BLIND TURBO DECODING OF SIDE-INFORMED DATA HIDING USING ITERATIVE CHANNEL ESTIMATION

Félix Balado

University College Dublin
Belfield, Dublin 4, Ireland

Fernando Pérez-González

University of Vigo
Campus Universitario, 36200 Vigo, Spain

ABSTRACT

Distortion-Compensated Dither Modulation (DC-DM) has been theoretically shown to be a near-capacity achieving data hiding method, thanks to its use of side information at the encoder. In practice, channel coding is needed to approach its achievable rate limit. However, the most powerful coding methods, such as turbo coding, require knowledge of the channel model. We investigate here the possibility of undertaking blind iterative decoding of DC-DM. To this end, we undertake maximum likelihood estimation of the channel model, intertwining the Expectation-Maximization algorithm within the decoding procedure.

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 similar communications setting has been decisive to show that host-signal-induced self-distortion can be effectively removed through a clever design of the transmission codebook. In the context of data hiding this result was first pointed out by Chen and Wornell, who showed [2] that their DC-DM scheme was, asymptotically, only 1.53 dB away from Costa's capacity.

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 [3, 4, 5], using near-capacity achieving turbo codes over *scalar* side informed methods 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. It is a fact that the type of channel and the level of distortion are necessary for undertaking iterative decoding. Incidentally, all these works have worked under the hypothesis that this information was known by the decoder. Here we explore how to perform blind[1] iterative decoding of DC-DM, i.e., without the aforementioned assumptions.

Enterprise Ireland is kindly acknowledged for supporting this work (Grant ATRP-230). Work also funded by *Xunta de Galicia* under grants PGIDT01 PX132204PM and PGIDT02 PXIC32205PN; CYCIT, AMULET project, ref. TIC2001-3697-C03-01, FIS, IM3 Research Network, ref. FIS-G03/185, and the European NoE E-CRYPT

[1]This term refers to the ignorance of the channel model by the decoder; not to be confused with blind vs. non-blind data hiding.

1.1. Framework

We assume that we pseudorandomly choose N samples $\mathbf{x} = (x[1], \ldots, x[N])$ from a host signal; the samples in $\mathbf{x}$ are independent identically distributed (i.i.d.) zero-mean random variables with covariance matrix $\Gamma_x = \sigma_x^2 \cdot I$. The corresponding watermarked signal $\mathbf{y}$ 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 i.i.d. and independent of $\mathbf{x}$, with unknown probability density function (pdf) and variance σ_n^2.

In binary DC-DM [2] one information symbol $b[k] \in \{\pm 1\}$ is hidden by quantizing a sample of the host signal $x[k]$ to the nearest centroid $Q_{b[k]}(x[k]) \in \Lambda_{b[k]}$ belonging to the uniform lattice[2] $\Lambda_{b[k]}$ given by

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

with $d[k]$ a key-dependent value that can we taken as zero for the analysis. The watermarked signal is obtained as

$$y[k] = x[k] + \nu \cdot e[k] = Q_{b[k]}(x[k]) - (1-\nu)\cdot e[k], \qquad (1)$$

i.e, the watermark is the quantization error $e[k] \triangleq Q_{b[k]}(x[k]) - x[k]$ weighted by an optimizable constant ν, $0 \leq \nu \leq 1$. The relation $\Delta \ll \sigma_x$ usually holds true due to perceptual reasons. Then, for a wide range of hosts, $e[k]$ can be assumed to 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

$$\hat{b}[k] = \arg\min_{b\in\{\pm 1\}} \left|Q_b(z[k]) - z[k]\right|. \qquad (2)$$

Following what we stated in the introduction, we will hide a binary codeword $\mathbf{c} = (c[1], \ldots, c[N])$ instead of hiding N uncoded bits using the previous scheme. 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 embedding and decoding we will consider that the codeword symbols are given in antipodal form, i.e., $c[k] \in \{\pm 1\}$. We will center our attention in parallel concatenated codes with iterative decoding, i.e., turbo codes. We recall that the parallel concatenated turbo codewords have the form

$$\mathbf{c} = (\mathbf{c}^s \,|\, \mathbf{c}^{p_1} \,|\, \mathbf{c}^{p_2}), \qquad (3)$$

[2]Extending the usual definition of lattice, which in principle must include the origin.

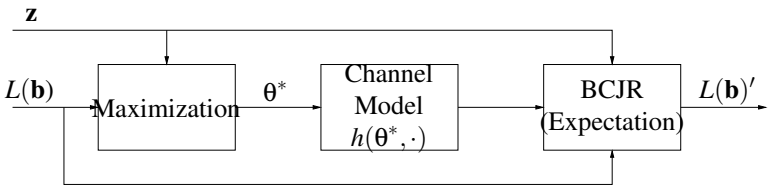

Fig. 1. One step of the iterative EM algorithm intertwined with iterative turbo decoding. Necessary interleavings/deinterleavings of $\mathbf{z}^s$ and $L(\mathbf{b})$ for BCJR are not explicitly shown for simplicity.

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 choice of ν is important because there is a different optimum at each watermark-to-noise ratio (WNR) for the achievable rate of DC-DM [4]. The WNR $= 10\log_{10}\nu^2\Delta^2/(3\sigma_n^2)$ is not known beforehand by the encoder, as he/she cannot know σ_n. Previous works [3, 4, 5] have worked under this assumption, and so they have used the optimal scaling of their lattices —i.e., the optimal distortion compensation factor ν— at each WNR. Here, we will use a fixed ν regardless of the WNR, what is more realistic. Turbo codes present a distinctive waterfall of the decoded bit error rate at a WNR value relatively close to the minimum for asymptotically errorless decoding. Then, 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 the code cannot be perfect, the optimum will actually correspond to a slightly higher WNR. Notice however that this choice requires knowledge of the channel model for computing the achievable rate vs. WNR plots [4], but it is all it can be done. In addition, this optimization 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

First, 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. The decoder receives the noisy signal $\mathbf{z}$ and proceeds to perform MAP iterative decoding. This requires the probabilities $p(z[k] \mid c[k] = c)$, $c \in \{\pm 1\}$, for computing the reliability log-likelihood ratios. Considering (1), we have that $y[k]$ is uniform, and then that the pdf of $z[k] = y[k] + n[k]$ is the convolution of a uniform and a Gaussian pdf's. We can put this pdf as $f(z[k]) * \delta\{z[k] - Q_c(x[k])\}$, with

$$f(z) \triangleq \frac{1}{2(1-\nu)\Delta}\left\{Q\left(\frac{z-(1-\nu)\Delta}{\sigma_n}\right) - Q\left(\frac{z+(1-\nu)\Delta}{\sigma_n}\right)\right\}, \quad (4)$$

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 (2) at the decoder, the decision $\hat{c}[k]$ can be seen as being based on the modular offsets

$$\tilde{z}_c[k] \triangleq \{z[k] \mod \Lambda_c\} - \Delta = \left\{z[k] + \Delta\frac{(c+1)}{2}\right\} \mod 2\Delta - \Delta \quad (5)$$

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

$$\hat{c}[k] = \arg\min_c \left|\tilde{z}_c[k]\right|. \quad (6)$$

Considering (6), it is clear that the reliability measure for the decision $\hat{c}[k] = c$ is just

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

with $\tilde{f}(\cdot)$ the pdf followed by $\tilde{z}_c[k]$. Notice that the operation (5) implies that this pdf is just the aliasing of the sections of (4) corresponding to the Voronoi regions of the lattice $2\Delta\mathbb{Z}$, that is

$$\tilde{f}(z) = \begin{cases} \sum_{w\in 2\Delta\mathbb{Z}} f(z-w), & |z| \le \Delta \\ 0, & |z| > \Delta \end{cases}. \quad (7)$$

3. BLIND ITERATIVE DECODING OF DC-DM

We assume next that the decoder does not know (7). In the communications field, we can find can some approaches that estimate blindly the pdf of an unknown additive channel such as the one by Li et al. [6], who propose to heuristically refine a kernel-based model at each iterative decoding step using the increasingly accurate intermediate decoded information. We will follow a similar approach, but using sounder theoretical grounds. Taking profit that the support set of $\tilde{f}(z)$ is limited to $|z| < \Delta$, we can resort to approximating (7) using a simple but general model based on a finite number N_q of rectangular kernels. This model depends on the parameters vector $\theta = (\theta[1], \ldots, \theta[N_q])$ and it is given by

$$h(\theta, z) \triangleq \sum_{i=1}^{N_q} \theta[i] \cdot \Pi\big(z - (i-1)\cdot\Delta_q + \Delta\big). \quad (8)$$

In the expression above the kernels $\Pi(z)$ are defined as

$$\Pi(z) \triangleq \begin{cases} 1/\Delta_q, & 0 < z \le \Delta_q \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

with $\Delta_q \triangleq 2\Delta/N_q$, which we assume integer. Of course, $h(\theta, z) = 0$ for $|z| > \Delta$. Notice that a further advantage of (8) is that it makes no assumptions on the symmetry of the attack pdf. This model is usually considered to be nonparametric, although we can see it as a parametric one in which θ has to be adjusted.

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

$$\hat{\theta} = \max_\theta P(\mathbf{z}, \theta). \quad (10)$$

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 [7] was conceived, aiming at finding the solution of (10) iteratively with theoretically proven convergence properties. Unfortunately, we cannot afford the hypothesis of independence between the elements of $\mathbf{z}$ that correspond to the codeword parities, what obscures the solution to (10). For this reason we will resort to solving instead

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

with $\mathbf{z}^s$ the subvector of $\mathbf{z}$ corresponding to the systematic part $\mathbf{c}^s = \mathbf{b}$ of the codeword $\mathbf{c}$, following the notation in (3). Anyway, and as we will see next, the turbo code can be used to improve the EM algorithm beyond what we could get with $\mathbf{z}^s$ alone. In this way, we can intertwine the iterative turbo decoding with the iterative estimation problem. We describe next the two steps of the EM algorithm and their application to our problem, that is summarized in Figure 1.

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 θ, that is

$$q(\mathbf{b}) \triangleq P(\mathbf{b} \mid \mathbf{z}^s, \theta). \tag{11}$$

   Actually, each iterative turbo decoding stage optimally updates the previous *extrinsic* pmf of $\mathbf{b}$ using the BCJR algorithm, which takes into account $\mathbf{z}$ (and not only $\mathbf{z}^s$), the code used for the current parity, and the channel model given by θ. Therefore, the probabilities $q(b[k])$, for $k = 1, \ldots, M$, given by the BCJR algorithm, are the best way to compute (11). Assuming that the information bits $b[k]$ are independent, we can write

$$q(\mathbf{b}) = \prod_{k=1}^{M} q(b[k]). \tag{12}$$

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

2. **Maximization Step**. Now, using the pdf (12) and $\mathbf{z}^s$ we need to compute the new θ that maximizes the EM functional [7], that can be written as

$$\max_{\theta} E_{q(\mathbf{b})}\{\log P(\mathbf{z}^s, \mathbf{b}, \theta)\}. \tag{13}$$

   It is shown in Appendix A that the solution θ^* to this optimization problem is given by the expression (17).

After the maximization step we may go back to the expectation step, for which a new iteration of turbo decoding is performed using the increasingly more reliable pdf updated using (17) (see Figure 1). This procedure is continued until convergence.

In order to gain further insight from (17) we can consider to use, instead of the soft values $q(b[k])$, the decisions $\hat{b}[k] = \operatorname{sign} L(b[k])$ 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 (17) becomes the normalized histogram of $\mathbf{z}^s$ on the bins B_i defined in the Appendix, using the hard decisions $\hat{b}[k]$ to make the bin assignments of the corresponding $z^s[k]$. This decision-based approximation, that would be the intuitive way to update θ in the EM iterative process (see [6]), achieves convergence in less steps, and generally to a good approximation of the real optimum.

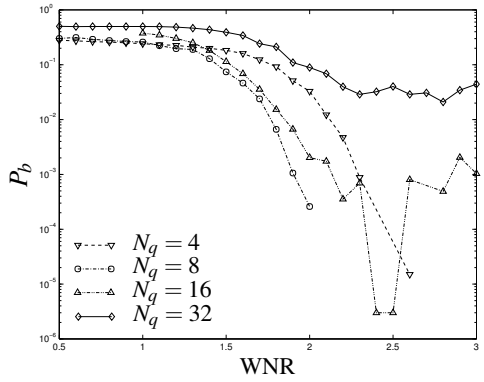

Fig. 2. Gaussian noise. Performance of turbo-coded DC-DM with blind decoding for pdf models with different resolutions.

Last, there is partial information available for the initialization of θ, using the symbol-by-symbol hard decisions (2) 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 (17), just as we have explained in the preceding simplification of the method. Nevertheless, notice that with this approach only values of $h(\theta, z)$ 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 (8) 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 some results of the tests carried out using turbo coding and the suboptimal intuitive updates of θ. We use the RSC (1 27/31), a pseudorandom interleaver with size $M = 1000$ and $\nu = 0.65$. First we show in Figure 2 the decoding performance of the blind decoder proposed in front of Gaussian noise, for a pdf model (8) 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 θ, and therefore the estimated pdf becomes eventually too noisy and useless for decoding, as we can see in the figure for values $N_q > 8$.

For non-Gaussian distortions the gain due to using a blind decoder instead of a Gaussian-matched one should be displayed. For a fair comparison we assume that the Gaussian-matched decoder

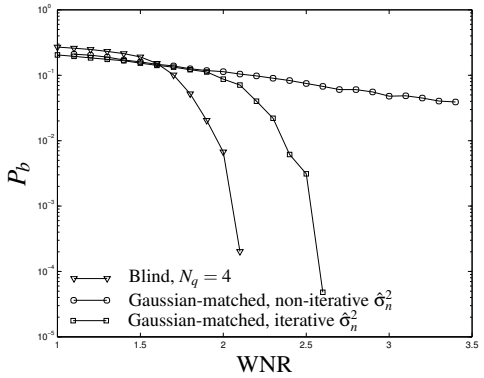

Fig. 3. Uniform noise. Performance comparison of blind decoding versus Gaussian-matched decoding with noise variance estimation.

estimates the noise power σ_n^2 using the expression

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

and iteratively refining this estimation over successive decoding steps. In Figure 3 we show the performance obtained with this approach versus blind decoding when the attack is uniform i.i.d. noise. We also show the Gaussian-matched decoder with a non-iterative estimation of σ_n^2, stressing the importance of having a good estimate of the channel variance in order to correctly decode the turbo-coded information. We can see that the blind method is able to yield a gain over the less adaptive Gaussian-matched one.

5. REFERENCES

[1] Max H.M. Costa, "Writing on dirty paper," *IEEE Trans. on Information Theory*, vol. 29, no. 3, pp. 439–441, May 1983.

[2] Brian Chen and Gregory W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol. 47, no. 4, pp. 1423–1443, May 2001.

[3] 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.

[4] J.J. Eggers, R. Bäuml, R. Tzschoppe, and B. Girod, "Scalar costa scheme for information embedding," *IEEE Trans. on Signal Processing*, vol. 51, no. 4, pp. 1003–1019, April 2003.

[5] 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.

[6] Yuan Li and Kwok H. Li, "Iterative PDF estimation and decoding for CDMA systems with non-Gaussian characterization," *IEE Electronics Letters*, vol. 36, no. 8, pp. 730–731, April 2000.

[7] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

A. OPTIMAL UPDATE OF THE PARAMETERS

Assuming independence of the samples in $\mathbf{z}^s$ and $\mathbf{b}$ we can write (13) as

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

Using again the independence of the $b[k]$, we can write

$$\begin{aligned} F(\theta) &= \sum_{k=1}^{M} E_{q(b[k])}\{\log P(z^s[k], b[k], \theta)\} \\ &= \sum_{k=1}^{M}\sum_{b=\pm 1} q(b[k]=b)\log P(z^s[k], b[k]=b, \theta). \end{aligned} \quad (14)$$

We will find it convenient next to rewrite (14) using some useful definitions. First, we define the intervals B_i of the support set corresponding to the i-th kernel in (8), that is, $B_i \triangleq \big((i-1)\cdot\Delta_q - \Delta, i\cdot\Delta_q - \Delta\big]$, with $i = 1,\ldots,N_q$. Using them we can define in turn the sets of indices

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

with $b = \pm 1$, $i = 1,\ldots,N_q$, and $\tilde{z}_b^s[k]$ the modularization (5) applied on $z^s[k]$. Now, (14) can be put as

$$F(\theta) = \sum_{i=1}^{N_q}\sum_{b=\pm 1}\sum_{k\in\mathcal{P}_b^i} q(b[k]=b)\log\theta[i]. \quad (15)$$

According to (13) we have now to maximize (15) with the restriction $\sum_{i=1}^{N_q}\theta[i] = 1$, that guarantees that (8) is a pdf. To this end, we build the Lagrangian

$$L(\theta) = F(\theta) - \gamma\left(\sum_{i=1}^{N_q}\theta[i] - 1\right).$$

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

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

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 obtaining

$$\gamma = \sum_{i=1}^{N_q}\sum_{b=\pm 1}\sum_{k\in\mathcal{P}_b^i} q(b[k]=b). \quad (16)$$

As $q(\mathbf{b})$ is a pmf, and as we are summing up in (16) the pmf's for every $b[k]$, we have that $\gamma = M$. Therefore, the optimal parameter vector θ^* is given by the expression

$$\theta^*[i] = \frac{\sum_{b=\pm 1}\sum_{k\in\mathcal{P}_b^i} q(b[k]=b)}{M}, \quad i = 1,\ldots,N_q. \quad (17)$$