIS, IM3 Research Network, reference FIS-G03/185.

APPENDIX A. OPTIMAL UPDATE OF $\boldsymbol{\theta}$ FOR BLIND ITERATIVE DECODING

First, notice that the expression (19) can be considerably simplified assuming the independence of the samples in $\mathbf{z}^s$ and $\mathbf{b}$, and using the linearity of the expectation operator:

$$\begin{aligned} F(\boldsymbol{\theta}) &\triangleq E_{q(\mathbf{b})}\{\log P(\mathbf{z}^s, \mathbf{b}, \boldsymbol{\theta})\} \\ &= E_{q(\mathbf{b})}\left\{\sum_{k=1}^{M} \log P(z^s[k], b[k], \boldsymbol{\theta})\right\} \\ &= \sum_{k=1}^{M} E_{q(\mathbf{b})}\left\{\log P(z^s[k], b[k], \boldsymbol{\theta})\right\}. \end{aligned} \tag{24}$$

A small digression now to discuss why the assumptions in the preceding paragraph are the reason for only considering the problem (15) and not (14). Even if it is possible to solve the E-step computing the pmf $q(\mathbf{c})$ for *all* the codeword symbols —using for instance the SISO algorithm by Benedetto et al.[19] that generalizes

BCJR—, the parity symbols cannot be assumed to be independent. This does not allow the simplification in (24), obscuring an analytical solution to the M-step.

Returning to our problem, and using again the independence of the $b[k]$, we can write

$$\begin{aligned} F(\boldsymbol{\theta}) &= \sum_{k=1}^{M} E_{q(b[k])}\left\{\log P(z^s[k], b[k], \boldsymbol{\theta})\right\} \\ &= \sum_{k=1}^{M} \sum_{b=\pm 1} q(b[k]=b) \log P(z^s[k], b[k]=b, \boldsymbol{\theta}). \end{aligned} \tag{25}$$

In order to undertake the optimization of this expression, we will find it convenient to rewrite it using some useful definitions that we will establish next. First, we define the intervals B_i of the support set corresponding to the i-th kernel in (12), that is,

$$B_i \triangleq \left((i-1)\cdot\Delta_q - \Delta \,,\, i\cdot\Delta_q - \Delta\right], \tag{26}$$

with $i = 1, \ldots, N_q$. Using them we can define in turn the sets of indices

$$\mathcal{P}_b^i \triangleq \left\{k \mid \tilde{z}_b^s[k] \in B_i\right\}, \tag{27}$$

with $b = \pm 1$, $i = 1, \ldots, N_q$, and $\tilde{z}_b^s[k]$ the modularization (7) applied on $z^s[k]$.

Now, (25) can be put as

$$F(\boldsymbol{\theta}) = \sum_{i=1}^{N_q} \sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k]=b) \log \theta[i]. \tag{28}$$

According to (19) we have now to maximize (28) with the restriction

$$\sum_{i=1}^{N_q} \theta[i] = 1, \tag{29}$$

that guarantees that (12) is a pdf. To this end, we build the Lagrangian

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N_q} \sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k]=b) \log \theta[i] - \gamma \left(\sum_{i=1}^{N_q} \theta[i] - 1 \right). \tag{30}$$

Differentiating with respect to $\theta[i]$, and equating to zero to obtain the extreme, we can write

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta[i]} = \sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k]=b) \frac{1}{\theta[i]} - \gamma = 0, \tag{31}$$

for $i = 1, \ldots, N_q$. The solution is a maximum due to the negativeness of the second derivative. In order to solve the Lagrange multiplier γ we just plug the solution of the equation above into the restriction (29), obtaining

$$\gamma = \sum_{i=1}^{N_q} \sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k]=b). \tag{32}$$

As $q(\mathbf{b})$ is a pmf, and as we are summing up in (32) the pmf's for every $b[k]$, we have that $\gamma = M$. Therefore, the optimal parameter vector $\boldsymbol{\theta}^*$ is given by the expression

$$\theta^*[i] = \frac{\sum_{b=\pm 1} \sum_{k \in \mathcal{P}_b^i} q(b[k]=b)}{M}, \quad i = 1, \ldots, N_q. \tag{33}$$

REFERENCES

1. M. H. Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory* **29**, pp. 439–441, May 1983.
2. B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory* **47**, pp. 1423–1443, May 2001.
3. J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. on Signal Processing* **51**, pp. 1003–1019, April 2003.
4. M. Kesal, M. K. Mıhçak, R. Koetter, and P. Moulin, "Iteratively decodable codes for watermarking applications," in *Proc. 2nd Symposium on Turbo Codes and Their Applications*, (Brest, France), September 2000.
5. J. J. Eggers, J. Su, and B. Girod, "Performance of a practical blind watermarking scheme," in *Proc. of SPIE, Security and Watermarking of Multimedia Contents III* **4314**, pp. 594–605, (San José, USA), January 2001.
6. J. Chou, S. Pradhan, and K. Ramchandran, "Turbo coded trellis-based constructions for data embedding: Channel coding with side information," in *Proc. of Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, USA), October 2001.
7. C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes (1)," in *Proc. IEEE Int. Conf. on Communications*, pp. 1064–1070, (Geneva, Switzerland), May 1993.
8. F. Pérez-González, F. Balado, and J. R. Hernández, "Performance analysis of existing and new methods for data hiding with known-host information in additive channels," *IEEE Trans. on Signal Processing* **51**, pp. 960–980, April 2003.
9. T. A. Summers and S. G. Wilson, "SNR mismatch and online estimation in turbo decoding," *IEEE Trans. on Communications* **46**, pp. 421–423, April 1998.
10. M. C. Reed and J. Asenstorfer, "A novel variance estimator for turbo-code decoding," in *Int. Conf. on Telecommunicacions*, pp. 173–178, (Melbourne, Australia), April 1997.
11. L. Wei, Z. Li, M. R. James, and I. R. Petersen, "A minimax robust decoding algorithm," *IEEE Trans. on Information Theory* **46**, pp. 1158–1167, May 2000.
12. X. Huang and N. Pahmdo, "Turbo decoders which adapt to noise distribution mismatch," *IEEE Communications Letters* **2**, pp. 321–323, December 1998.
13. Y. Li and K. H. Li, "Iterative PDF estimation and decoding for CDMA systems with non-Gaussian characterization," *IEE Electronics Letters* **36**, pp. 730–731, April 2000.
14. A. Dempster, N. Laird, and D. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B* **39**(1), pp. 1–38, 1977.
15. B. Lu, X. Wang, and K. R. Narayanan, "LDPC-based space-time coded OFDM systems over correlated fading channels: Performance analysis and receiver design," *IEEE Trans. on Communications* **50**, pp. 74–88, January 2002.
16. M. González-López, J. Míguez, and L. Castedo, "Turbo aided maximum likelihood channel estimation for space-time coded systems," in 13th *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, (Lisbon, Portugal), September 2002.
17. N. Noels, C. Herzet, A. Dejonghe, V. Lottici, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *IEEE Intl. Conference on Communications*, (Anchorage, USA), May 2003.
18. R. M. Neal and G. E. Hinton, *Learning in Graphical Models*, ch. A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants, pp. 355–370. MIT Press, Cambridge, 1999.
19. S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Comms. Letters* **1**, pp. 22–24, January 1997.