Journal of Information Hiding and Multimedia Signal Processing ©2011 ISSN 2073-4212
Ubiquitous International Volume **2**, Number **2**, April **2011**

High Payload Audio Watermarking: toward Channel Characterization of MP3 Compression

Raúl Martínez-Noriega
Graduate School of Electro-Communications
University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, 182-8585, Japan
raul@ice.uec.ac.jp

Mariko Nakano
Research Section and Graduate School
National Polytechnic Institute
1000 Av. Santa Ana, Coyoacan, 04430, Mexico
mnakano@ipn.mx

Brian Kurkoski and Kazuhiko Yamaguchi
Graduate School of Informatics and Engineering
University of Electro-Communications
1-5-1 Chofugaoka, Chofu-shi, 182-8585, Japan
kurkoski@ice.uec.ac.jp; yama@ice.uec.ac.jp

Received April 2010; revised July 2010

Abstract. *A new audio watermarking algorithm resilient to MPEG 1 layer III (MP3) with bit rate of 64 kbps is proposed. High embedding capacity is its main characteristic. The proposed algorithm is able to use either semi-blind or blind decoding. With the former, the algorithm achieves an embedding capacity of 204 bits per second (bps) and with the latter obtains 155 bps, them both with bit error rate lower than* 10^{-4}. *Using semi-blind decoding, benefit of more than 33 bps is obtained in comparison with, in our knowledge, any other previously proposed algorithm for compression at 64 kbps. The algorithm uses dither modulation to embed a coded-watermark in certain frequencies of wavelet domain. Those frequencies were found to be robust against compression. The coded-watermark is generated with the concatenation of low-density parity-check (LDPC) codes and repetition codes. Also, an accurate estimation of the statistics of the channel noise, due to compression, is introduced. Finally the proposed algorithm is improved using erasures at the decoder. This improvement achieves more than 15 bps in comparison with normal decoding. The highest embedding capacity achieved is 229 bps. All the watermarked audio files obtain SNR higher than 40 dB. The audio quality was also measured with a subjective evaluation in which the watermarked audio obtained a score higher than 4, where the best audio quality is scored with 5.*

Keywords: Audio watermarking, MP3 compression, dither modulation, LDPC codes

1. **Introduction.**

Data hiding techniques provide a way to embed extra information within the original contents, without serious degradation of the quality. With the development of proper and robust data hiding systems, more technologies have found new and promising applications. For example, high payload techniques have been used to extend narrow bands from 8 kHz to 16 kHz in telephony speech [1], that results in high quality speech with the same physical resources. Another uses of data hiding are: adding commercials in free media, usage control, index information for data bases and so on.

Nowadays, MPEG 1 layer III (MP3) is a common format of compressed digital audio. Its popularity is because MP3 offers good audio quality within small storage space. As repercussion, on-line stores have proliferated because compression algorithms allow to deliver the audio files in a short period of time through the Internet.

Digital watermarking is a data hiding technique with a trade-off between the embedding capacity, robustness and quality. Generally, when applications of digital watermarking are developed, limits on robustness and quality must be defined. Therefore, would be desirable to increase the payload meanwhile the other constraints are kept as much as possible because with a higher payload, digital watermarking can be applied to a wider number of applications.

In this paper, we focus on the embedding capacity[1] for audio files. Our aim is to develop watermarking schemes with high payload and low probability of error against the lowest compression with commercial value, i.e. radio quality which is equivalent to MP3 with a bit rate of 64 kbps. This scenario is detailed in Sec. 2.

Since the pioneer paper of Boney [2] in 1996 for audio watermarking, many algorithms with robustness to MP3 compression have been proposed. For example in 2003, Cvejic [3] proposed a scheme using spread spectrum and characterization of the attack, with a payload of 27.1 bits per second (bps). In the same year, In-Kwon [4] used patchwork algorithm to embed the watermark in audio files producing a scheme with 10 bps. Later, in 2004, Wang [5] introduced a decoding algorithm using linear predictive coding to recover the watermark from wavelet domain; the achieved payload was 10.72 bps. Another two proposals [6] and [7] were described in 2006, the former includes neural networks to increase the robustness and the latter proposes a solution for time-scale modification, obtaining 86 bps and 4.26 bps respectively. Between 2007 and 2008, Xiang presented two schemes in [8] and [9] based on modifying the histogram of the audio files; with embedding rates of 3 bps and 2 bps. At the same time, two papers with high embedding rates were published, [10] with a payload of 170 bps and [11] with 220 bps; however those payloads were achieved with MP3 of 128 kbps and 96 kbps respectively. More recently, in 2009 Fan [12] proposed an embedding of a chaotic-based watermark on discrete fractional sine transform domain, with 86 bps of payload. Also, in the same year, Wang [13] introduced a robust algorithm against MP3 at 64 kbps but the embedding payload was not reported. Finally, in 2010 a self-synchronized algorithm was introduced by Megías [14] with an embedding rate of 30.09 bps. Almost all the works presented above have an embedding rate lower than 100 bps and those with payload higher than 100 bps are not robust to MP3 with quality of 64 kbps.

Among the variety of embedding algorithms, quantization-based are very popular because its ease of implementation, computational flexibility, high embedding rate and amenability to theoretical analysis. In 1999, Chen *et ál.* [15] introduced a new quantization-based data hiding algorithm called quantization index modulation (QIM). Since then, several variations have been developed. For example, in [15] Chen also proposed dither modulation (DM) and, Pérez-González in [16] introduced an invariant method to gain attacks called rational dither modulation (RDM).

DM is a practical implementation of QIM. Its main characteristic is the use of scalar quantizers. DM incorporates a private key for decoding and it has low-complexity. A formal description of DM is included in Sec. 3. In our proposal, we use DM to embed a coded-watermark in wavelet domain. Similar proposals can be found in [17] and [18]. The former is an algorithm with self-synchronization in wavelet domain that embeds the watermark in the low-frequency coefficients. The scheme achieves 172 bps, however is well-known that modifications to low-frequencies produce audible distortion, and the authors did not report evidence of the audio quality. The latter is a recent proposal, 2010, that embeds the watermark by quantizing the Euclidean norm of a singular value

[1] The term “capacity” refers to watermark payload and it is different from the theoretical channel capacity defined by information theory.

decomposition of an audio segment. The reported payload is 196 kbps, however the bit error probabilities are computed with short simulations, around 5 seconds.

Our algorithm embeds a coded-watermark in a specific range of frequencies which were found to be robust against MP3, Sec. 4. The coded-watermark was obtained with the concatenation of an outer low-density parity-check (LDPC) encoder and inner repetition codes. Watermarking channels tend to have very high bit error rate (BER), therefore repetition codes are needed to increase the signal-to-noise (SNR) ratio [19]. LDPC codes are powerful error-control coding (ECC) suitable for very noise channels in which common ECC like BCH codes or convolutional codes have shown to be inefficient [20].

LDPC codes [21] have stated near performance to the theoretical Shannon limit for noisy channels. They work under the general principle: the longer the code length, the closer from the theoretical channel capacity. Unlike BCH codes, decoding complexity of LDPC codes increase linearly to its code length.

Audio watermarking with powerful ECC can be found in [22] and [23]. Turbo codes together with spread spectrum are used in the former, achieving 21.6 bps. The latter is our previous proposal which has an embedding capacity of 61.25 bps for MP3 at 64 kbps. Specially LDPC codes have been already used, but in image watermarking [24]; they produced very high embedding rates, however only normal decoding is applied.

The proposal of this paper is explained in Sec. 5. The decoder is implemented in two versions: semi-blind and blind decoding. In semi-blind decoding, the "watermarked audio without noise" is compared with the noisy audio in order to compute the noise variance. The statistics of the channel noise, variance, can be approximated with different techniques that are not the aim of this paper. However we implemented a blind decoding with the aid of pilot symbols.

The proposed algorithm achieves an embedding capacity of 155 bps with blind decoding and 204 with semi-blind decoding, shown in Sec. 6. Then, the proposal was improved using erasures at LDPC decoder, Sec. 7. The result is an embedding capacity of 229.7 bps which represents a benefit of more than 15 bps in comparison with normal decoding and more than 33 bps compared with any other proposal robust to MP3 at 64 kbps, Sec. 8.

It is worth to mention that there are very high payload techniques for data hiding in audio which achieve embedding rates of 689 bps in [25], 2996 bps in [26] and 11000 bps in [27] but they are not robust to MP3 of 64 kbps.

Finally, the last two Sections, 9 and 10, are dedicated to the audio quality and conclusions respectively.

2. **Problem Statement and Preliminaries.** Digital watermarking has a trade-off between watermark robustness, embedding capacity and audio quality. That is, if one of them is increased the other two will be affected.

On the other hand MP3 represents a very popular audio format. However, it is considered as one of the strongest attacks for audio watermarking.

Assuming that in a certain watermarking scheme the audio quality and the watermark robustness is fixed, would be desirable to obtain the highest reliable payload. This paper addressed how to develop audio watermarking algorithms with high embedding capacity but also robust to MP3 with bit rate of 64 kbps. Fig. 1 pictures the considered scenario.

Bold letters, e.g. $\mathbf{X}$ or $\mathbf{x}$, represent vectors and X_i or x_i refer to their respective ith element.

3. **Dither Modulation.** Dither modulation (DM) is a data hiding algorithm based on quantization and proposed by Chen *et ál.* [15]. DM is a practical and low-complexity implementation of QIM with scalar and uniform quantizers.

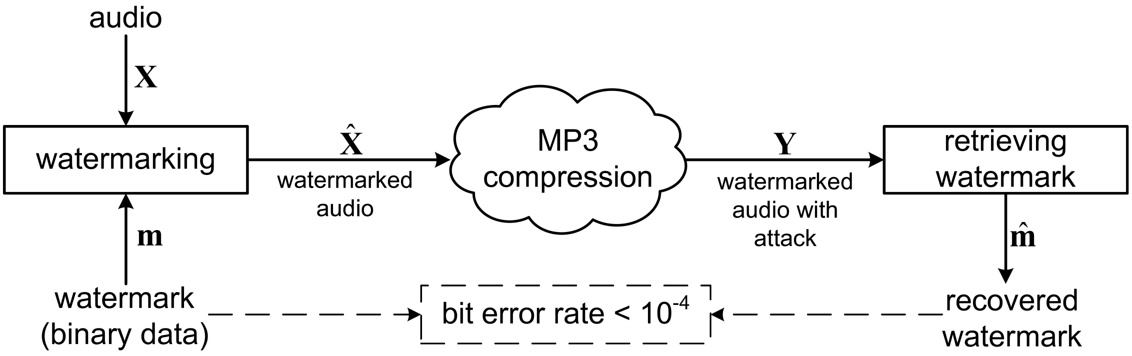

Figure 1. General diagram of the studied system.

DM uses only one base quantizer Q and two dithered parameters if the watermark is binary. The first dithered parameter $v(0)$ is generated in pseudo-random way with an uniform distribution between $[-\Delta/2, \Delta/2]$, where Δ is the quantization step size. The second parameter $v(1)$ is computed according to:

$$v(1) = \begin{cases} v(0) + (\Delta/2) & \text{if } v(0) < 0 \\ v(0) - (\Delta/2) & \text{if } v(0) \geq 0. \end{cases}$$

In that way, it is ensured that both dithered parameters differ in $\Delta/2$ each other.

DM embedding function is represented by:

$$\hat{X}_i = Q\big(X_i + v(m_i), \Delta\big) - v(m_i),$$

where $\mathbf{X} = X_1, X_2, \dots$ is the host signal, $\mathbf{m} = m_1, m_2, \dots$ is a binary watermark and $\hat{\mathbf{X}} = \hat{X}_1, \hat{X}_2 \dots$ is the watermarked host signal.

Watermark extraction is conveyed by measuring the distance between the watermarked sample Y_i and its closest reconstruction point, where $\mathbf{Y} = Y_1, Y_2, \dots$ is the watermarked audio with noise. The recovered bit, $\hat{m}_i \in \{0, 1\}$, is the argument which minimizes:

$$\hat{m}_i = \arg\min_j |Y_i - Q\big(Y_i + v(j), \Delta\big) + v(j)| \quad \text{for } j = 0, 1.$$

4. **Repercussions of MP3 on Wavelet Domain.** When a signal is analyzed, some of its characteristics may be not readily seen in time domain. Mathematical transformations are tools which allow us to represent signals in other spaces where some specific information is more readily available.

Wavelet is a mathematical transform which provides a time-frequency representation. Unlike Fourier transform, wavelet transform produces more accurate information about the signal's frequencies and its position on time, specially for non-stationary signals like audio files. Although the wavelet natural representation is based on scale-time, the representation of the information in frequency-time is straightforward.

Wavelet transform offers a decomposition oriented toward low-frequencies. In other words, if we picture the wavelet decomposition as binary tree, where the left node represents the low-frequencies and right node the high-frequencies then, the decomposition is applied always to the left node and the right node keeps without any further decomposition. Wavelet packet offers richer analysis by decomposing not only the left node but also the right one. In each level, both nodes are decomposed until a desired level.

In this paper, we are focused in one of the most popular manipulation on audio files, MP3 compression. The key to MP3 is lossy. Nonetheless, this algorithm can give transparent, perceptually lossless compression. To achieve this transparency, the compression is done according to the human auditory system which has different responses depending on the frequency. Therefore, it is expected that different range of frequencies of the audio files will be affected with different intensity.

When the audio file is decomposed with wavelet packet, each sub-band represents a different range of frequencies. We are interested in measuring the distortion on different sub-bands after MP3 compression to find suitable sub-bands for embedding.

The audio files were divided in blocks of 512 samples and wavelet packet decomposition was applied to each block using 5 levels. Simple *Haar* was used as mother wavelet. After decomposition, 32 wavelet sub-bands are obtained. Then, the binary watermark was embedded in each sub-band individually using DM and repetition codes, no other ECC was involved. Finally, the audio files were reconstructed and attacked with MP3 compression at 64 kbps.

Two experiments were conducted. First, Fig. 2a shows the error percentage of each wavelet sub-band for three different embedding rates 10.76, 21.53 and 43.06 bps. Classic music: *Egmont Op. 84* mono, 44.1 kHz/16 bits and with 8 minutes long was used. Second, Fig. 2b shows the bit error percentage of each wavelet sub-band for three different music genres with an embedding rate of 21.53 bps. Every music gender was a cluster with 10 different audio files of 2 minutes long each.

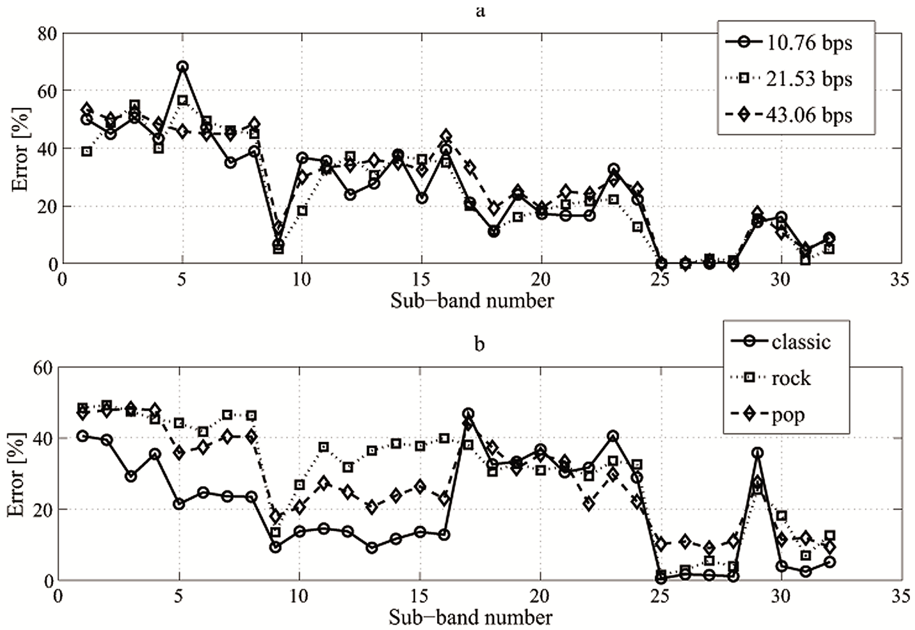

Figure 2. Error percent for different wavelet sub-bands after MP3 at 64 kbps. “a” using different embedding rates and “b” for different music.

In both results the wavelet sub-bands 25, 26, 27 and 28 obtained better robustness against compression, those sub-bands belong to frequencies ranging from 11 kHz to 13.7 kHz.

With Fig. 2a is guaranteed that the sub-band robustness behaves similar for different embedding rates, and then, the result is expanded to different music with Fig. 2b. In the simulations, Δ was fixed according to SNR $>$ 30 dB. We define SNR as the ratio between original audio signal power and that of the audio file with information embedded, $\mathrm{SNR} = 10\log_{10}(\sigma^2_{\mathbf{X}}/\sigma^2_{\mathbf{S}})$, where $\mathbf{S} = \hat{\mathbf{X}} - \mathbf{X}$.

5. **Proposed Method.** The main idea is to embed a coded-watermark inside wavelet domain using DM. The coded-watermark is obtained by concatenation of an outer LDPC code and inner repetition codes. The watermark is decoded by retrieving soft-information with DM, computing the metric and finally decoding the metric with sum-product [28] algorithm. The statistics of the channel noise, noise variance σ^2, are involved in the computation of the metric and the performance of the sum-product algorithm depends highly in a good estimation of σ^2. These ideas are summarized as block diagram in Fig. 3.

Two different decoding strategies are introduced. The first one considers general statistics of the channel noise to compute the metric, i.e. the log-likelihood ratio **llr**, and the second one computes independent statistics about the noise for different segments along the audio file.

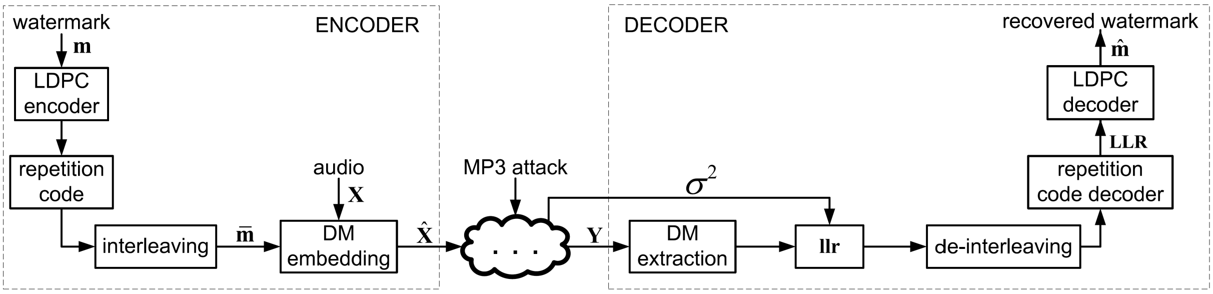

FIGURE 3. General watermarking scheme.

5.1. **Embedding.** In the embedding process, the audio file in time domain $\mathbf{X}$ is divided in non-overlapped blocks, T_h, of 512 samples. Wavelet packet decomposition is applied in five levels using simple *Haar* wavelet. From each time-domain block T_h, 32 wavelet sub-bands with 16 coefficients each are obtained. According to Sec. 4, only coefficients from the sub-bands 25 to 28 will be used for embedding. From now on, we will use the notation "25-28" to means wavelet sub-bands from 25 to 28. Frequency coefficients from 25-28 are arranged in a vector $\mathbf{x} = x_1, x_2, \ldots, x_n$ which contains not only the coefficients from the block T_h but from all blocks ordered according to time.

The binary watermark $\mathbf{m} = m_1, m_2, \ldots$ is firstly encoded with a Margulis half-rate LDPC code with code length of 2640. The output of the LDPC encoder is again encoded with an inner repetition code of length l, the output is permuted with an interleaving and finally the coded-watermark $\bar{\mathbf{m}} = \bar{m}_1, \bar{m}_2, \ldots, \bar{m}_n$ is obtained.

The watermark robustness is variable depending solely on the repetition code because the LDPC code will be the same and also the quantization step size Δ is fixed. If the repetition code is larger then the robustness is better but the embedding capacity is lower and vice versa.

Each bit from the coded-watermark $\bar{\mathbf{m}}$ is embedded in one coefficient x_i using DM, $\hat{x}_i = Q\big(x_i + v(\bar{m}_i), \Delta\big) - v(\bar{m}_i)$ where Q represents the quantization function with step Δ, $v(\bar{m}_i)$ is a dithered parameter that modulates the watermark symbol $\bar{m}_i$ and $\hat{x}_i$ is the watermarked coefficient. Then $\mathbf{x}$ is replaced with $\hat{\mathbf{x}}$ and the inverse wavelet packet transform is applied.

The watermarked audio $\hat{\mathbf{X}}$ is susceptible to suffer any sort of degradation due to common signal manipulations, e.g. MP3 compression, or direct attacks which attempt to destroy the watermark. Therefore, $\mathbf{Y}$ is the watermarked audio with degradations or attacks.

5.2. **Decoding.** $\mathbf{Y}$ is divided in blocks $\hat{T}_h$ of 512 samples and wavelet packet in five levels is computed. Watermarked coefficients from 25-28 are extracted and arranged as the vector $\mathbf{y} = y_1, y_2, \ldots, y_n$. The distance $\mathbf{d} = d_1, d_2, \ldots, d_n$ is computed as the absolute value of $\mathbf{y}$ and its closest reconstruction point with respect to $v(0)$:

$$d_i = \left|y_i - Q\left(y_i + v(0), \Delta\right) + v(0)\right|.$$

The soft information $\mathbf{r} = r_1, r_2, \ldots, r_n$ is computed with $\mathbf{r} = \mathbf{d} - \Delta/4$. At this point hard-decision decoding, i.e. raw decision between zeros and ones, could be applied by defining $\hat{m}_i = 1$ if $r_i > 0$ or $\hat{m}_i = 0$ otherwise. However, it is well-known that soft-decision performs better, therefore hard-decision decoding will not be treated in this paper.

LDPC decoder uses the $\mathbf{llr} = llr_1, llr_2, \ldots, llr_n$ as metric for sum-product algorithm. The llr_i of a embedded bit $\bar{m}_i$ is computed as the ratio between the probability that a given value r_i could be 1 (represented by $\Delta/4$) or 0 (represented by $-\Delta/4$):

$$llr_i = \ln \frac{P(\bar{m}_i = \Delta/4|r_i)}{P(\bar{m}_i = -\Delta/4|r_i)}. \tag{1}$$

Knowledge of statistics of the channel noise are needed to compute (1). Heuristically, we have seen that the noise, due to MP3, in the coefficients of 25-28 behave very similar to a Gaussian distribution. Therefore, the $\mathbf{llr}$ is computed with a Gaussian kernel:

$$\begin{aligned} llr_i &= \ln \left(\frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_i-(\Delta/4))^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(r_i+(\Delta/4))^2}{2\sigma^2}\right)} \right), \\ &= \frac{\Delta r_i}{2\sigma^2}, \end{aligned}$$

where σ^2 is the noise variance.

In the next Subsections, two different decoding variations are explained. However all the steps explained above are common for them both.

5.2.1. *Global Noise Variance.* A good estimation of the noise variance σ^2 is important to obtain good performance. In traditional communications many sources of noise can be modelled as Gaussian, moreover in many of them a single noise variance characterizes the channel. Digital watermarking has been shown to be a form of communications and also the noise in wavelet coefficients, due to MP3, behave similar to Gaussian. Therefore, this approach is an analogy to those communication systems which use a single noise variance to model the noise in the whole channel.

Log-likelihood ratio is computed straightforward with:

$$\mathbf{llr} = \frac{\Delta \mathbf{r}}{2\sigma^2}. \tag{2}$$

De-interleaving is applied to $\mathbf{llr}$ and then the repetition code, of length l, is decoded with:

$$LLR_k = \sum_{i=l(k-1)+1}^{lk} llr_i. \tag{3}$$

Finally, $\mathbf{LLR} = LLR_1, LLR_2, \ldots$ is forwarded to the LDPC decoder and the watermark $\hat{\mathbf{m}}$ is recovered.

5.2.2. *Particular Noise Variance.* MP3 takes into account the human auditory system, then the compression is not constant for every segment along the audio because it is not a stationary signal. Therefore, different amount of distortion is expected in different segments of the audio along time.

Fig. 4 shows the histograms of the soft-information $\mathbf{r}$ for different segments along the audio after MP3 at 128 kbps. In those graphs, we can distinguish between soft-information which belongs to the embedded bits $\bar{m}_i = 1$ with solid lines and $\bar{m}_i = 0$ with dotted lines. Fig. 4a represents the density when all the frequency coefficients from a certain audio are taken into account. That is, the coefficients of 25-28 from all the blocks $\hat{T}_h$ are used. Figs. 4b, 4c and 4d, are densities from small continuous segments of a certain audio file. These segments contain only 320 frequency coefficients, that is, the equivalent to 5 continuous blocks $\hat{T}_h$. The segments were chosen randomly along the audio file.

The audio used to generate Fig. 4 was *A change of season* by Dream Theater sampled at 44.1 kHz with 16 bits. The binary information was embedded in sub-bands 25-28 using DM with only repetition codes of length 4, Δ was set to .01.

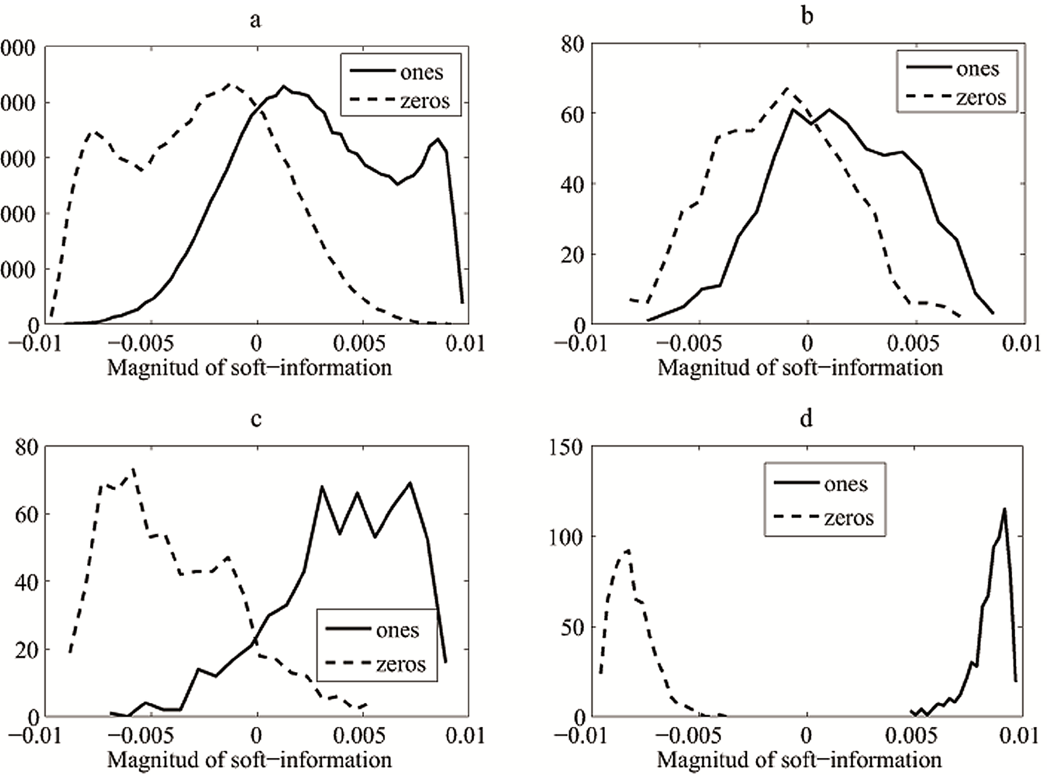

FIGURE 4. Histograms of the soft-information obtained from frequency coefficients after MP3. “a” using the whole audio file. “b”, “c” and “d” are histograms using random segments of 320 frequency coefficients.

Figs. 4b, 4c and 4d show that the noise in the audio file due to MP3 is not constant. For example in Fig. 4d, the difference between ones and zeros is perfectly distinguishable and therefore, a decoding without errors is expected. Other regions like Fig. 4b suffered more distortion and probably many errors will be produced. Nevertheless, if the statistics of the channel noise are computed using the whole audio file, Fig. 4a, the reliability is not good.

Based on the previous experiment, will be more reliable to compute individual noise variances for different segments along the audio file. The decoding proposal in this Section aims to compute independent noise variances along time to obtain a better model of the noise produced by MP3.

The log-likelihood ratio **llr** is computed with:

$$llr_i = \frac{\Delta r_i}{2\varsigma^{\mathbf{2}}}, \tag{4}$$

where $\varsigma^{\mathbf{2}} = \varsigma_1^2, \varsigma_2^2, \ldots \varsigma_p^2$ is estimated by dividing the soft-information $\mathbf{r}$ in p blocks and computing an individual noise variance for each block. For example, if r_i belongs to the block j therefore $llr_i = (\Delta r_i)/(2\varsigma_j^2)$. Then, de-interleaving is applied to $\mathbf{llr}$ and the repetition code is decoded with (3). The $\mathbf{LLR}$ is forwarded to the LDPC decoder and the watermark $\hat{\mathbf{m}}$ is recovered.

6. **Main Results.**

This Section is divided in two parts: decoding for semi-blind schemes and decoding for blind schemes. This division is because an accurate estimation of the noise variance is needed at the decoding part, therefore the difference between them is about how to compute the noise variance. With the semi-blind scheme the decoder has knowledge of the watermarked audio $\hat{\mathbf{X}}$ and it is capable to compute the real noise generated in the audio due to MP3. Semi-blind decoding was developed to show the potential of this proposal.

In blind decoding, an estimation of the noise is needed. Several techniques about channel estimation have been proposed and implemented in practical communications schemes, e.g. “training symbols”, “least-squares” or even more complex techniques which involves “turbo equalization”.

Our aim in this paper is to develop reliable watermarking techniques rather that focus on describing the noise generated by MP3 compression. Therefore, we propose a blind detection using a basic technique with *pilot symbols*, nevertheless our blind decoding produces better performance than, in our knowledge, most of the previous watermarking schemes for MP3 at 64 kbps.

The audio files used in the simulations have the next characteristics: WAVE files, mono and sampled at 44.1 kHz with 16 bits. All the results are an average of simulations based on three audio files, classic: *Egmont Op. 84* (7 minutes), pop: *Billie Jean* by M. Jackson (4 minutes) and rock: *A change of seasons* by Dream Theater (8 minutes). Those audio pieces were chosen because its rich variety of sounds, silent passages and abrupt changes.

The step Δ, which is different for each audio file, was decided according to SNR > 40 dB.

6.1. **Semi-blind decoding.**

In Section 5.2 were defined two decoding strategies: *global noise variance* and *particular noise variance*. In this Section we will show results of them both using $\hat{\mathbf{X}}$ to compute the noise variance.

Let us assume that $\mathbf{Y}$ is the watermarked audio after compression. $\hat{\mathbf{x}}$ and $\mathbf{y}$ are frequency coefficients taken from 25-28 of $\hat{\mathbf{X}}$ and $\mathbf{Y}$ respectively, and they are properly ordered according to time.

For *global noise variance* method, the statistics of channel noise are computed with:

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\big((\hat{x}_i - y_i) - (\overline{\hat{\mathbf{x}} - \mathbf{y}})\big)^2. \tag{5}$$

This unique σ^2 is used to compute (2).

An alternative way, *particular noise variance* method, to compute the noise variance is dividing the frequency coefficients $\hat{\mathbf{x}}$ and $\mathbf{y}$ in blocks $\check{\mathbf{x}}_p$, $\check{\mathbf{y}}_p$ and computing independent statistics about the channel noise:

$$\varsigma_p^2 = \frac{1}{j}\sum_{i=jp-j+1}^{jp}\big((\hat{x}_i - y_i) - (\overline{\check{\mathbf{x}}_p - \check{\mathbf{y}}_p})\big)^2, \tag{6}$$

where j is the number of samples for each block $\check{\mathbf{x}}_p$, $\check{\mathbf{y}}_p$ and ς_p^2 is its noise variance. $\check{\mathbf{x}}_p$ and $\check{\mathbf{y}}_p$ are vectors with elements from $jp - j + 1$ to jp, i.e. $\check{\mathbf{x}}_p = \hat{x}_{jp-j+1}, \ldots, \hat{x}_{jp}$.

The next step is to decide how many samples j are needed to estimate a reliable ς_p^2. Our proposal divides the audio file in blocks $\hat{T}_h$ of 512 samples. From each time-domain block

$\hat{T}_h$, 64 frequency coefficients are obtained considering that only elements from 25-28 are taken. Therefore, the number of elements j is preferable to be a multiple of 64 because in that way the system keeps relation with the division on time domain.

Fig. 5 is a simulation for *particular noise variance* method using different number of samples j to compute the noise variance after compression with MP3. The best performance is achieved when ς_p^2 is computed using $j = 64$ frequency coefficients which is equivalent to estimate an independent noise variance for each block $\hat{T}_h$ of 512 samples. If the number of samples j is increased the performance tends to decrease. Therefore, the best way to characterize the noise produced by MP3 is to compute independent statistics of the channel noise according to the division in time domain.

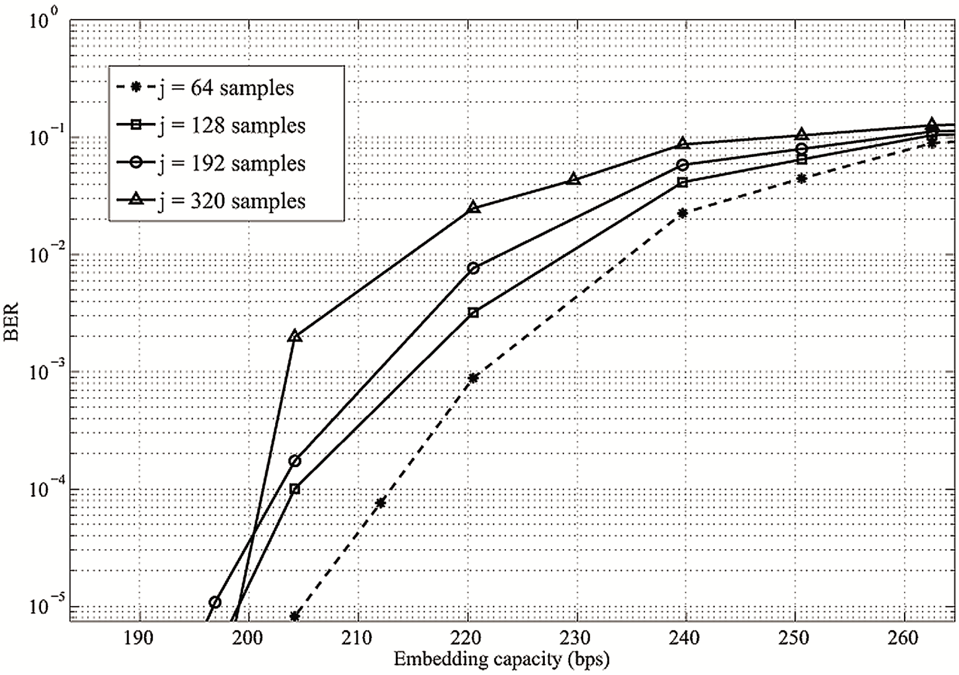

FIGURE 5. *Particular noise variance* scheme using different number of samples j to compute the noise variance ς_p^2.

The proposed methods, *global noise variance* and *particular noise variance* using semi-blind decoding are compared in Fig. 6, together with another two helpful simulations. In "Full-band repetition code", the watermark was embedded in all wavelet sub-bands, from 1 to 32, and only repetition codes were used. The performance is the worst and we were unable to obtain lower BER.

"Full-band" uses a concatenation of outer LDPC code with inner repetition code to encode the watermark. The coded-watermark was embedded in all wavelet sub-bands. With this method low BER was achieved but the performance is still poor. The LDPC decoder uses semi-blind estimation of the noise variance.

"Global noise variance" is the method described in Sec. 5.2.1, the watermark is only embedded in 25-28 achieving an embedding capacity of 23.56 bps.

Finally "Particular noise variance", proposed in Sec. 5.2.2, is the best method. In this case, the watermark was embedded in 25-28 as well. The payload is 204.2 bps with probability of error lower than 10^{-5}. This method embeds 8.2 more bits per second than the proposal in [18].

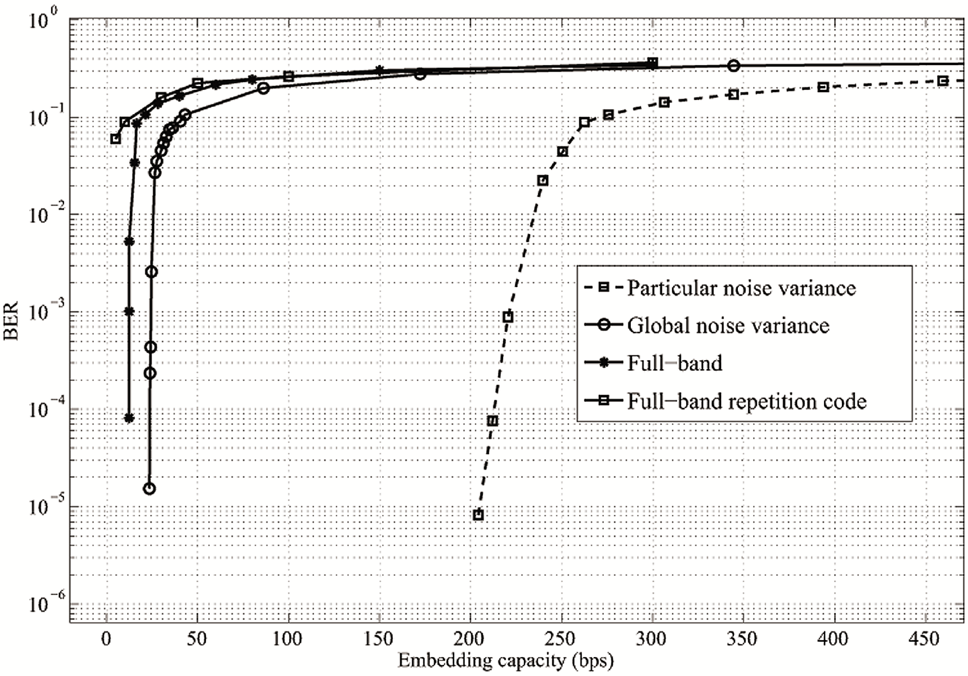

FIGURE 6. Performance of the proposed methods using semi-blind decoding.

6.2. **Blind decoding.** In the previous Section the watermarked audio $\hat{\mathbf{X}}$ is needed to estimate the noise. However blind decoding systems are desirable because audio files, specially in WAVE format, need considerable storage space. Therefore, we propose a blind algorithm using *pilot symbols* only for *particular noise variance* method which was the best method from Sec. 6.1.

In *pilot symbols* approach, a few bits are known for both encoder and decoder. Thus, the decoder is capable to estimate the noise variance with the those bits.

The pilot bits are multiplexed with the encoded-watermark $\bar{\mathbf{m}}$ and embedded in the audio file. We have seen that *particular noise variance* performs better, therefore the noise variance must be computed individually for each block in time domain.

Let us assume that T_h represents a block of 512 audio samples in time domain. After wavelet packet transformation of T_h and gathering only coefficients from 25-28, a block F_h with 64 frequency coefficients is obtained. The coded-watermark $\bar{\mathbf{m}}$ is divided in blocks W_h of $64 - j$ elements. Then, j pilot bits are generated in pseudo-random way using a secret key. The block W_h is multiplexed with the j pilot bits and the result is embedded in F_h using DM. The output of this process will be the watermarked frequency coefficients which include the watermark and the pilot bits, Fig. 7.

The decoder uses a demultiplexer to separate the noisy version of the pilot bits and the watermark. Since the decoder has perfect knowledge of the pilot bits, and estimate ς_h^2 is computed for each block $\hat{F}_h$ and the log-likelihood ratio is obtained with (4).

However there is a trade-off between the number of pilot bits j and the watermark payload. That is, the more pilot bits per block F_h, the better channel estimation that results in a better decoding. But, the more pilot bits the lower redundancy of the watermark and therefore weaker watermarks.

Fig. 8 is a comparison of blind decoding of *particular noise variance* method using a different number of pilot bits j after MP3 at 64 kbps. With a few pilot bits, $j = 5$, per block F_h the performance is the worst. Choosing $j = 20$ produces a good channel estimation, however the watermark is weak and MP3 generates many errors. The best

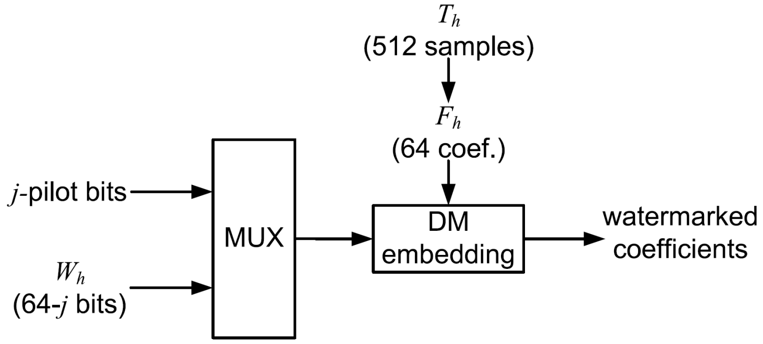

Figure 7. Embedding pilot bits and the coded-watermark.

performance is achieved when $j = 10$ pilot bits per block are used to compute the noise variance ς_h^2. The embedding capacity of this proposal using blind decoding is 155 bps.

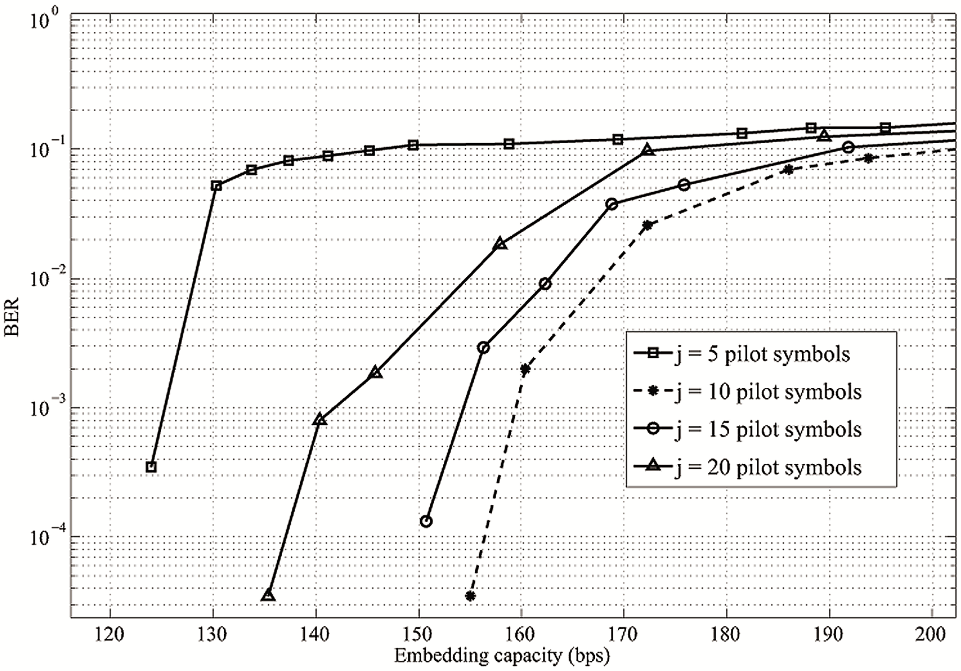

Figure 8. *Particular noise variance* method using different number of pilot bits j per block F_h.

The difference between blind decoding and semi-blind decoding for *particular noise variance* method is shown in Fig. 9, they differ in 49.2 bps. The gap between both methods can be reduced if more advanced techniques to compute the noise variance are implemented in the blind method. Nevertheless our blind proposal achieves similar payload than, in our knowledge, the highest payload algorithm [18] from the literature for blind audio watermarking with attacks of MP3 at 64 kbps.

7. **Increasing The Embedding Capacity with Erasures.** Error correcting codes are capable to correct twice erasures than errors, if the non-erased bits are correct. An erasure means that no information about a certain bit is provided. For example, erasure of the bit $\bar{m}_i$ can be represented with $r_i = 0$.

From Sec. 5.2.2 and also from Fig. 4, we have seen that certain parts of the audio files are susceptible to produce more errors after MP3 compression. If there is a correlation related

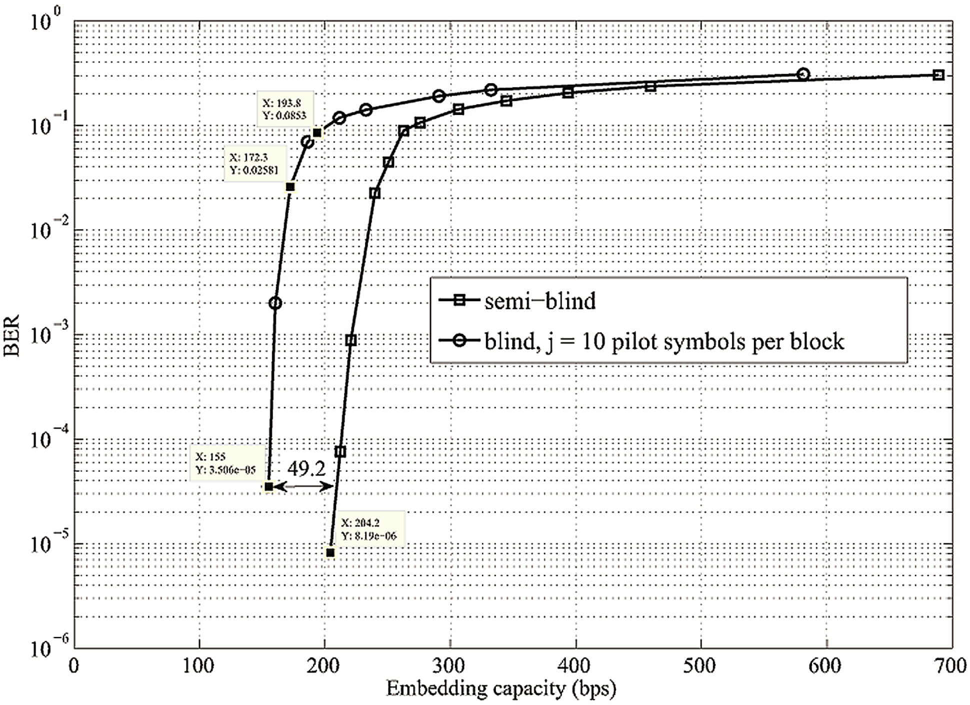

Figure 9. Performance of blind decoding versus semi-blind decoding for *particular noise variance* method.

with the errors, we can define erasures in regions with high density of noise and avoid many errors. Table 1 shows the correlation between the amount of noise in sub-bands 25-28 after MP3 and some characteristics of the audio file. High correlation is obtained between the amount of noise and the average energy of sub-bands 25-28. Based on this correlation, erasures can be defined in places with high average energy of sub-bands 25-28.

Table 1. Correlations related with the amount of noise in wavelet sub-bands 25-28.

Amount of noise from sub-bands 25-28 and	**Value of correlation**
Variance (time)	0.193
Block energy (time)	0.193
Block av. energy (time)	0.190
Frequency (time)	0.651
Av. energy of sub-band 1-4	0.134
Av. energy of sub-band 5-8	0.572
Av. energy of sub-band 9-12	0.678
Av. energy of sub-band 13-16	0.715
Av. energy of sub-band 17-20	0.567
Av. energy of sub-band 21-24	0.609
Av. energy of sub-band 25-28	**0.819**
Av. energy of sub-band 29-32	0.782

The encoding method for this proposal is the same method already explained in this paper. At the decoder, $\mathbf{r}$ is computed as in Sec. 5.2. For each block $\hat{T}_h$, the average energy

e_h of its wavelet sub-bands 25-28 is computed. If r_i comes from a block where $e_h > t$ then $r_i = 0$ is defined as erasure, where t is a threshold. De-interleaving is applied to $\mathbf{r}$ and the rest of the decoding process is exactly as Sec. 5.2.2.

Using erasures, the embedded capacity can be increased by more than 15 bps in comparison with normal decoding, that is without erasures, Fig. 10. The final embedding capacity achieved is 229.7 bps.

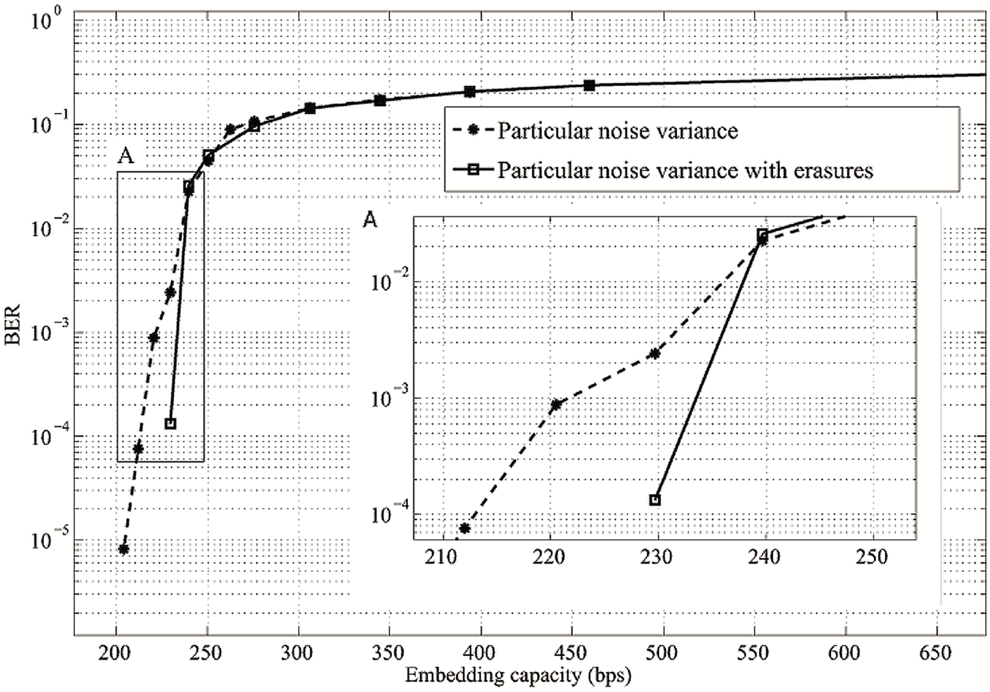

FIGURE 10. Outperform of semi-blind *particular noise variance* scheme using erasures at decoder.

Due to huge dynamic range of audio signals, we have not found a constant threshold t for audio files, even for the same audio file the best threshold varies for different watermark rates. So far, it has been noticed that the best threshold can be found in at most 10 iterations using a exhaustive search in the range from $t = .001$ to $t = .05$.

Even there is a strong noise-correlation, higher than .8, we have noticed that noisy blocks $\hat{F}_h$ only have a number of errors slightly higher than half of the total bits embedded. The reason of this behavior is that DM only does a decoding mistake when $\mod(|\hat{x}_i - y_i|, \Delta/2) > (\Delta/4)$ and it is not strictly related with the noise strength. Therefore, when the whole noisy block $\hat{F}_h$ is defined as erasures, we are discarding erroneous information that represents slightly more than half of the embedded bits in that block, but we are also rejected a considerable amount of correct information.

Better schemes could be developed if there is a correlation between each wavelet coefficient itself and the errors. In that case, there would not be need to define the whole block as erasures, only the most likely erroneous coefficients will be defined as erasures instead.

8. **Comparison with related algorithms.** MP3 is a challenging attack for audio watermarking, and this fact is reflected in Table 2 where 71% percent of the reported algorithms have a payload lower than 100 bps. In some cases, e.g. [6] and [14], the authors claim that their algorithms are robust to compression but they did not report the BER.

Algorithms that overcome the barrier of 100 bps are reported in [10], [11], [17] and [18]. However the achieved payload by [10] and [11] is not with compression of 64 kbps.

Wu *et ál.* [17] proposed an interesting algorithm that achieves 172 bps for compression at 64 kbps with BER = 0.043. Our basic algorithm using pilot symbols achieves the same payload, 172.3 bps with a lower BER = 0.025, see Fig. 9.

The best algorithm, in our knowledge, reported in the literature is the scheme proposed in [18] by Bhat *et ál.* This algorithm has a payload of 196 bps and it is resistant to MP3 at 64 kbps although its BER = .01 is still high, because 1 of every 100 embedded bits is erroneous. Our basic algorithm using pilot symbol has similar performance, achieving 193.8 bps with BER = .08. Using semi-blind decoding our algorithm embeds 8.2 bps more than Bhat's algorithm with an even much lower BER = 8.19×10^{-6}, that is, around 8 erroneous bits of every million embedded bits, refers to Fig. 9.

Moreover our best result, Fig. 10, has a benefit of 33.7 bps over Bhat's algorithm. In summary, our algorithms obtain a payload of 155 bps for the basic algorithm with pilot symbols, in Sec. 6.2; 204.2 bps for the semi-blind algorithm of Sec. 6.1 and 229.7 bps for the algorithm with erasures at decoder, described in Sec. 7.

Since the aim of this paper is not the channel estimation and for the sake of simplicity, we introduced only a basic blind decoding which uses pilot symbols. However, our semi-blind decoding algorithm could be easily converted to blind using a good channel estimation technique, because $\hat{\mathbf{X}}$ is only required to compute the noise variance.

Table 2. BER and payload comparison of related algorithms against MP3 compression.

Algorithm		**MP3 Quality [kbps]**	**BER**	**Payload [bps]**
Cvejic	[3], 2003	32	.0028	27.1
In-kwon	[4], 2003	96	.0020	10
Wang	[5], 2004	128	.0571	10.72
Wu	[17], 2005	64	.0434	172
Chang	[6], 2006	56	YES	86
Li	[7], 2006	32	.0156	4.26
Xiang	[8], 2007	128	.1500	3
Xiang	[9], 2008	64	.1750	2
Erçelebi	[10], 2008	128	.4900	170
Deshpande	[11], 2008	96	.0025	220
Fan	[12], 2009	48	.0347	86
Wang	[13], 2009	64	.0100	Not reported
Megías	[14], 2010	96	YES	30.09
Bhat	[18], 2010	64	.0100	196
Ours		64	1.323×10^{-4}	229.7

9. **Audio Quality Test.** Quality of the watermarked signal is important because a signal with bad quality loses its commercial value. This aspect becomes even more important in audio files because the human auditory system is more sensible to perceive changes than other senses, e.g. the sight.

All the simulation presented in this paper produced watermarked audio with an average SNR = 44.1 dB and variance of .02. However, SNR is not the most suitable metric for measuring audible distortion. The quality of the watermarked audio is also measured with the *ITU-R BS.1116 standard* [29] which is a subjective evaluation of small impairments of high-quality perceptual audio codecs. The highest grade 5.0 refers to the best audio quality and the lowest grade 1.0 means the poorest audio quality. The test was applied to 23 persons with three different audio files: classic, pop and rock music. The results are shown in Table 3, in all of them the quality is higher than 4.0.

Table 3. Subjective evaluation of audio quality based on *ITU-R BS.1116 standard.*

Audio	**Score**
Classic: *Egmont Op. 84*	4.14
Pop: *Billie Jean* by M. Jackson	4.78
Rock: *A change of seasons* by Dream Theater	4.28

10. **Conclusion.** High payload algorithms for audio watermarking resilient to MP3 compression have been proposed. Our algorithms embed a binary coded-watermark inside the frequency of audio files using DM. The coded-watermark is generated with a concatenation of LDPC codes with repetition codes.

Effects of MP3 on wavelet coefficients were analyzed. Coefficients from sub-bands 25 to 28 were found to suffer less distortion considering that the audio is decomposed with wavelet packet until the fifth level. Sub-bands 25 to 28 belong to middle frequencies of the audio, equivalent to frequencies from 11 kHz until 13.7 kHz.

The statistics of the channel noise were computed using independent estimations along the audio file. This produced a more accurate variance of the channel noise.

Reflexions from the previous paragraphs allow us to propose an algorithm called *particular noise variance* which has an embedding capacity higher than all previous known proposal resilient to MP3 at 64 kbps. This algorithm is capable to use semi-blind and blind decoding. With semi-blind decoding the algorithm obtained a benefit of 8.2 bps over the algorithm with the highest payload reported in literature [18] for MP3 of 64 kbps and with blind decoding, our algorithm has similar performance to [18].

A strong correlation between the amount of noise and the average energy from sub-bands 25-28 was found. This correlation was used to define erasures on places which are likely to be noisy. This idea increased the embedding capacity of the *particular noise variance* method by more than 15 bps. The final achieved payload was of 229.7 bps, which overperforms the algorithm in [18] by 33.7 bps.

Finally, the distortion in the watermarked audio was evaluated. The quality of audio was measured with two methods. The first one shows SNR higher than 40 dB. The second one was a subjective evaluation with more than 20 persons. The results show a score higher than 4 of 5 possible for the watermarked audio, where 5 represents the best audio quality.

REFERENCES

[1] N. Aoki, Improvement of a band extension technique for G.711 telephony speech by using steganography, *Proc. of the 5th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, pp. 487–490, 2009.

[2] L. Boney, A. H. Tewfik, and K. N. Hamdy, Digital watermarks for audio signals, *Proc. of the 3rd IEEE Int. Conf. on Multimedia Computing and Systems*, pp. 473-480, 1996.

[3] N. Cvejic and T. Seppnen, Spread spectrum audio watermarking using frequency hopping and attack characterization, *Signal Processing*, vol. 84, no. 1, pp. 207-213, 2003.

[4] Y. In-Kwon and J. K. Hyoung, Modified patchwork algorithm: a novel audio watermarking scheme, *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 4, pp. 381-386, 2003.

[5] R. Wang, D. Xu, J. Chen, and C. Du, Digital audio watermarking algorithm based on linear predictive coding in wavelet domain, *Proc. of Int. Conf. on Signal Processing*,vol. 3, pp. 2393-2396, 2004.

[6] C-Y. Chang, W-C. Shen, and H-J. Wang, Using Counter-propagation Neural Network for Robust Digital Audio Watermarking in DWT Domain, *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 2, pp. 1214-1219, 2006.

[7] W. Li, X. Xue, and P. Lu, Localized audio watermarking technique robust against time-scale modification, *IEEE Trans. Multimedia*, vol. 8, no. 1, pp. 60- 69, 2006.

[8] S. Xiang and J. Huang, Histogram-Based Audio Watermarking Against Time-Scale Modification and Cropping Attacks, *IEEE Trans. Multimedia*, vol. 9, no. 7, pp. 1357-1372, 2007.

[9] S. Xiang, H. J. Kim, and J. Huang, Audio watermarking robust against time-scale modification and MP3 compression, *Signal Processing*, vol. 88, no. 10, pp. 2372-2387, 2008.

[10] E. Erçelebi and L. Batakçı, Audio watermarking scheme based on embedding strategy in low frequency components with a binary image, *Digit. Signal Process*, vol. 19, no. 2, pp. 265–277, 2009.

[11] A. Deshpande and K. M. M. Prabhu, A substitution-by-interpolation algorithm for watermarking audio, *Signal Processing*, vol. 89, no. 2, pp. 218-225, 2008.

[12] M. Fan, H. Wang, Chaos-based discrete fractional Sine transform domain audio watermarking scheme, *Computers and Electrical Engineering*, vol. 35, no. 3, pp. 506-516, 2009.

[13] X. Y. Wang, P. P. Niu, and H. Y. Yang, A robust digital audio watermarking based on statistics characteristics, *Pattern Recognition*, vol. 42, no. 11, pp. 3057-3064, 2009.

[14] D. Megias, J. Serra-Ruiz, and M. Fallahpour, Efficient self-synchronised blind audio watermarking system based on time domain and FFT amplitude modification, *Signal Processing*, In Press, 2010.

[15] B. Chen and G.W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, *IEEE Trans. Information Theory*, vol. 47 no. 4, pp. 1423-1443, 2001.

[16] F. Pérez-González, C. Mosquera, M. Barni, and A. Abrardo, Rational dither modulation: A high-rate data-hiding method invariant to gain attacks, *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3960–3975, 2005.

[17] S. Wu, J. Huang, D. Huang, and Y.Q. Shi, Efficiently self-synchronized audio watermarking for assured audio data transmission, *IEEE Trans. Broadcasting*, vol. 51, no. 1, pp. 69-76, 2005.

[18] V. Bhat, I. Sengupta, and A. Das, An audio watermarking scheme using singular value decomposition and dither-modulation quantization, *Multimedia Tools and Applications*, 2010.

[19] C. Desset, B. Macq and L. Vandendorpe, Block error-correcting codes for systems with a very high BER: Theoretical analysis and application to the protection of watermarks, *Signal Processing and Image Communications.*, vol. 17, no. 5 pp. 409–421, 2002.

[20] L. Gu, Y. Fang and J. Huang, Revaluation of error correcting coding in watermarking channel*, *Lecture Notes in Computer Science*, vol. 3810, pp. 274-287, 2005.

[21] R. Gallager, Low-density parity-check codes, *IRE Trans. Information Theory*, vol. 8, no. 10, pp. 21–28, 1962.

[22] N. Cvejic, D. Tujkovic, and T. Seppanen, Increasing robustness of an audio watermark using turbo codes, *Proc. of Int. Conf. on Multimedia and Expo*, Baltimore, USA, pp. 217-220, 2003.

[23] R. Martínez-Noriega, M. Nakano, and K. Yamaguchi, On the channel characteristic of dither modulation data hiding for MP3 compression, *Proc. of 5th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing*, Kyoto, Japan, pp. 90–93, 2009.

[24] A. Bastug and G. Sankur, Improving the payload of watermarking channels via LDPC coding, *IEEE Signal Letters*, vol. 11, no. 2, pp. 90–92, 2004.

[25] J.J. Garcia-Hernandez, M. Nakano, and H. Perez-Meana, Data hiding in audio signal using rational dither modulation, *IEICE Electronics Express*, vol. 5, no. 7, pp. 217-222, 2008.

[26] M. Fallahpour and D. Megias, High capacity audio watermarking using FFT amplitude interpolation, *IEICE Electronics Express*, vol. 6, no. 14, pp. 1057-1063, 2009.

[27] M. Fallahpour and D. Megias, High capacity audio watermarking using the high frequency band of the wavelet domain, *Multimedia tools and Applications*, 2010.

[28] D. J. C. MacKay, Good error-correcting codes based on very sparse matrices, *IEEE Trans. Information Theory*, vol. 45, no. 2, pp. 399-431, 1999.

[29] M. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and applications of digital watermarking and content protection*, Artech House, 2003.