Resynchronization methods for audio watermarking

Leandro de C. T. Gomes*
InfoCom-Crip5, Université René Descartes – Paris, FRANCE
tgomes@math-info.univ-paris5.fr
http://www.math-info.univ-paris5.fr/crip5/infocom/

Emilia Gómez
Audiovisual Institute, Pompeu Fabra University – Barcelona, SPAIN
emilia.gomez@iua.upf.es – http://www.iua.upf.es/mtg

Nicolas Moreau
TSI, École Nationale Supérieure des Télécommunications – Paris, FRANCE
moreau@tsi.enst.fr – http://www.tsi.enst.fr/

Abstract

Depending on the application, audio watermarking systems must be robust to piracy attacks. Desynchronization attacks, aimed at preventing the detector from correctly locating the information contained in the watermark, are particularly difficult to neutralize. In this paper, we introduce resynchronization methods for audio watermarking based on the use of training sequences. These methods reverse the effect of a large class of desynchronization attacks. Simulation results confirm the efficiency of the proposed methods.

1 Introduction

1.1 Watermarking: definition and applications

Digital signals can be copied and distributed easily and with no degradation, creating an environment that is propitious to piracy. Audio watermarking has been proposed as a solution to this problem. It consists in embedding a mark (the watermark) into an audio signal. Watermarking-compliant devices are supposed to check for the presence of this mark and act according to the information contained therein. Watermarking can also be used to identify the source of illicit copies (fingerprinting) by inserting a unique serial number in each copy.

Besides copyright protection, many other applications have been proposed for audio watermarking [1, 2], such as:

- verification of the integrity of an audio signal

*The author would like to thank CNPq-Brazil for financial support.

- storage of additional information for the end-user (e.g. the lyrics of a song)
- identification of songs or commercials aired by a radio or TV station (broadcast monitoring and verification)
- automatic measurement of audience (by using different watermarks for each radio or TV station).

Depending on the application, an audio watermarking system should comply with certain requirements [1]. Some of the most common requirements are listed below:

- inaudibility: the watermark should not result in perceptible distortion in the audio signal
- robustness: the watermark should be robust to modifications applied to the audio signal, as long as sound quality is not severely degraded
- reliability: the system should present a high rate of correct detection and a low rate of false alarms
- low complexity: for real-time applications, watermark insertion and/or detection should not be excessively time-consuming
- low cost in bit rate: for compressed audio, the watermark should not excessively increase bit rate.

Robustness is generally a major concern, as discussed in section 3, in particular for copyright protection applications.

1.2 Generic watermarking scheme

Figure 1 shows a generic watermarking scheme. Key 1 is used to generate the watermark,

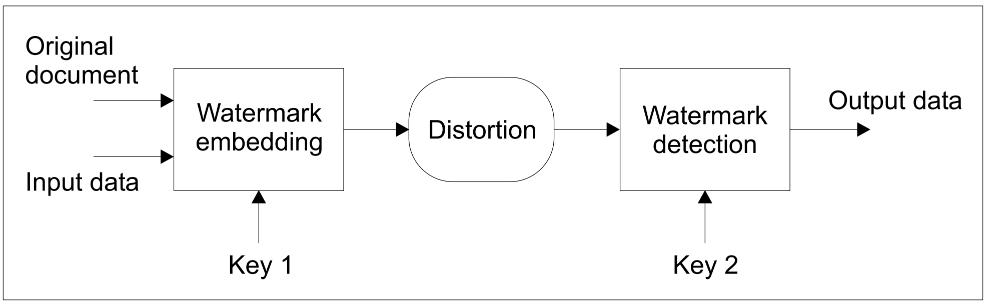

Figure 1: Generic watermarking scheme.

while key 2 is needed to detect it. If these keys are identical, the watermark scheme is symmetric; otherwise, the scheme is asymmetric and key 2 should provide different functionality than key 1. For example, key 1 can be a private key, giving full access to the watermark, and key 2 a public key, allowing the user to retrieve (part of) the information contained in the watermark but not allowing the suppression of the watermark from the audio signal.

Examples of symmetric watermarking schemes are presented in [3, 4, 5]. Examples of asymmetric schemes can be found in [6, 7, 8].

2 Watermarking technique

Watermarking can be viewed as a noisy communication channel [5]: the watermark is the transmitted information and the audio signal (along with distortions imposed on the watermarked signal) is the noise, which is several times stronger than the watermark (due to the inaudibility condition). This approach is illustrated in Figure 2.

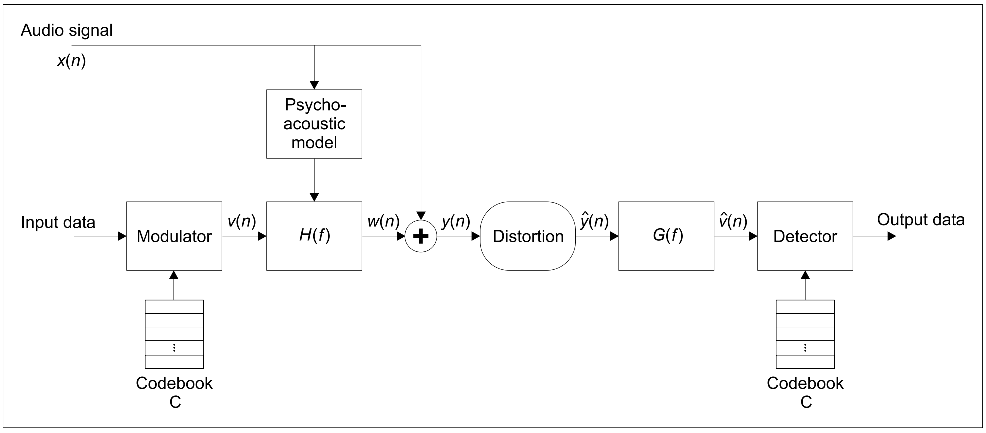

Figure 2: Watermarking as a communication channel.

As in a standard communication channel, information is represented by symbols. If the total number of symbols is K and the symbols are equiprobable, each symbol carries $\log_2(K)$ bits of information. Let C be a codebook associating N-length vectors $\mathbf{u_k} = [u_k(0) \cdots u_k(N-1)]$ to the symbols. These vectors are normally distributed and orthogonal. The modulator receives a sequence of input symbols $\mathbf{s} = [s_0 \cdots s_{M-1}]$ and produces a signal $v(n)$ by concatenating the corresponding vectors:

$$v(mN+n) = u_{s_m}(n).$$

To guarantee inaudibility of the watermark, $v(n)$ is frequency-shaped to fit a masking threshold obtained from a psychoacoustic model [9, 10]. This task is accomplished by the filter $H(f)$, whose amplitude response follows the masking threshold. The resulting signal $w(n)$ is added to the audio signal $x(n)$, producing the watermarked signal $y(n)$.

The observed watermarked signal $\hat{y}(n)$ is first filtered by $G(f)$, a Wiener filter estimated from $\hat{y}(n)$ and intended to increase the watermark-to-signal ratio. Its output, $\hat{v}(n)$, is an estimation of $v(n)$. The detector receives $\hat{v}(n)$ and, based on correlation measures, produces a sequence of detected symbols.

3 Desynchronization

Depending on the application, the watermark must present a certain degree of resistance to distortions. For most applications, resistance to licit operations (e.g. MPEG encoding/decoding, filtering, resampling) is required. In addition, copyright protection applications require the system to resist malicious attacks aimed at rendering the watermark undetectable (e.g. addition of noise, cutting/pasting, filtering).

If commercial value is to be preserved, a pirate trying to prevent watermark detection has to respect the inaudibility constraint. This imposes a limit on the amount of noise that can be added to the signal, as well as on non-additive distortions such as cutting/pasting.

Desynchronization attacks are particularly difficult to neutralize. In order to retrieve the watermark, the detector must be synchronous to the transmitter, i.e. the starting and finishing times of each symbol must be known. This is necessary because the correlation between the embedded signal and the corresponding vector in the codebook falls rapidly as the analysis window is shifted from the correct position.

Many signal processing operations can result in desynchronization. For example, an encoding/decoding process can introduce a delay at the beginning of the signal. A pirate can also delete or add samples to the signal. Experiences have shown that, for audio signals sampled at 32 kHz, up to one sample in 2,500 can be randomly deleted or added with no perceptible distortion to the ordinary listener [11]. By choosing stationary regions of the signal, many more samples can be imperceptibly erased or inserted.

Another attack that causes desynchronization consists in modifying the length of the signal (time warp). If this change in length is slight enough, it will be imperceptible to the listener. By using time-stretching techniques (i.e. modifying the length while keeping the pitch constant), the pirate will be able to impose stronger variations in length without severely degrading signal quality.

Finally, the pirate can exploit the lack of sensibility of the human ear to phase modification (as long as phase continuity is preserved) by passing the watermarked signal through an all-pass filter. Although this attack does not change the starting and finishing times of a symbol, it will reduce the correlation between the signal and the corresponding vector in the codebook, thus inducing detection errors.

In this study, we have focused on desynchronization attacks that cause the location of the symbols to be unknown but do not significantly modify their length. Resistance to all-pass filtering and MPEG compression/decompression is also analyzed.

4 Resynchronization methods

4.1 Training sequences: basic idea

One of the most common synchronization techniques in digital communications consists in using training sequences, i.e. sequences of data that are known to both the transmitter and the receiver. The training sequences are interposed between useful data, allowing the detector to retrieve synchronization whenever such a sequence is found. The same idea can be applied to watermarking: portions of the watermark can be composed of known successions of symbols (the training sequences). When synchronization is lost, detection is performed for each possible symbol location (by means of a sliding window) until a training sequence is found. This method enables synchronization to be tracked along the audio signal [11, 12].

When samples are deleted or added to the watermarked signal, the peak of the intercorrelation function between the signal and the corresponding vectors in the codebook is shifted accordingly, as shown in Figure 3. From these correlation measures, the actual location of the data sequence can be estimated. The training sequence must be short enough to allow the resynchronization process to be completed in a reasonable amount of time, but it must be long enough for the correlation computations to be meaningful.

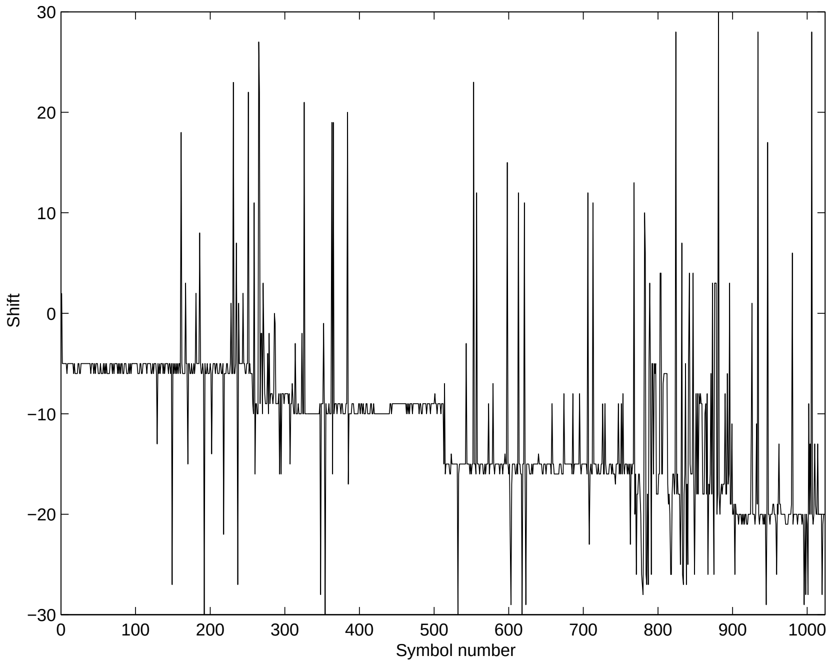

Figure 3: Shifts of the sliding window corresponding to the maximal correlation for each symbol in the watermarked signal. In this example, 5 samples were deleted after each block of 256 symbols.

This approach presents two major drawbacks. During the periods corresponding to training sequences, the watermark does not carry useful information. As samples are deleted or added more frequently by the pirate, successive training sequences have to be placed closer to each other in the watermarked signal in order to reverse the attack, which results in a progressive reduction of the watermark data rate. Furthermore, if the training sequences themselves are attacked by the pirate, resynchronization efficiency might be severely reduced.

4.2 Spread training sequences

In this section, we present two resynchronization methods that overcome the difficulties mentioned in the previous paragraph. The idea consists in spreading the training sequence over the watermarked signal, avoiding preferential regions where piracy attacks would be more effective and enabling the detector to track synchronization continuously. As the training sequence is present all the time, it must coexist with useful data.

4.2.1 Independent synchronization watermark

The training sequence can be spread in time by means of a watermark $\dot{w}(n)$ that is used for the sole purpose of synchronization. Another watermark, $\ddot{w}(n)$, carries useful information. In order to avoid interference between the watermarks, they are constructed on the basis of two orthogonal codebooks. The total watermark $w(n)$ (whose power spectral density must be situated under the masking threshold to ensure inaudibility) is obtained by adding $\dot{w}(n)$ and $\ddot{w}(n)$. A desynchronization attack will have exactly the same effect on both watermarks, as they are superposed; thus, if the detector is able to retrieve synchronization for $\dot{w}(n)$, the same will be true for $\ddot{w}(n)$. The resulting watermarking scheme is shown in Figure 4.

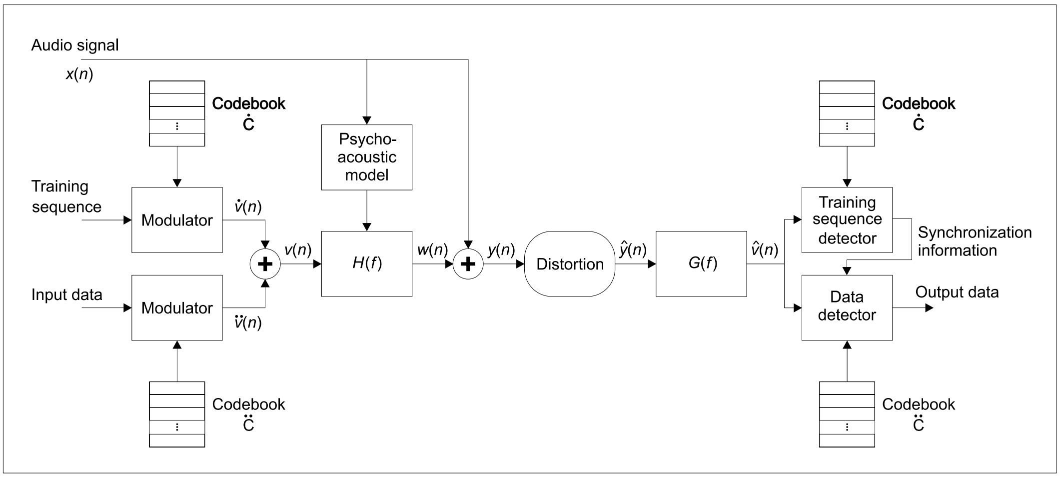

Figure 4: Watermarking system with resynchronization through an additional watermark.

Let $\dot{\mathbf{C}}$ be the codebook used to construct the synchronization watermark $\dot{w}(n)$. It contains $\dot{K}$ vectors $\dot{\mathbf{u}}_k = [\dot{u}_k(0) \cdots \dot{u}_k(N-1)]$ $(k \in [0, \dot{K}-1])$ associated with $\dot{K}$ symbols. The training sequence $\mathbf{z} = [z_0 \cdots z_{M-1}]$ is obtained according to the rule

$$z_m = m \bmod \dot{K}$$

where z_m is the m-th symbol in the training sequence $(m \in [0, M-1])$. The resulting sequence is completely known to the detector.

Now, let $\ddot{\mathbf{C}}$ be the codebook used to construct $\ddot{w}(n)$. It contains $\ddot{K}$ vectors $\ddot{\mathbf{u}}_k = [\ddot{u}_k(0) \cdots \ddot{u}_k(N-1)]$ $(k \in [0, \ddot{K}-1])$ associated with $\ddot{K}$ symbols. The sequence of symbols $\mathbf{s} = [s_0 \cdots s_{M-1}]$ represents the actual information to be embedded into the audio signal. The watermark $w(n) = \dot{w}(n) + \ddot{w}(n)$ is constructed by successively concatenating the vectors associated with the symbols in sequences $\mathbf{z}$ and $\mathbf{s}$, plus a filtering operation to ensure inaudibility (see section 2):

$$\begin{aligned} w(mN+n) &= \dot{w}(mN+n) + \ddot{w}(mN+n) \\ &= [\dot{v}(mN+n) + \ddot{v}(mN+n)] * h(n) \\ &= [\dot{u}_{z_m}(n) + \ddot{u}_{s_m}(n)] * h(n) \end{aligned}$$

where n corresponds to time within the current analysis window $(n \in [0, N-1])$ and $h(n)$ to the impulse response of a filter synthesized from the masking threshold.

In the detection phase, a sliding window is used to calculate N correlation measures for each of the M symbols in watermark $\dot{w}(n)$ and for each of the $\dot{K}$ vectors in codebook $\dot{C}$:

$$\dot{r}(\lambda, k, m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN+n+\lambda)\dot{u}_k(n) \right|$$

where $\lambda \in [-\Lambda, \Lambda - 1]$ is the shift of the sliding window ($\Lambda = N/2$ for N even) and $\hat{v}(n)$ is the reconstructed watermark (as described in section 2). Then, through maximization in k, we construct two matrices $\mathbf{A} = \{\alpha_{\lambda,m}\}$ and $\mathbf{B} = \{\beta_{\lambda,m}\}$ whose rows correspond to the shifts λ and whose columns correspond to the position m of the symbols in the sequence:

$$\alpha_{\lambda,m} = \max_k \dot{r}(\lambda, k, m)$$

$$\beta_{\lambda,m} = \arg\max_k \dot{r}(\lambda, k, m).$$

Thus, $\mathbf{A}$ contains the highest correlation measures for each shift and each position in the sequence of symbols and $\mathbf{B}$ contains the corresponding symbols from codebook $\dot{\mathbf{C}}$.

As will be explained in section 4.3, a dynamic programming algorithm is used to find an optimal path in matrices $\mathbf{A}$ and $\mathbf{B}$ from the first column ($m = 0$) to the last one ($m = M-1$). The optimization takes into account the expected order of symbols (i.e. the training sequence) and the correlation measures. This results in a set of M chosen values for λ, $[\hat{\lambda}_0 \cdots \hat{\lambda}_{M-1}]$, one for each position in the training sequence.

Detection is then performed for watermark $\ddot{w}(n)$. Correlation measures are calculated for each symbol in sequence $\mathbf{s}$, using the set of shift values that has just been obtained:

$$\ddot{r}(k, m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN + n + \hat{\lambda}_m) \ddot{u}_k(n) \right|$$

and the sequence of detected symbols $\hat{\mathbf{s}}$ is extracted by choosing, for each m, the symbol in codebook $\ddot{\mathbf{C}}$ that corresponds to the maximum correlation measure:

$$\hat{s}_m = \arg\max_k \ddot{r}(k, m)$$

where $\hat{s}_m$ stands for the m-th detected symbol.

4.2.2 Sequence of codebooks

Another method for spreading the training sequence in time consists in using several orthogonal codebooks for coding information. These codebooks are used consecutively, creating a sequence of codebooks that plays the role of a training sequence.

Let us define P orthogonal codebooks $\mathbf{C}_p$ ($p \in [0 \cdots P-1]$). Each of these codebooks contains K vectors $\mathbf{u}_{p,k} = [u_{p,k}(0) \cdots u_{p,k}(N-1)]$ ($k \in [0 \cdots K-1]$) associated with K symbols. Corresponding symbols in the codebooks (i.e. symbols corresponding to the same index k) are equivalent in the sense that they represent the same information, but they are associated with different vectors. Therefore, the detector is able to know from which codebook each detected symbol comes. The resulting scheme is illustrated in Figure 5.

The sequence of codebooks $\mathbf{z} = [z_0 \cdots z_{M-1}]$ (i.e. the training sequence) is obtained according to the following rule:

$$z_m = m \bmod P$$

where z_m is the m-th codebook in the sequence. The resulting sequence is to be precisely retrieved when detection is synchronized.

The information to be embedded into the audio signal is represented by the sequence of symbols $\mathbf{s} = [s_0 \cdots s_{M-1}]$. The watermark $w(n)$ is constructed by successively concatenating

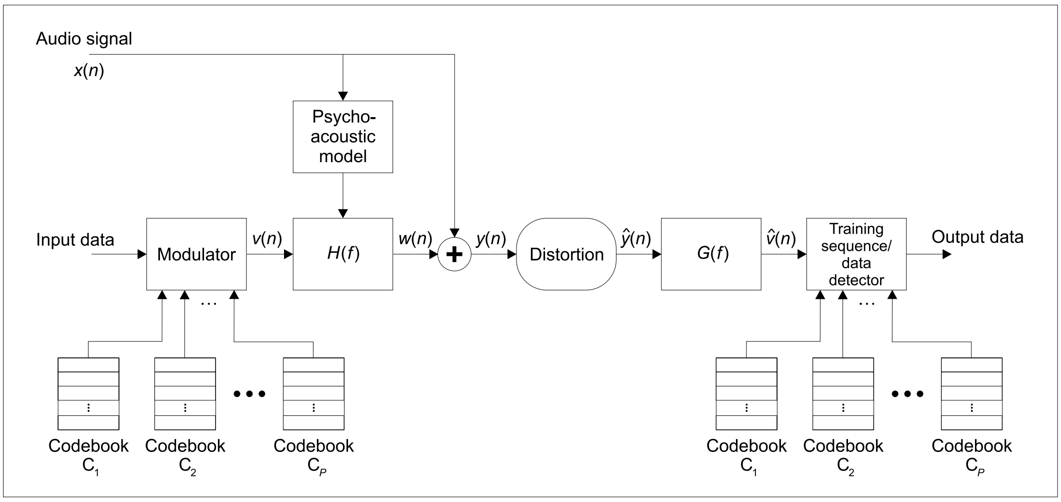

Figure 5: Watermarking system with resynchronization through multiple codebooks.

the vectors associated with the symbols in this sequence, in accordance with the sequence of codebooks, plus a filtering operation to ensure inaudibility:

$$w(mN + n) = u_{z_m, s_m}(n) * h(n)$$

where n corresponds to time within the current analysis window ($n \in [0, N-1]$) and $h(n)$ to the impulse response of a filter synthesized from the masking threshold.

During detection, a sliding window is used to calculate N correlation measures for each of the M symbols in the watermark and for all K vectors in each of the P codebooks C_p:

$$r(\lambda, p, k, m) = \left| \sum_{n=0}^{N-1} \hat{v}(mN + n + \lambda) u_{p,k}(n) \right|$$

where λ and $\hat{v}(n)$ are defined as in the previous section. Then, through maximization in p and k, we construct three matrices $\mathbf{A} = \{\alpha_{\lambda,m}\}$, $\mathbf{B} = \{\beta_{\lambda,m}\}$ and $\mathbf{\Gamma} = \{\gamma_{\lambda,m}\}$ whose rows correspond to the shifts λ and whose columns correspond to the position m of the symbols in the sequence:

$$\alpha_{\lambda,m} = \max_{p,k} r(\lambda, p, k, m)$$

$$\beta_{\lambda,m} = \arg_p \max_{p,k} r(\lambda, p, k, m)$$

$$\gamma_{\lambda,m} = \arg_k \max_{p,k} r(\lambda, p, k, m).$$

Thus, $\mathbf{A}$ contains the highest correlation measures for each shift and each position in the sequence of symbols, $\mathbf{B}$ contains the corresponding codebooks, and $\mathbf{\Gamma}$ contains the corresponding detected symbols.

A dynamic programming algorithm is used to find an optimal path in matrices $\mathbf{A}$ and $\mathbf{B}$, as will be explained in the next section. The optimization takes into account the expected order of codebooks (i.e. the training sequence) and the correlation measures. The sequence of detected symbols $\hat{\mathbf{s}}$ is then obtained straightforwardly by following this optimal path in matrix $\mathbf{\Gamma}$.

4.3 Dynamic programming optimization

In order to determine the shifts of the sliding window that best correspond to the actual symbol locations, a dynamic programming algorithm is employed. The optimization procedure minimizes a cost function calculated in terms of matrices $\mathbf{A} = \{\alpha_{\lambda,m}\}$ and $\mathbf{B} = \{\beta_{\lambda,m}\}$ (defined in sections 4.2.1 and 4.2.2). This results in a set of M chosen values for the shift λ, $[\lambda_0 \cdots \lambda_{M-1}]$, defining a path in matrices $\mathbf{A}$ and $\mathbf{B}$ from which the sequence of detected symbols $\hat{\mathbf{s}}$ can be obtained.

The cost $c(\lambda, \lambda', m)$ for passing from node $[\lambda', m-1]$ to node $[\lambda, m]$ is composed of three terms:

$$c(\lambda, \lambda', m) = c_1(\lambda, \lambda', m) + c_2(\lambda, \lambda', m) + c_3(\lambda, m).$$

The first one, responsible for enforcing observance of the training sequence, is defined as

$$c_1(\lambda, \lambda', m) = \begin{cases} \epsilon(\beta_{\lambda,m} - \beta_{\lambda',m-1} - 1) & \text{if } \beta_{\lambda,m} \geq \beta_{\lambda',m-1}; \\ \epsilon(\beta_{\lambda,m} - \beta_{\lambda',m-1} - 1 + P) & \text{otherwise} \end{cases}$$

where ϵ is a positive constant. If the training sequence is precisely respected, this cost is null; otherwise, the cost is proportional to the leap in the training sequence. This definition is justified by the fact that, due to the inaudibility constraint, the pirate is not likely to erase or add long segments to the watermarked signal.

The second term penalizes changes in the shift λ when passing from node $[\lambda', m-1]$ to node $[\lambda, m]$, which is intended to keep the optimal path in the same row when the training sequence is respected (i.e. in the absence of desynchronization):

$$c_2(\lambda, \lambda', m) = \eta_{m-1}(\lambda - \lambda')^2$$

where the square causes the penalty to increase rapidly as λ moves away from λ' (which is also justified by the fact that long segments are not likely to be erased or added to the watermarked signal). The factor η_m is defined as

$$\eta_m = \begin{cases} \eta_{m-1} + \kappa_1 & \text{if } \lambda_m \neq \lambda_{m-1}; \\ \max(\eta_{m-1} - \kappa_2, \eta_0) & \text{otherwise} \end{cases}$$

with κ_1 and κ_2 being positive constants (generally $\kappa_1 > \kappa_2$), λ_m the row number corresponding to column m on the current path, and η_0 being initialized at a positive value. This definition is intended to avoid zigzag paths, as η_m will tend to grow in such a situation.

The third term in the cost definition is related to the correlation measures in matrix $\mathbf{A}$:

$$c_3(\lambda, m) = \rho \left(1 - \frac{\alpha_{\lambda,m}}{\max_{\tilde{\lambda}} \alpha_{\tilde{\lambda},m}} \right)$$

where ρ is a positive constant. The expression in parentheses takes values between 0 (when the shift λ corresponds to the highest correlation) and 1 (when the correlation for shift λ is null). This definition penalizes shifts λ leading to low correlation measures.

Let us define the accumulated cost $C(\lambda, m)$ as the minimal cost for reaching node $[\lambda, m]$ from a node in the first column ($m = 0$). This cost is initialized at 0 for $m = 0$ and all λ. The optimization algorithm is described as follows:

For $m = 1 \cdots M-1$
For $\lambda = -\Lambda \cdots \Lambda - 1$
$\bar{\lambda} = \arg\min_{\lambda'}[C(\lambda', m-1) + c(\lambda, \lambda', m)]$
$C(\lambda, m) = C(\bar{\lambda}, m-1) + c(\lambda, \bar{\lambda}, m)$
$I(\lambda, m) = \bar{\lambda}$
$\hat{\lambda}_{M-1} = \arg\min_{\tilde{\lambda}}[C(\tilde{\lambda}, M-1)]$
For $m = M-2 \cdots 0$
$\hat{\lambda}_m = I(\lambda_{m+1}, m+1)$.

This results in the set of shifts $[\hat{\lambda}_0 \cdots \hat{\lambda}_{M-1}]$ corresponding to the optimal path. The detected symbols are then obtained as described in sections 4.2.1 and 4.2.2.

5 Simulations

5.1 Experimental conditions

Four signals (4.8 seconds each, single channel, sample rate of 32 kHz, 16 bits per sample) were used during the tests: "svega" ("Tom's diner", a cappella version, by Suzanne Vega), "violin" (a piece of violin), "baron" (a piece of Caribbean music by Baron) and "queen" (a piece of pop music). After watermarking, each signal was submitted to the following operations:

- Random suppression/addition of one sample in 2,500
- All-pass filtering (figure 6)

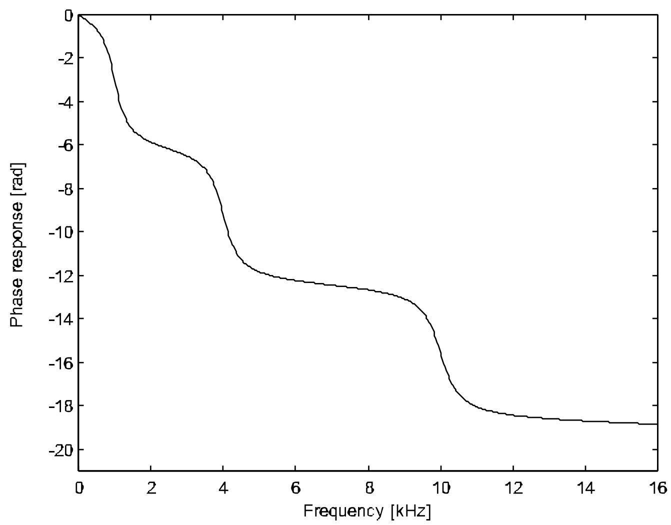

Figure 6: Phase response of the all-pass filter used in the tests.

Windows of length $N = 512$ were used, with a maximum shift $\Lambda = 256$ for the sliding window. Processing was performed for groups of $M = 50$ windows. A bit rate of 125 bits/s was used.

The following values were used for the constants in the optimization procedure: $\epsilon = \Lambda$, $\eta_0 = 1$, $\kappa_1 = 5$, $\kappa_2 = 1$ and $\rho = 10\Lambda$. These parameters have been chosen experimentally.

Masking thresholds were obtained from the MPEG-2 psychoacoustic model number 1. As shown in the next sections, signal-to-watermarking ratios were always above 15 dB, which is generally the limit of audibility.

In the absence of attacks, low bit-error rates (between 0 and 0.005) were obtained for all test signals. After attack and without resynchronization, the bit-error rates approached 0.5.

5.2 Experimental results

In the first method (additional synchronization watermark), codebooks $\dot{\mathbf{C}}$ and $\ddot{\mathbf{C}}$ contained both $\dot{K} = \ddot{K} = 4$ normally-distributed vectors. The resynchronization watermark and the data watermark had the same power. In the second method (sequence of codebooks), $P = 4$ codebooks were used, each one containing $K = 4$ normally-distributed vectors.

Table 1 shows the bit error rates for all test signals. Besides random suppression/addition of samples and all-pass filtering, the signals were submitted to an MP3 compression/decompression process (layer 3, mono, 128 kbps). The average signal-to-watermark power ratio and the signal-to-noise power ratio (corresponding to the MP3 compression) are also shown. As can be seen from this table, both methods lead to bit-error rates that are significantly lower than those obtained without resynchronization (≈ 0.5). The reduction is stronger for the second method (sequence of codebooks). This is explained by the fact that, when two watermarks are present simultaneously, their individual power must be reduced in order to keep inaudibility, thus increasing detection errors rates.

Signal	First method			Second method		
	SWR	SNR	BER	SWR	SNR	BER
svega	16.2 dB	10.4 dB	0.035	17.6 dB	10.5 dB	0.010
violon	16.5 dB	10.6 dB	0.080	18.0 dB	10.5 dB	0.037
baron	16.2 dB	12.6 dB	0.098	17.2 dB	12.2 dB	0.030
queen	16.6 dB	8.5 dB	0.100	17.6 dB	8.7 dB	0.020

Table 1: Signal-to-watermark ratios (SWR), signal-to-noise ratios (SNR) and bit-error rates (BER) for the first method (additional watermark) and for the second method (sequence of codebooks).

Figure 7 shows the bit error rates as a function of the signal-to-watermark power ratio for signal "svega" with both resynchronization methods. Noise was added to the signal to simulate an attack (signal-to-noise ratio of 20 dB after spectral shaping according to the masking threshold to avoid audibility). As expected, the error rate increases as the watermark power is reduced. This experiment confirms the better performance obtained with the second method.

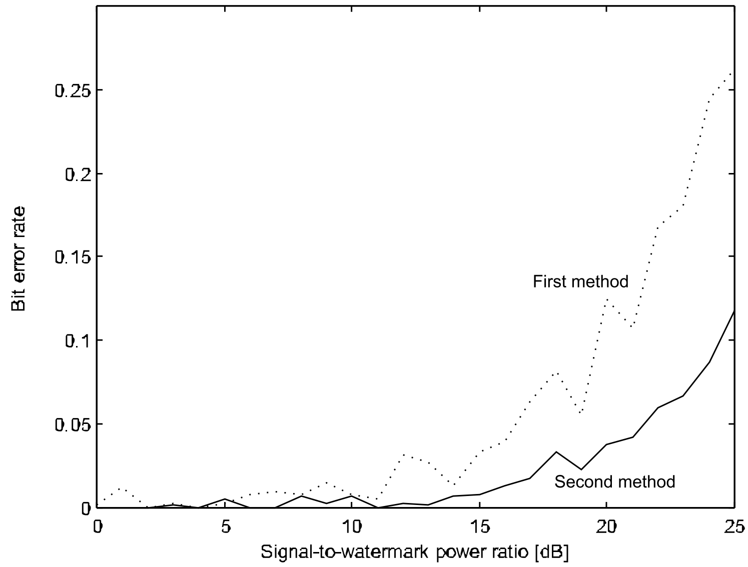

Figure 7: Bit error rate as a function of the signal-to-watermark ratio with the first method (additional watermark) and with the second method (sequence of codebooks) for signal "svega".

6 Conclusions

We have presented resynchronization methods that enable a watermarking system to resist a large class of piracy attacks. These methods are based on the use of training sequences that are spread over the watermarked signal. Simulation results show that these methods succeed in reversing the effect of desynchronization attacks consisting in erasing or adding samples to the watermarked signal.

By using error-correcting codes, the error rates after resynchronization could be further reduced (at the cost of bit rate), thus enabling the use of this watermarking technique in applications that require highly reliable detection.

Additional research is necessary to improve resistance to attacks that significantly modify symbol length (time warp and time stretching). In order to do so, the methods presented in this paper could be extended by computing correlation measures between the watermarked signal and versions of the codebooks modified by such attacks.

References

[1] C. Neubauer and J. Herre, *Advanced watermarking and its applications,* 109th AES Convention, Los Angeles, September 2000.

[2] N. Packham and F. Kurth, *Transport of context-based information in digital audio data,* 109th AES Convention, Los Angeles, September 2000.

[3] L. Boney, A. Tewfik, and K. Hamdy, *Digital watermarks for audio signals,* IEEE Int. Conf. on Multimedia Computing Systems, Hiroshima, June 1996.

[4] L. Boney, A. Tewfik, and K. Hamdy, *Digital watermarks for audio signals,* Eurospeech, Rhodes, September 1997.

[5] R.A. Garcia, *Digital watermarking of audio signals using a psychoacoustic auditory model and spread spectrum theory,* 107th AES Convention, New York, September 1999.

[6] T. Furon and P. Duhamel, *An asymmetric public detection watermarking technique,* Proc. of the 3rd Int. Work. on Information Hiding, Dresden, September 1999.

[7] T. Furon, N. Moreau, and P. Duhamel, *Audio public key watermarking technique,* Proc. Int. Conf. Acoust., Speech and Signal Processing, Istambul, June 2000.

[8] L. de C.T. Gomes, M. Mboup, M. Bonnet, and N. Moreau, *Cyclostationarity-based audio watermarking with private and public hidden data,* 109th AES Convention, Los Angeles, September 2000.

[9] E. Zwicker and E. Feldtkeller, *Psychoacoustique, l'oreille récepteur d'information,* Masson, Collection Technique et Scientifique des Télécommunications, 1981.

[10] M. Perreau Guimarães, *Optimisation de l'allocation des ressources binaires et modelisation psychoacoustique pour le codage audio,* PhD Thesis, Université Paris V, Paris, 1998.

[11] E. Gómez, *Tatouage de signaux de musique (méthodes de synchronisation),* technical report (DEA ATIAM), Université de la Méditerranée / ENST, Paris, France, July 2000.

[12] N. Moreau, P. Dymarski, and L. de C.T. Gomes, *Tatouage audio : une réponse à une attaque désynchronisante,* CORESA, Poitiers, France, October 2000.